



UNIVERSITAT DE
BARCELONA

System building

**Estudio etnográfico de los proyectos de investigación de la
School of Computer Science de *Carnegie Mellon University*,
un "computer-intensive campus" norteamericano**

Arturo Serra Hurtado



Aquesta tesi doctoral està subjecta a la llicència **Reconeixement- Compartiqual 3.0. Espanya de Creative Commons.**

Esta tesis doctoral está sujeta a la licencia **Reconocimiento - Compartiqual 3.0. España de Creative Commons.**

This doctoral thesis is licensed under the **Creative Commons Attribution-ShareAlike 3.0. Spain License.**

TESIS DOCTORAL.

"DESIGN CULTURE".

(Estudio etnográfico de los proyectos de investigación
de la School of Computer Science de Carnegie Mellon University,
un "computer-intensive campus").

ARTURO SERRA HURTADO.

DEPARTAMENTO DE ANTROPOLOGIA CULTURAL
E HISTORIA DE AMERICA Y AFRICA.
UNIVERSIDAD DE BARCELONA.

1992.

A la comunidad de Carnegie Mellon.

AGRADECIMIENTOS.

Este proyecto tiene su origen en las Jornadas Experimentales denominadas "Modelos de futuro, nuevas tecnologías y tradición cultural" que en Noviembre de 1988 organizó el Equipo de Dinamización de Nuevos Modelos de Cultura y Sociedad, dirigido por Arcadio Rojo y Arturo Serra, y el Departamento de Antropología Cultural e Historia de América y África, dirigido por la Dra. María J. Buxó, sin cuyo respaldo no hubieran podido realizarse.

Dichas Jornadas recibieron, a su vez, el apoyo del Dr. Salvador Giner, entonces director del Departamento de Sociología y Metodología de las Ciencias Sociales de la Universidad de Barcelona. A su vez contaron con el patrocinio de la Fundació Caixa de Barcelona, dirigida por el Sr. Joan Rigol.

A dichas Jornadas invitamos al Dr. Angel Jordan , entonces Provost de Carnegie Mellon, que ofreció al equipo dirigido por la Dra. Buxó su Institución como un primer "laboratorio" experimental para iniciar el trabajo de campo de un proyecto de investigación que elaboramos a partir de dichas jornadas y que denominamos: "Ciencias del diseño, nuevas tecnologías y tradición cultural".(1990-1991). Esta tesis es uno de sus frutos.

El Rector de la Universidad de Barcelona, Josep M. Bricall, el Dr. Angel Jordan, por Carnegie Mellon , y el Sr. Santiago Guillén por el Centre Divulgador de la Informàtica, S.A. de la Generalitat de Catalunya firmaron un convenio de colaboración entre esas tres instituciones para dar respaldo a dicho programa. A su vez, este programa recibió pequeñas ayudas de la CIRIT, dirigida por Artur Bladé, del Comité Conjunto Hispano-Norteamericano, dirigido por María Jesús de Pablos y de la Fundació Catalana per a la Recerca, el Dr. Planas y Castellví.

A todas estas personas que han hecho posible la realización de esta tesis, mi más sincero agradecimiento, en particular, a mi colega y amigo, Arcadio Rojo, a mi directora de tesis María J. Buxó, a mi generoso "informante" Angel Jordán, y al patrocinador del proyecto, Santiago Guillén, sin cuya financiación éste hubiera sido inviable.

En cualquier caso, esta tesis no hubiera sido posible sin el buen humor y la entereza de mi mujer Tonia, durante estos dos largos años de separación para elaborar este trabajo y su inapreciable labor con Alejandro y Clara en mi ausencia. Gracias, en especial, a vosotros.

INDICE GENERAL.

PRIMERA PARTE.

"DESIGN CULTURES": TEORIA DE LAS TECNOCULTURAS.
(OBJETIVOS DE LA INVESTIGACION, MARCO TEORICO E HIPOTESIS DE TRABAJO).

SEGUNDA PARTE.

CARNEGIE MELLON, UN "COMPUTER-INTENSIVE CAMPUS".
(DE INSTITUTO DE TECNOLOGIA A UNIVERSIDAD DE INVESTIGACION).

TERCERA PARTE.

LA SCHOOL OF COMPUTER SCIENCE:
UN CENTRO PRINCIPAL DE INVESTIGACION DE CMU.

1. QUE SIGNIFICA COMPUTER SCIENCE EN LA SCS.
2. LA "ARTIFICIAL INTELLIGENCE" DE CMU.
3. EL "PROGRAMMING SYSTEMS" DE LA SCS.
4. "COMPUTER SYSTEMS" EN LA SCS.
5. "THEORY" EN COMPUTER SCIENCE DE CMU.

CUARTA PARTE.

CARNEGIE MELLON COMO "DESIGN CULTURE".
LIMITES DE UN COMPUTER-INTENSIVE CAMPUS Y PERSPECTIVAS.

QUINTA PARTE.

METODOLOGIA Y BIBLIOGRAFIA.

PARTE PRIMERA.

"DESIGN CULTURES":

TEORIA DE LAS TECNOCULTURAS.

(Objetivos de la investigación, marco teórico , e hipótesis del trabajo).

1. OBJETIVOS DE LA INVESTIGACION.

1.1. TECNOANTROPOLOGIA Y TECNOCULTURAS.

Este es un estudio antropológico sobre la investigación en un "computer-intensive campus", donde durante los años 80s, la tecnología informática se generalizó formando una "computer culture". Este es un estudio sobre esta comunidad de conocimiento.

La doctora Maria J. Buxó, en colaboración con el Dr. Angel Jordán, ha definido la tecnoantropología como : "el estudio de la tecnología como un sistema cultural. Ello significa el análisis del contexto social (producción, consumo, recursos humanos, redes de colaboración,...) así como el conocimiento cultural (ciencia, ideología, sentido común...) donde esta tecnología es construída (industria, sociedad, instituciones...) y su feedback sobre nuevas pautas de adaptación social y de innovación de conocimiento. Desde un punto de vista prospectivo, la tecnoantropología elabora los sistemas expertos de conocimiento desde los cuales el diseño cultural puede ser realizado para la innovación de la productividad y la calidad del trabajo humano en la industria, corporaciones e instituciones de investigación y de enseñanza."

Por su parte, el Dr. Salvador Giner ha acuñado el término "tecnoculturas" para designar un fenómeno social emergente en las sociedades industrializadas, nacido de las tecnologías de la información, y caracterizado por su incesante capacidad de innovación. "Estamos pasando de una sociedad apoyada sobre un flujo de innovaciones técnicas materiales (como lo ha sido hasta ahora la sociedad industrial) a otra que reposa en un flujo de innovaciones técnicas abstractas..." ("Ensayos civiles". 1987: 141).

Dentro de estos marcos de referencia, mi estudio está centrado en un tipo particular de tecnoantropología, la dedicada al diseño y análisis de las tecnoculturase entendidas como un tipo de comunidades posibles basadas

predominantemente en el conocimiento y la actividad humana de invención e innovación, como valores culturales que se aplicables al conjunto de esferas de la vida social: los sistemas de conocimientos, de actividades, de organizaciones, de sistemas tecnológicos, y de ecosistemas y relaciones organizadas con el resto de la naturaleza, dentro de este planeta y más allá del mismo.

Es cierto que en toda cultura humana se han producido procesos de innovación, y en particular de innovación tecnológica, pero sólo recientemente se está empezando a considerar esta actividad con una función de hegemonía cultural. Sólo recientemente la tecnología se ha empezado a considerar "high technology", y ésta empieza a considerarse un conocimiento básico de una tecnocultura.

Desde este punto de vista, esta rama de la antropología surge de la combinación del concepto central de la antropología : el concepto de cultura, y de una tarea característica de la moderna tecnología informática: la actividad de innovación mediante el diseño, y en particular, la programación.

Para el fundador de la antropología cultural, Edward B. Tylor, "la Cultura o la Civilización, tomada en su amplio sentido etnográfico, es ese complejo conjunto que incluye el conocimiento, las creencias, las artes, la moral, las leyes, las costumbres y cualesquiera otras aptitudes y hábitos adquiridos por el hombre como miembro de la sociedad" (1977:19). En palabras de Claudio Esteva: "Definimos como cultura, sobre todo, el modo común de pensar organizado de los individuos de una sociedad en orden a producir actividades sociales coherentes, tanto de acción material como de acción espiritual" (1984:65).

Por su parte, Herbert Simon, uno de los padres fundadores de la Inteligencia Artificial, y profesor de CMU, considera que el diseño es la principal marca que distingue las profesiones de las ciencias: "Design...is the core of all professional training; it is the principal mark that distinguishes the professions from the sciences" (1981: 129).

En este sentido, el concepto tecnocultura o cultura de diseño puede traducirse por el de "design culture", dado que el diseño es una actividad intelectual central a este modo cultural emergente.

El campo de estudio no son pues todas las culturas tecnológicas, sino en particular, aquellas que se desarrollan como "high tech", y van asociadas a tecnologías como el ordenador, la bioingenierías o los nuevos materiales. Como veremos, estas son tecnologías particulares, distintas a las ingenierías

tradicionales y a la ciencia. El adjetivo "high" no es accesorio. Indica el tránsito de unas culturas tecnológicas ,hasta ahora concebidas como aplicación de la ciencia, a un nuevo tipo de culturas tecnologías con una teoría emergente propia distinta a la de la ciencia, y que puede disputarle a ésta el papel de conocimiento hegemónico en las culturas industriales.

La tecnoantropología que esta tesis desarrolla tiene como campo general de estudio este tipo de fenómeno tecnológico , y persigue contribuir al diseño de este tipo de tecnoculturas.

1.2. CMU COMO "COMPUTER-INTENTIVE UNIVERSITY" : EL OBJETO DEL ESTUDIO ETNOGRAFICO.

No obstante, el tema de las tecnoculturas es tan amplio que exige un recorte en este estudio. Existen "computer cultures" industriales o en organismos administrativos que no son objeto de este análisis. Nos hemos ceñido a estudiar una comunidad de conocimiento, aquel tipo de comunidad dentro de un sistema cultural dedicada a la elaboración de los patrones de conocimiento del conjunto del sistema.

Este estudio es un análisis etnográfico de la "computer science community" de Carnegie Mellon University, el primer campus universitario de Estados Unidos en establecerse como "computer-intensive campus". Dentro de éste, hemos estudiado los proyectos de investigación principales de la School of Computer Science, un centro de investigación de esta comunidad universitaria.

Este "computer- intensive campus" es una "computer culture" universitaria. Este estudio está ceñido a una "research university", un tipo particular de universidad cuya actividad fundamental es la investigación, actividad de donde obtiene una parte sustancial de sus ingresos. Existen universidades que son esencialmente centros docentes que han establecido redes informáticas, que no conciernen a este estudio.

Este es un estudio de una comunidad de conocimiento, dedicada principalmente a producir nuevo conocimiento mediante la investigación.

Dentro de esta comunidad de conocimiento hemos seleccionado, por tanto, aquella parte más específicamente propia de la misma: su actividad de investigación, por la cual es reconocida como una universidad de excelencia en el conjunto del mundo universitario norteamericano. Y dentro de esta actividad de investigación

hemos escogido aquella, la realizada en la área llamada "Computer Science", que es la más reconocida en la cultura universitaria no sólo del país sino a escala internacional. Otras áreas culturales de esta comunidad como su sistema de jerarquías o su infraestructura material se analizan sólo en tanto en cuanto tienen relación con ese objeto de estudio.

CMU es pues, una universidad de investigación, cuyo campo principal, no único, en estos momentos, es "computer science ". A lo largo del estudio se utiliza el término "computer science" y no la traducción castellana "informática", dado que tiene un particular sentido en este contexto cultural. A diferencia de las escuelas de informática como en España, integradas inicialmente en Universidades Politécnicas, Computer Science en CMU se constituye en el período en que esta institución pasó de ser un instituto de tecnología a denominarse universidad. La evolución de esta cultura en CMU ha asociado dos términos "computer" y "science" en una particular relación que no es traducible por el término "informática".

El campo de invención principal de CMU está centrado en el desarrollo del conocimiento básico de los ordenadores. Este conocimiento tiene como áreas centrales: la "Inteligencia Artificial", el "Programming Systems", el "Computer Systems" y la "Computational Theory". Pero no se reduce a ello. La cultura de "computer science" se extiende a otras ramas como "Robotics", ramas de Humanidades y Ciencias Sociales: "Computational Linguistics", Cognitive Science, Educational Computing, Computer Music. Se extiende a la Ingeniería, Electrical and Computing Engineering. Al Management, Information Systems. Al resto de Ciencias, en forma de "Scientific Computing".

1.3.EL PROBLEMA: ¿ CULTURA DE LA "SCIENCE AND TECHNOLOGY" O CULTURA DE LA "TECHNOLOGY AND SCIENCE"?

Vivimos en una época, que socialmente se denomina "sociedad de la información". En ella, se afirma que el "conocimiento" es un valor cultural central de esta era.

Esta denominación surgió en los años 60s y 70s, gracias a un conjunto de estudios entre los que destaca, hasta la fecha el más completo, el del sociólogo de Harvard, Daniel Bell , "The Coming of the Postindustrial Society" (1973). Según este clásico, la significación de este tipo de sociedad era ante todo " La consolidación de la ciencia, y los valores cognoscitivos como necesidad

institucional básica de la sociedad".(1976: 64). La sociedad post-industrial era una "sociedad del conocimiento" (ib:249), en el que "la universidad...es la institución principal"(ib: 65) .

Para D. Bell: "El grupo más crucial en la sociedad del conocimiento es, por supuesto, el de los científicos"(ib:253). Para Bell, la cúspide de esa clase alta de la "Ciudad Científica" en 1968 estaba compuesta en Estados Unidos por una élite de 297.492 personas, con una mayoría de ciencias físicas (53%), un 20% de ciencias de la vida, y un 11% Matemáticas y Computadoras (2% para computadoras), y 17% para ciencias sociales. Los ingenieros, según este autor, no formaban parte de esta élite.

Según esta teoría, el poder de las naciones era proporcional al poder de su conocimiento científico. Una forma de medirlo era el índice de gastos en Investigación y Desarrollo en relación con el producto interior bruto. A más R&D, más conocimiento científico y más poder nacional. En los años 70s, este modelo parecía evidente en Estados Unidos. La área clave, por lo tanto, de la política de un país era la "policy of science".

Pero este modelo ha empezado a entrar en crisis en los años 90s.

En los años 80s, determinados estudios han empezado a señalar los límites de este modelo cultural.

Uno de los más influyentes fue el informe "Made in America" (1989) del Massachusetts Institute of Technology . Este informe critica la idea de la "sociedad postindustrial" basada en los servicios, que amenaza la desindustrialización del país,(ib.:39). A su vez, el excesivo énfasis puesto tras la II Guerra Mundial, en Estados Unidos en la ciencia en detrimento de la ingeniería se considera una de las razones de la pérdida de liderazgo tecnológico frente a japoneses y alemanes . (ib:77-80). Esta pérdida de liderazgo se ha intensificado al finalizar la Guerra Fría.

Científicos y expertos en Estados Unidos, como Lewis Branscomb, antiguo defensor de la " policy of science", sostienen ahora la necesidad de una "Technology Policy" para Estados Unidos (1991). Alan Bromley, asesor para asuntos científicos del presidente Bush, sostiene por primera vez la urgencia de dicha "technology policy" para la defensa del interés nacional. Un panel de expertos, encabezado por William D. Phillips, Associate Director for Industrial Technology, de la Office of Science and Technology Policy de la Executive Office of the Presidente, ha emitido un reciente informe que basa en un conjunto de 22

tecnologías claves , el grueso de las cuales se agrupan en torno al grupo "information and communications"(1991).

El hecho que ha evidenciado la crisis del modelo anterior es el declive económico del país que aún cuenta con el más poderoso sistema científico-técnico del mundo: Estados Unidos. Por el contrario, se asiste al surgimiento de competidores como , Japón, y Alemania, que con una producción científica inferior tienen un creciente poder tecnológico e industrial. Un campo donde este hecho se ha puesto en evidencia es el de las "computer industries". Estados Unidos, con una mayor capacidad en Computer Science está empezando a perder terreno industrial frente a Japón. Ello ha venido provocando desde finales de los 80s, sucesivas reflexiones sobre la crisis de este modelo.

En Europa, esta diferencia entre cultura científica y cultura tecnológica ha sido detectada por el francés, Jean Jacques Salomon, profesor del Conservatoire National des Arts et Metiers.(1989).

Según este experto en políticas de I+D: "L'effort d'innovation ne se confond pas avec l'effort de recherche scientifique proprement dit...Le succes spectaculaires du Japon dans les domaines technologique et industriel n'ont jusqu'a present mis en jeu que des contributions mineures au progres de la science en tant que telle" (1989:35).

Así los éxitos industriales de Alemania, en comparación con Francia, según Salomon, se deben al cuidado en la eficacia económica de las inversiones en R&D, que son realizados en un 62% por la industria, mientras que en Francia, es el Estado el que financia la mayoría, el 53%.

Estas reflexiones apuntan a la puesta en marcha de "politiques d'innovation", basadas en una "culture technique", distintas a las tradicionales "science policy". (1989:35).

Pero este modelo, curiosamente, no es japonés, ni alemán sino, inicialmente, norteamericano. Japón está en gran parte adaptando un modelo que nació en la cultura norteamericana : la primacía de la cultura tecnológica respecto a la científica.

Como indica Ralph Gomory, presidente de la Sloan Foundation: "The United States was the leading industrial power well before it became the leading scientific power. When during the 1920s, the capitals of science were the European universities, the United States excelled in worker productivity and per capita income and had the biggest trade surplus...Now U.S. universities are the

capitals of science, and Japan has the trade surplus". (1989:99).

La nueva cultura tecnológica, que está naciendo de la "high tech", está provocando cambio de valores sustanciales. El valor clásico de la cultura científica, que según Robert Merton era el "universalismo", está siendo reemplazado por el de "innovation", una característica propia de la tecnología. Carnegie Mellon es un caso interesante para analizar la evolución de estos modelos. Antes de la II Guerra Mundial fue un "institute of technology". Tras ella, se convirtió en una "research university"

Y ahora este modelo está en crisis en los 90s, abriéndose una reflexión sobre su porvenir. Su estudio puede darnos pistas para entender la crisis del modelo y dar elementos para su solución.

Durante los últimos 30 años, Carnegie Mellon se ha ido constituyendo, ante todo, en una "computer-intensive campus". 1. Es una universidad de investigación de excelencia en Estados Unidos, país que inventó este sistema de R&D. 2. Su investigación más avanzada es en el campo de los ordenadores, investigación en la área que se denomina "computer science", estando clasificada entre las tres primeras universidades de Estados Unidos, junto con el M.I.T. y Stanford. 3. CMU se ha distinguido dentro de esta área por ser el centro pionero en "Artificial Intelligence" y "Cognitive Science" del país. Este fue uno de los centros fundacionales de la "Cognitive Revolution". Y 4. CMU fue la primera universidad del país en establecer un "computer-intensive campus", una red informática para todos los profesores y estudiantes del campus, con la que intercomunicarse y construir "educational software".

Carnegie Mellon ha ido haciendo de la "innovation" un objetivo cultural característico y diferenciador de su comunidad: "Carnegie Mellon is committed to innovation and professional excellence. While offering each student a strong professional education, the University seeks to confront students with perspectives drawn from diverse disciplines across the arts, humanities, sciences and technology. This confrontation of perspectives, while not always comfortable, encourages new ways of thinking"(University Goals for General Education. The Search for the President.1989).

Pero , al mismo tiempo, Carnegie Mellon se enorgullece de haber conseguido colocarse entre las universidades de élite del país, donde la ciencia es el saber dominante. Allí donde se imparte la educación en "arts and sciences" tradicional. De ser un "institute of technology", en 1967 cambió su nombre por

el de "university" bajo la dirección del físico H. Guyford Stever. La "etapa científica" de esta institución es precisamente la etapa en la que la "computer science" ha reinado como principal área de investigación, desplazando el protagonismo de la ingeniería. Y esta etapa es la que está ahora en crisis en todo el país.

No obstante, CMU ha conservado este doble carácter como se refleja en su modelo de educación denominado: "liberal-professional education", que la diferencia de la educación habitual en las otras universidades que se denomina "liberal arts education".

Esta ambigüedad se refleja, también en el carácter de su área fundamental de investigación: "computer science". La área tradicional de esta escuela era la denominada "engineering", y ésta incluía en su interior a la ciencia. La facultad principal de la antigua Carnegie Tech era el College of Engineering and Science.

Tras la II Guerra Mundial, nació el Department of Computer Science en esta escuela. Nació formado por matemáticos (Perlis) y científicos sociales (Simon y Newell) que diseñaban sistemas informáticos. Ya no sólo los ingenieros diseñaban sino también otras disciplinas del campus. Este protagonismo de científicos, no de ingenieros, al menos en CMU, en el diseño de este tipo de máquinas, condujo a que la disciplina se denominara "computer science" en este campus. Y este modelo sirvió de base para lo que Herbert Simon llamó "sciences of design". Ello ha tenido interesantes repercusiones.

El sistema denominado "Science and Technology" fué inventado en Norteamérica tras la II Guerra Mundial, dirigido por la comunidad de físicos que habían participado en el Manhattan Project. La "computer science" siguió inicialmente este modelo, adaptándose al esquema Science&Technology. Pero este modelo ha encajado en este esquema con dificultades. Durante décadas ha habido y siguen habiendo debates en Estados Unidos sobre si Computer Science era una ciencia o una ingeniería, o una combinación de ambas y cuál primaba sobre cuál.

La "cybernetics", el "complex information processing", la "computer science", disciplinas fundacionales de las "computer cultures", son conocimientos de síntesis entre ciencia e ingeniería, cuando no entre ciencia, ingeniería y ciencias sociales como la psicología, la lingüística y la lógica, en el caso de la Inteligencia Artificial.

Estos conocimientos se denominan tanto "knowledge" como "know-how", esto es, "un

conocimiento de la forma de hacer las cosas" (Simon, 1977:160), un saber-como-hacer una determinada tarea. Se consideran no sólo meras prácticas profesionales o habilidades propias del experto ("skills") sino también conocimiento teórico, pero distinto al saber propio de la física o la química. Tradicionalmente, la técnica se ha considerado en la cultura occidental una mera habilidad, una mera operación, un "doing", que se aprendía con el mero ejercicio, "learning by doing". Por el contrario, la Academia, al modo platónico, se encargaba de la "theory", o conocimiento como contemplación. La moderna ciencia de Galileo introdujo la experimentación en el estudio de los fenómenos naturales, pero su objetivo era confirmar la certeza de las hipótesis sobre la naturaleza establecidas por medio de las matemáticas.

El computer science es distinto. Su objeto de estudio es un artefacto: produce un saber sistemático sobre como inventar nuevas máquinas y sus programas. Se muestra como un conocimiento diferente a la mera práctica operativa, pero, al mismo tiempo, distinto a las otras ciencias meramente explicativas o interpretativas, destinadas a observar o a razonar sobre el mundo.

Este tipo de conocimiento lo expresa en su forma más acabada el programa informático. No es una teoría al modo de las leyes de la mecánica newtoniana, pues indica un modo de operar. No persigue la búsqueda de la verdad, sino la ejecución de una acción por un objeto técnico inventado por el propio investigador. También los científicos naturales han inventado artefactos para hacer ciencia, como el telescopio de Galileo. La diferencia es que ahora, el caso es el inverso. Ahora el computer scientist utiliza las matemáticas, y otras ciencias como la psicología cognitiva, para hacer tecnología. Prima el objeto artificial, sobre el llamado "orden natural".

De ahí se derivan dos modelos de investigación distintos: uno, el aún dominante, se denomina "Science and Technology". El segundo, podríamos denominarlo a la inversa: "Technology and Science", para indicar que el papel de iniciativa le corresponde a la tecnología, y su patrón cultural es la innovación, no el universalismo, propio de la ciencia clásica, según Merton.

El primero modelo es, principalmente, el de la National Science Foundation, basado en la definición de Vannevar Bush, director de la Office of Scientific Research and Development en su famoso libro "Science, the Endless Frontier"(1945). Está compuesto por tres etapas sucesivas: "Basic Research"- "Applied Research"- "Development". Según Bush, la primera es: "Basic

Research" o "Pure research is research without specific practical goals. It results in general knowledge and understanding of nature and its laws. This general knowledge provides the means of answering a large number of important practical problems, though it may not give a specific solution to any one of them" (Bush, 81). 2. La segunda y tercera: "Applied Research and Development". Su diferencia con la anterior es que éstas se dirigen a objetivos y sus resultados tienen un valor práctico o comercial. Este ha sido el programa de conocimiento cultural conocido hasta hoy por Research and Development. (R&D).

Su arquitecto Vannevar Bush lo concibió explícitamente como un contrapeso a lo que denominaba la "perverse law governing research": "Under pressure for immediate results, and unless deliberate policies are set up to guard against this, applied research invariably drives out pure". (1945:83).

Este modelo es el que Nathan Rosenberg y L.E. Birdzell, Jr. denominan: "Science, Technology and the Western Miracle" (1990): "Western science had become an institution with a broad general goal (to explain natural phenomena), a division of labor into specialized departments with their own subsidiary goals...A fundamental factor holding the enterprise together was its adoption of a single standard of scientific truth based on observation, reason, experiment and replicability. The standard enabled scientists to make use of findings from other laboratories...It also permitted artisans, merchants, manufacturers and the rest of the working population to apply scientific discoveries to everyday labors". (1990:44).

No obstante, es el segundo, el que se ha seguido mayoritariamente en centros de investigación como CMU. Tuvo su origen en la II Guerra Mundial, en el denominado Manhattan Engineering District, y ha sido continuado, hasta hoy. Es el que está en el origen de la "computer science and technology", aunque este modelo como veremos también participa del anterior, en tanto que se ha considerado parte de la ciencia.

Este modelo se ha financiado principalmente por determinadas agencias del Departamento de Defensa, como DARPA, y su investigación tiende a ser: "mission oriented". Este modelo ha tendido a borrar esta división investigación básica-investigación aplicada tendiendo a fundirla en una sola actividad: la construcción de lo que Henry Riggs, profesor de Electrical Engineering de Stanford denomina "fundamental technology", (1988:250).

Hay pues dos modelos de R+D: uno el que sostiene, preferentemente, la National

Science Foundation (N.S.F) e instituciones como los National Institutes of Health y otro el que sostiene DARPA y agencias similares. El primero es el culturalmente hegemónico. Los orígenes de ambos son distintos: el primero es el tradicional, el admitido oficialmente por la Academia, y el propio de las Universidades tradicionales, el segundo es la que tuvo su origen en la actividad excepcional de diseño de la bomba atómica en el llamado "Manhattan Engineer District" (Hughes, 1989:353) y el que DARPA ha continuado durante 40 años. Este modelo como veremos tiene como uno de sus orígenes el laboratorio de inventos de Thomas.A. Edison, y en instituciones como el M.I.T, minoritarias, pero influyentes, en el mundo académico norteamericano.

No obstante, existen programas en la NSF como el de Engineers Research Centers, iniciado en 1985, que intentan dar una orientación más aplicada a dichas investigaciones.

Computer Science en CMU participa de ambos. Mientras que su área denominada "Theory" es financiada prioritariamente por la NSF, el resto de áreas las financia DARPA.

Este doble modelo de R+D explica que en científicos de CMU como Herbert Simon apareciera incluso una división de campos de investigación distintos: el "natural world" y el "artificial world", el primero el mundo tradicional que estudian los físicos y otros científicos, y el segundo el que diseñan los computer scientists. Nacían dos tipos de científicos, los "natural" y los "scientists of design". Pero el código de conducta de este segundo tipo de científico es distintos al del primero.

Según Robert Merton, el primer valor de la ciencia es el "universalismo". El objetivo institucional de la comunidad científica sería la "la extensión del conocimiento verificado". (1942: 67).

Pero el valor que aparece central en el "scientist of the artificial" es el de "innovation". Y éste es un valor central de la ingeniería. En palabras de Dick Teare, antiguo Dean del College of Engineering and Science de la antigua Carnegie Institute of Technology: " Look at what characterized engineering: perhaps the most striking thing --and it's getting more striking all the time--is the rate of innovation". (Focus, February, no.5 1980).

La ciencia clásica se basa en el descubrimiento de patrones en la realidad natural que se repiten, regularidades, modelos formalizables matemáticamente."Science is all about discovery things" H.Simon. "The Cat That

Curiosity Couldn't Kill". 9-4-91). Su sello distintivo es el descubrimiento. Por su parte, a tecnología se centra en la invención de nuevos artefactos. Su tarea es el diseño.

Es cierto que la tecnología tradicional ha admitido ser una aplicación de la ciencia. Pero ahora está naciendo una "fundamental technology" (H.Riggs) como Inteligencia Artificial, Robotics, Bioengineering,... que son tecnologías que José Sanmartín, profesor de lógica de la Universidad de Valencia, denomina "tecnologías sintéticas"(1989), dirigidas a la creación de objetos artificiales, a diferencia de las "tecnologías de control", como las tradicionales orientadas a la canalización de fuerzas naturales. El control ha sido el tema clave de las tecnologías mecánicas, incluso de la cibernética. El design es el tema clave de las "high tech".

Pero ello plantea una relación nueva con el científico. Si la clave de la "high tech" es el diseño, la ciencia empírica de estas tecnologías, como el "computer science", ha de estudiar lo que diseña el ingeniero, o el propio científico. La relación ciencia-tecnología se invierte. No puede haber ciencia empírica del ordenador sin ordenadores. Por lo cual, primero hay que inventarlo y luego analizarlo.

En su construcción se sigue una metodología de invención ingeniera: Para hacer un sistema informático, lo primero a saber son los "goals", la función de dicho sistema, luego hay que saber con qué se parte para hacerlo, y finalmente qué estrategias o caminos se pueden seguir para construirlo, el cómo hacerlo.

Este tipo de innovación es distinta a la concepción de la tecnología como mera "aplicación" de la ciencia. Se está desarrollando un sistema de conocimiento de una tecnología básica que no es ciencia ni mera aplicación "técnica" de la ciencia. Esa tecnología básica, estratégica, es la que producen centros como CMU, y está protagonizada por un tipo particular de científicos cuya finalidad no es el mero descubrimiento de orden de las cosas sino diseñar nuevos ordenes artificiales.

Norman Chigier, William J. Brown Professor of Mechanical Engineer en CMU resume en esta frase el tipo de conocimiento que podría definir el adoptado por computer science: " ...Design is what one imagines--translated often in engineering by a drawing. It's the concept. As distinct from the person who just goes out and constructs something. The designer plans....Design is the concept, the coordination of the ideas, the detailed instructions of how elements need to

be constructed; it is testing, the proving, and finally it includes the commercial phase...All that is design. They have to conceive the idea and they have to carry it all the way through. This is the process that scientists don't normally get involved in" (Focus. Feb 1983).

La filosofía griega clásica, a excepción de los sofistas, nunca consideró la técnica ("techné") como saber, como "episteme" (Medina, M. 1989). El saber tenía que ser saber objetivo, saber universal. La universidad, decía con razón Daniel Bell, es por ello la institución central de conocimiento de la moderna cultura.

La técnica era tradicionalmente un saber secundario, "aplicado", nacido del saber fundamental y original. Arrancando de esta tradición, nace la revolución copernicana en astronomía, la "sciencia nova" física de Galileo, el sistema newtoniano del Universo, y la teoría de la relatividad de Einstein.

Robert Merton extrajo de este programa de conocimiento un "ethos" de la comunidad científica: "el universalismo", "el comunismo", "el desinterés", y el "escepticismo organizado". (1942:67-77). Daniel Bell, de Harvard, concluyó que la universidad iba a ser la institución central de la sociedad postindustrial.

Pero la Computer Science, y las tecnologías que le siguen, están entre dos mundos culturales. Por un lado se definen como ciencias, y han encontrado ubicación al lado de la matemática. Pero por otro, se rigen por un patrón de innovación, propio de la ingeniería. CMU, hoy, está entre estas dos culturas, y su equilibrio es inestable.

CMU no es la clásica universidad modelada al estilo europeo tradicional, sino más un instituto tecnológico, convertido en un "computer-intensive campus", cuyo centro de innovación principal son los grandes programas de investigación en Computer Science and Technology financiados por DARPA.

El "computer scientist" no se pregunta sólo por el llamado "know-how", no pretende sólo construir sistemas, sino acumular saber sobre su diseño. Su objetivo es el nuevo saber, no sólo el nuevo sistema. Su objetivo no es la mera "performatividad" (F. Lyotard) sino aumentar el diseño, como un tipo específico de conocimiento propio de esta tecnología.

En esto se diferencia del antiguo ingeniero. Su finalidad no es sólo construir máquinas sino aumentar su saber acerca de cómo construirlas. Este matiz es clave en "computer science". Es un nuevo tipo de teórico. Su objetivo final es aumentar su conocimiento, no simplemente fabricar artefactos. Pero a diferencia del científico académico tradicional, su conocimiento no pretende descubrir unas

leyes universales de la naturaleza o de la sociedad, sino saber qué nueva realidad artificial, un nuevo algoritmo, software o hardware puede inventar y cómo hacer que funcione y cumpla unos objetivos que se los pone el propio "computer scientist", en colaboración con la agencia de donde obtiene fondos. Hasta ahora a este saber se le ha llamado "science of design". Sería más correcto denominarlo, "new engineering" pues su matriz cultural es el diseño, no el descubrimiento. Es una nueva ingeniería, concebida como la invención de "fundamental technology", a diferencia de la vieja ingeniería concebida como ciencia aplicada.

Estas nuevas disciplinas ingeniero-científicas están influyendo en la renovación del universo tanto de las disciplinas científicas, como demuestra el campo de la ciencia cognitiva, como de las antiguas ingenierías.

Ello conlleva una dinámica de creación de conocimiento distinta a la tradicional de la Academia. Las "nuevas tecnologías" abren una dinámica hacia la "síntesis de conocimientos". La cibernética era una síntesis de ingeniería de comunicaciones, matemáticas y fisiología. (Wiener. 1948). El "Information Processing", incluye ya una ciencia social más compleja, la psicología cognitiva. La Artificial Intelligence, necesita también la lingüística, la lógica, . Se funden conocimientos y nacen otros nuevos de la síntesis. La propia Ciencia Cognitiva es un resultante de este proceso nacido de la propia "computer culture" y admite en su seno disciplinas científicas e ingenieras, es una combinación de filosofía, lingüística, Inteligencia Artificial, psicología cognitiva, neurofisiología, y antropología. (H. Gardner. 1987).

Estas disciplinas, utilizan como todas las ciencias, unas metodologías analíticas. Pero su finalidad es la síntesis, la producción de un conocimiento sobre qué nuevos objetos artificiales son posibles y cómo construirlos.

La dinámica de la ciencia moderna, según el patrón establecido en el siglo XIX por la universidad alemana (Ben-David, J. 1962:46-59) fue la siguiente: partiendo de cuatro troncos teóricos iniciales: teología, filosofía, derecho y medicina, que englobaban casi el conjunto del conocimiento superior en Occidente en ese momento, se fueron formando disciplinas especializadas. El tronco original denominado filosofía se escindió en filosofía, historia, literatura y ciencias naturales. De esta última, salió la física, la química, la biología. Y así, sucesivamente, hasta llegar a los 24 "campos", 248 "disciplinas" y cientos de "subdisciplinas" establecidas por la UNESCO (Comisión Interministerial de

Ciencia y Tecnología. 1989: 27-69).

La dinámica de la "computer science" ha sido la inversa. Partiendo de diversos campos de la ingeniería (rama que no estaba aceptada siquiera en la antigua universidad alemana), y de la matemática aplicada, se ha ido configurando un nuevo tipo de saber que va englobando a los restantes en su interior, en una dinámica de "putting it all together" (Newell, 1989: 399).

Estos cambios en el conocimiento conllevan cambios en el sistema organizativo de la Universidad. El centro organizador de estos saberes no es ya la propia universidad sino las redes estratégicas de conocimiento que engloban "computer departments", "industrial laboratories" y "federal agencies" como DARPA. Al ser un saber mitad ciencia mitad ingeniería y de dimensión estratégica, pues influye al conjunto de disciplinas y actividades humanas en las sociedades avanzadas, estos saberes han sido y son objeto de atención prioritaria por los gobiernos y las industrias, formándose una red de organizaciones estratégicas en torno a ellos.

La Universidad está desbordada ante estos conocimientos, perdiendo gran parte de la autonomía antes conservada respecto al poder político y el mundo empresarial. Tradicionalmente esta autonomía venía dada por el hecho de sostenerse de las matrículas de los estudiantes. Esto ha desaparecido en la "research university" norteamericana. Los fondos públicos de investigación son los que sostienen principalmente los gastos universitarios, investigación de la que las universidades obtienen el 58% justificados como gastos para pagar el uso de la propia institución.

Por otra parte la independencia de las disciplinas tiende a desaparecer.

Los sistemas científicos tradicionales se organizan en especialidades y departamentos que como su propio nombre indica departamentan, especializan el conocimiento como se ha hecho antes con el objeto de estudio, gracias a lo cual se puede analizarlo y hacerlo de forma exhaustiva, entendiendo como funciona.

Por el contrario, las nuevas tecnologías tienden a utilizar todo tipo de conocimiento para la construcción del sistema tecnológico. Su dinámica es centrípeta, buscan el "putting it all together", expresión propia de CMU, que indica esta tendencia hacia la síntesis.

Las distintas materias se convierten en ramas del computer science: electronic music, information management, electronic engineering, applied mathematics,...

El fenómeno de las "computer cultures" en la Universidad ha partido

principalmente de aquellas con una cultura tecnológica, más arraigada. La cibernética nació en el M.I.T, el "information processing" y la Inteligencia Artificial en la RAND co. y la antigua Carnegie Tech. El "knowledge engineering" y los sistemas expertos en esa institución y Stanford University. Se dice que la "sociedad del conocimiento" tiene como centro la Universidad. Pero esto no es el caso para las "nuevas tecnologías". Son universidades e instituciones particulares con este tipo de cultura tecnológica los centros de las mismas. Por otra parte, el tipo de sistema de financiación que ha alentado el desarrollo de este fenómeno cultural, ha sido y es una política pública, "a public policy", una actividad política estratégica central, ni siquiera local o regional. Los fondos que se requieren para financiar programas de alto riesgo sólo lo ha podido cubrir determinadas agencias especiales como DARPA de la Administración federal, agencias dedicadas a proyectos avanzados y con una gran flexibilidad y agilidad en su funcionamiento interno, y regidas con una mentalidad de tipo "entrepreneur".

Los motores de innovación en la "high tech" son unos conocimientos como la computer science, la Inteligencia Artificial, el Programming, ... unas nuevas tecnologías, planificadas estratégicamente desde determinados departamentos, agencias y organismos públicos con una cultura de innovación, como DARPA, mediante determinadas políticas tecnológicas de carácter "entrepreneurial", cuyo objetivo es dinamizar la capacidad tecnológica de la nación.

Carnegie Mellon aporta elementos claves para entender este modelo cultural, al tiempo que revela su inestable situación y sus límites.

Un límite principal es que este modelo no tiene teoría propia. A nivel del sistema de conocimiento, busca su fundamentación en la matemática, en la psicología cognitiva, o en la ingeniería tradicional. No existe una teoría propia aún, de este tipo de conocimientos nuevos. A nivel organizativo, estas instituciones de innovación no acaban de encajar en las estructuras universitarias tradicionales. Así como las agencias que las financian no responden a la clásica estructura burocrática de la Administración. Una teoría de las tecnoculturas como culturas de innovación integrada podría servir de marco de referencia para el diseño y desarrollo de estos nuevos fenómenos culturales complejos.

2. EL ESTADO DE LA CUESTION.

2.1. PROSPECTIVA ANTROPOLOGICA, ANTROPOLOGIA COGNITIVA Y ANTROPOLOGIA INDUSTRIAL.

Esta tesis se alimenta de elaboraciones teóricas y prácticas que a partir de diversas ciencias sociales he realizado, de forma independiente y en conjunción con Arcadio Rojo y otros investigadores del denominado "Equipo de Dinamización de Nuevos Modelos de Cultura y Sociedad" a lo largo de los años 80s. Estas elaboraciones tuvieron una primera expresión en el libro titulado " Problemas en torno a un cambio de civilización" (1988). En el, escribía el artículo "La posibilidad de un cambio cultural diseñado", en el que se planteaban los problemas básicos que recorren esta tesis. En particular, destaco uno que es central a esta tesis, la posibilidad de un tipo de conocimiento cultural distinto a la religión y la ciencia para liderar en ese cambio: " Estas preguntas requieren posiblemente la elaboración de un nuevo tipo de saber que combine la capacidad proyectista de inventar nuevos fines sociales y culturales, 'la veritá fattuale delle chose', y por último, de trazar caminos prácticos de conseguir los nuevos fines sociales" (1988:54.).

La influencia de la antropología en esta tesis proviene de varias fuentes. La primera de ellas de la "prospectiva antropológica" introducida por Maria J. Buxó en 1987. En un artículo denominado "Antropología Cultural" publicado en ese año, se citaba como aspecto del sentido aplicado de esta disciplina, la "prospectiva antropológica". En el se decía: "La tarea prospectiva consiste en el diseño de alternativas de futuro, formas probables que pueden llegar a suceder por elección del colectivo social a partir de la generación constante y nueva de metas propias. En este sentido, la prospectiva antropológica es más exploratoria que predictiva".(1987:29).

Este artículo fue a su vez materia de un primer curso de doctorado en el Departamento de Antropología Cultural e Historia de America y Africa en 1986-87. A su vez M. J. Buxó y Josep M. Fericgla fundaron el Instituto de Prospectiva Antropológica.

A su vez, se hacía referencia a la Inteligencia Artificial como un posible campo

de estudio del antropólogo prospectivo. Esto concuerda por los esfuerzos de M. J. Buxó por introducir una "antropología cognitiva" en el panorama de la antropología española desde 1978, el campo dentro de esta disciplina más cercano al mundo de la ciencia cognitiva, en particular la área de la Inteligencia Artificial. A través de la antropología cognitiva se llega a la problemática de la "sociedad de la información", del carácter del conocimiento en los distintos modos culturales y su diferenciación. En este sentido, otro artículo clave de M.J. Buxó es "La cultura en el ámbito de la cognición" (1984), donde establece la idea de "modos cognitivos", que ha sido importante para mi distinción entre modos de conocimiento entre "mythos", "logos", y "design".

El artículo sobre la prospectiva antropológica abría a su vez una reflexión sobre una problemática usualmente relegada en el ambiente universitario, la de la antropología aplicada o profesional. El Dr. Claudio Esteva fue el primer antropólogo en introducirla en la antropología peninsular ya en la Primera Reunión de Antropólogos Españoles en Sevilla de 1975 ("La antropología aplicada y su problemática"), unida al tema del "cambio cultural". El tema de la cultura profesional es clave para entender la cultura de Carnegie Mellon, y en general de la tecnología y la figura del ingeniero.

A su vez, la obra de Esteva sobre Antropología Industrial (1973, 1984), permite introducirse en la cultura del diseño.

Este libro supone un notable esfuerzo por situar a la antropología cultural en el marco de estudio de las sociedades urbano-industriales, ampliando su campo de trabajo tradicionalmente limitado al estudio de las sociedades primitivas.

Partiendo de la premisa de que: "En este momento, las organizaciones industriales son el verdadero poder del mundo", Claudio Esteva analiza el tema de la tecnología y del "diseño industrial" según sus propias palabras "...la expresión más avanzada o de vanguardia alcanzada por la cultura científica y tecnológica de una sociedad" (ib:95). Esta actividad, según Esteva estaría protagonizada por "la clase intelectual del pensamiento, la ciencia y la tecnología" que constituiría la parte superior de la pirámide social.

2.2. "ANTHROPOLOGY AND ENGINEERING", "COMPUTING AS A CULTURAL PROCESS", "ANTHROPOLOGY OF TECHNOLOGY": TENDENCIAS RECIENTES EN LA ANTROPOLOGIA NORTEAMERICANA.

El tema de las "computer cultures", se puede considerar dentro de una área más amplia sobre el estudio de las culturas tecnológicas y los procesos de innovación.

En un sentido amplio, existe un creciente interés en la antropología norteamericana en el tema de la "innovation", estudios que encabeza Marietta Baba (1985, 1988), de la Wayne University. Sus trabajos sobre "innovation" y "University and Industry", han retomado el hilo iniciado por el ingeniero y antropólogo, Homer Barnett y su estudio "Innovation: The Basis of Cultural Change" (1953), y están directamente relacionados con una nueva antropología aplicada. Es de destacar por su carácter pionero la sesión organizada por esta profesora en el reciente congreso de la American Anthropological Association denominada "Anthropology and Engineering" en el que participaron tanto ingenieros como antropólogos.

El interés particular de Marietta Baba para esta tesis es su interés en una "applied anthropology", no en un mero estudio de la tecnología como fenómeno cultural a simplemente analizar. Es miembro de la National Association for the Practice of Anthropology (NAPA) y su enfoque es a la vez académico y profesional, como el que intenta realizarse en este estudio.

Así mismo son interesantes los trabajos de Gary L. Downey (1989) del Virginia Polytechnic Institute sobre una "anthropology of technology".

Sobre "anthropology of science" es un clásico el trabajo de Bruno Latour y Steve Woolgar, "Laboratory Life" (1979) sobre la construcción de la ciencia en un laboratorio de biología. No obstante el laboratorios científicos, no deben confundirse con los laboratorios de carácter ingeniero como los existentes en Computer Science en CMU, donde la construcción de "facts" se ha convertido en construcción de "deeds", ésto es, donde el hecho científico ha pasado a ser un artefacto mecánico hecho por el "computer scientist". Más reciente es la obra de Latour, "Science in Action" (1987).

Por último, es preciso conocer la obra de Yehuda Elkana de la Van Leer Jerusalem Foundation y profesor de historia de la ciencia en la Universidad de Tel Aviv sobre una "anthropology of knowledge" (1981. Su obra "Sciences and Cultures", conjuntamente con Everett Mendelsohn y otros autores (1981) es interesante para analizar las relaciones entre religión y ciencia en diversas culturas.

En el sentido más estricto de la antropología de las "computer cultures", a diferencia de la sociología que empezó en los años 70s, la antropología está

empezando a estudiar recientemente el fenómeno del "computing" como fenómeno cultural. El 90º Congreso de la American Anthropological Association, celebrado en Chicago en Noviembre de 1991, acaba de organizar un "Committee on Computing as a Cultural Process", para iniciar el análisis este fenómeno. Diversos antropólogos que llevan varios años trabajando el tema integran este comité, del que el autor de esta tesis forma parte.

Entre los estudios más avanzados se encuentran los de la antropóloga Lucy Schuman, del Xerox Palo Alto Research Center, especializada en Human-Computer Interaction (HCI). Su libro "Plans and Situated Actions: The Problem of Human-Machine Communication" (1987), analiza los sistemas cognitivos de los computer scientists que trabajan en Inteligencia Artificial. Y ella misma diseña dichos sistemas.

Así mismo, la Dra. Diana E. Forsythe (1990,1991) que trabaja en el Intelligent Systems Laboratory de la University of Pittsburgh, está especializada en Expert Systems y la construcción del conocimiento y del trabajo de los computer scientists de dicho laboratorio.

Por otra parte, David Hakken de SUNY (1991) ha desarrollado trabajos de campo en Inglaterra sobre el problema de la "computerization" en el mundo del trabajo. Los primeros trabajos sobre "culturas informáticas" datan de los años 80s. De los tres centros principales de dicha cultura en Estados Unidos, sobre la área de California, es de destacar el estudio de dos sociólogos, Everett M. Rogers y Judith K. Larsen, denominado "The Sillicon Valley Fever. Growth of High-Technolgy Culture".(1982), sobre el nuevo boom industrial en torno a Stanford.

Respecto a la costa Este, destaca el trabajo de campo de la psicoanalista Sherry Turkle, "The Second Self. Computers and the Human Spirit" (1984). Su objeto de estudio es la "computer culture" del Massachusetts Institute of Technology, y en particular el fenómeno "hacker".

El fenómeno de CMU, y la "high-tech" en esta región de Pennsylvania, es el menos conocido de los tres, al no haber engendrado aún un fenómeno industrial concentrado de empresas de alta tecnología, y estar más reducido a una cultura académica relacionada casi exclusivamente con la Defensa. Hemos podido encontrar una tesis doctoral de 1986, de Annette Lois Giovengo, sobre "The Role of the University in High-Technology Economic Development: Pittsburgh as a Case Study", realizada en el Departamento de Historia Aplicada ("Applied History") en la

propia CMU. Así como la tesis doctoral de Stuart S. Shapiro de CMU, "Computer Software as Technology" (1990), que trata en general dicho problema.

En 1987, Sara Kiesler y Lee Sproull, dos sociólogas de la Carnegie Mellon University, realizaron un estudio "Computing and Change on Campus" sobre la experiencia de la red Andrew en dicha universidad. La conclusión que extrajeron fue que, en efecto, existía una "computer culture" en dicha universidad, a la que denominaban paradójicamente "an alien culture" (una cultura extraña) basada también en tipos humanos ya analizados en anteriores trabajos como el "hacker" (fanáticos del ordenador) o el "user" (usuarios). Y concluían que el proceso de enseñanza no había cambiado en lo fundamental en dicha universidad por el hecho de utilizar los ordenadores : " It is true that computer systems are always changing, but the social settings remain relatively stable." (1987:194).

Con todo este estudio no entraba a analizar la School of Computer Science, lugar de nacimiento del Andrew system.

2.3. SOCIOLOGIA DE LAS NUEVAS TECNOLOGIAS, TECNOCULTURAS Y SOCIOLOGIA DE LA LIBERTAD.

Volviendo al panorama español, considero indispensable para entender el tema de las nuevas tecnologías, los trabajos pioneros de Manuel Castells, del Instituto de Sociología de las Nuevas Tecnologías de la Universidad Autónoma de Madrid. El informe de su equipo, dirigido por él, "Nuevas Tecnologías, Economía y Sociedad en España" (1985), fue junto con los encuentros de Buitrago del 86, el comienzo de una política tecnológica en España.

No obstante, a pesar de definir acertadamente el fenómeno como un fenómeno esencialmente tecnológico, de "nuevas tecnologías", aún lo analiza dentro del modelo antiguo de la "sociedad de la información" y del "Sistema Ciencia-Tecnología-Industria" (Ib:899).

Un enfoque más próximo al de la antropología cultural es el del profesor Salvador Giner del Instituto de Estudios Sociales Avanzados del CSIC. Aunque su programa de investigación no es sobre este fenómeno en particular sino sobre el tema de la "sociedad civil", sí ha aportado conceptos claves para entenderlo, como los de "tecnocultura" y "tecnocimientos" (1987: 140).

El artículo titulado "Tecnocultura", de su libro "Ensayos Civiles" (1987), de Salvador Giner, señala acertadamente que: " Estamos pasando rápidamente de una

sociedad cuya dinámica se basa en la innovación técnica a otra en la que , cada vez más, esa dinámica depende de la innovación cognoscitiva sistemática" (1987:141). Este "tecnocimiento" lo separa y lo sitúa por debajo de lo que denomina "sabiduría", el "saber racional, estético y moral".

Este estudio intentará demostrar que también existe una moral en esa tecnocultura, y que la "innovación", que el propio Giner, plantea como la característica clave de la "libertad", es una de sus principales metas. La tecnocultura es una cultura donde el ser humano puede ejercer su mayor nivel de responsabilidad, a diferencia de la hasta ahora cultura científica o logocultura, que ha permitido una separación entre juicios de valor y juicios de verdad. Como vió Max Weber, la cultura profesional es una cultura eminentemente ética, al estar dirigida a la acción ("ascesis"). Y la cultura tecnológica es la culminación de esa cultura profesional. Pero ello, como empezamos a ver ahora, conlleva una inversión de su relación con la cultura científica.

La aportación de la sociología, en su corriente más cercana a la antropología cultural , aquella que arranca de Max Weber, y su análisis de la ética protestante (1985), sigue por Robert Merton (1977), y su análisis del "ethos" de la ciencia moderna, y culmina en Daniel Bell (1973), y su teoría de "las contradicciones culturales del capitalismo" (1976) es imprescindible para analizar el modelo cultural de las "computer cultures". También son interesantes los trabajos de Berger y Luckman sobre sociología del conocimiento, desde un enfoque de "construcción social de la realidad".

Finalmente, es preciso señalar el nacimiento de un nuevo campo en la filosofía, el denominado Filosofía de la Tecnología, que en España encabeza el colectivo INVESCIT (Instituto de Investigaciones sobre Ciencia y Tecnología), dirigido por Manuel Medina (1989) y José Sanmartín, y que en Estados Unidos tiene como figuras claves a Carl Mitcham y George Bugliarello.

2.4. MODOS DE CULTURA , MODOS DE CONOCIMIENTO.

A fin de analizar el fenómeno de las "computer cultures", y sentar bases teóricas de una tecnocultura, hemos de partir de un concepto de cultura considerada no sólo como un fenómeno adquirido , sino también como una construcción e invención humana continua.

Admitimos con el antropólogo cognitivo W. Goodenough que cultura es "shared

knowledge" (1957), un sistema de conocimiento compartido. Este conocimiento compartido es, a su vez, una construcción cultural humana.

Hasta ahora, se ha analizado la cultura como un conjunto de conocimientos, pautas, hábitos e instituciones adquiridas por el ser humano como miembro de una comunidad. Se ha resaltado el carácter de aprendizaje, de adquisición. No se ha tenido tanto en cuenta el estudio de los procesos de construcción cultural, de diseño y edificación de nuevas formas culturales. En suma, los procesos de invención.

La doctora Maria J. Buxo, sin embargo, sí ha venido sosteniendo en línea con el pensamiento constructivista de Vico, Von Glaserfeld (1988), o Piaget, que : " La cultura es una construcción constante de estructuras de realidad...es un instrumento simbólico para la comprensión y la predicción en el ámbito de las relaciones sociales que se modifica dinámicamente según las necesidades e intereses de grupos e individuos" (1987:29).

Margaret Mead , y su "antropología del futuro" abogaba también por esta visión del fenómeno cultural: " Hoy, cuando empezamos a entender mejor los procesos circulares mediante los cuales se desarrolla y transmite la cultura , reconocemos que la característica más humana del hombre, no consiste en su capacidad para aprender, que comparte con muchas otras especies, sino en su capacidad para enseñar ...El aprendizaje, que se funda sobre la dependencia humana, es relativamente sencillo. Pero las aptitudes para crear refinados sistemas aptos para ser enseñados, para entender y utilizar los recursos del mundo natural, y para gobernar la sociedad y crear mundos imaginarios, son muy complejas." (1977: 121).

Un año después publicaba un artículo acerca de la contribución de la antropología a la ciencia del futuro, en el que planteaba como su tarea final: " a source of new designs for living for the extensive worldwide culture building which is needed in today's interconnected planetary system" (1978:3).

Tradicionalmente, en la antropología cultural se han dado generalmente dos tipos de enfoques para explicar el fenómeno de la cultura humana. Por una parte, se ha afirmado que ésta era un resultado necesario de la adaptación del ser humano al ecosistema en el que vive, adaptación que produciría un proceso de cambio tecnológico que determinaba el conjunto de la vida social. Proceso evolutivo que el antropólogo explicaba mediante el descubrimiento de leyes generales, similares a las de las ciencias naturales. (Marvin Harris. 1979).

Por otra parte, se ha sostenido que la cultura era un sistema históricamente transmitido de símbolos compartidos, una herencia o tradición que a modo de superorganismo determinaba el contenido, los hábitos y los sistemas jerárquicos de una cultura particular, siendo tarea del antropólogo la interpretación de sus sistemas de significados. (Clifford Geertz. 1973).

Para unos la cultura eran esencialmente artefactos, para otros conocimientos, símbolos. Esta escisión se verá como un estudio de las tecnoculturas la ayuda a disolver. Hay un conocimiento tecnológico que puede llegar a formar la base de un nuevo tipo de sistema cultural.

Conforme la antropología se ha visto confrontada al análisis de culturas en proceso de cambio, ha tenido que introducir conceptos de "innovation" (H. Barnett. 1953), "cambio cultural dirigido" (Linton), o "Technical Change" (Mead. 1953). Ha debido dar creciente importancia a la antropología aplicada. La innovación cultural empezó a ser estudiada en los años 50s por el antropólogo, Homer Barnett, a su vez ingeniero civil, en su clásico libro "Innovation: The basis of Cultural Change" (1953). Una de las razones para ello fue que los conceptos de enculturación, difusión y aculturación estudiados por la antropología no acababan de explicar como surgían los cambios en las culturas, como se generaba el cambio cultural.

El estudio de H. Barnett permitió comprobar que el cambio cultural se producía por un proceso de innovación que se daba tanto en culturas tradicionales (los Yurok, los Tsimshian, los Yakima) como en la cultura norteamericana contemporánea.

Las computer cultures, que ahora analizaremos, son el resultado de uno de estos procesos de innovación en el interior de culturas urbano-industriales occidentales, en particular en la norteamericana. No estamos en los años 50s. El papel del antropólogo aplicado en los 90s no es ya la incorporación de las "culturas primitivas" a la "modernización", sino el cambio en el interior de las culturas modernas hacia otro modo cultural que aún no sabemos en que consiste, y cuyo uno de sus elementos claves parece ser la tecnología informática, y en general, lo que se denomina la "high tech".

2.5. PROGRAMAS CULTURALES Y MODOS DE CULTURA.

Diversos autores han considerado las culturas como "programas de y para la

acción social que actúa en el ser humano durante el proceso socialización e interacción social adulta". (M.J. Buxo: 1984:33). Para Claudio Esteva: "Los sistemas culturales se caracterizan por el hecho de que se distribuyen como programas de acción en el seno de los diversos grupos sociales de una sociedad". (1984:75).

La definición de cultura como "pattern" es clásica en antropología. Los sistemas de conocimiento compartidos se estructuran en "patrones de comportamiento" (Kroeber y Kluckhohn. 1963), en "pautas tradicionales de acción" (Murdock), o como ha sostenido más recientemente Roy d'Andrade, en forma de "programas culturales" (1981).

Las innovaciones culturales son cambios en los programas culturales.

Para analizar el fenómeno de las computer cultures necesitamos distinguir entre distintos tipos o modos de cultura, y dentro de ellos entre distintos tipos de programas culturales.

Si bien todas las culturas son construcciones humanas en tanto que diseños de significados compartidos, hay que resaltar también que los miembros de dichas culturas han visto su propio proceso de construcción de muy diversa forma. Por ello, podemos distinguir distintos modos culturales o diferentes tipos de construcción cultural que los agrupamientos humanos han seguido para definir sus metas de finalidad, sus sistemas de valores, sus relaciones sociales y de jerarquía, sus sistemas tecno-económicos, en suma, sus distintos sistemas de conocimiento organizados, esto es, sus programas culturales.

Los modos de cultura más exhaustivamente estudiados por la antropología han sido las mito-culturas, o lo que Margaret Mead llamó "sociedades postfigurativas" (1977). Para esta autora, tanto a las sociedades primitivas como a las organizaciones religiosas podrían situarse en esta categoría de sociedades que extraen su autoridad del pasado, de una memoria colectiva codificada en forma de mitos y rituales que estructuran los significados fundamentales de la vida de dichas comunidades.

Agrupado bajo el término de mito-culturas aquél conjunto complejo de sociedades, estudiadas por la antropología bajo el término de "sociedades primitivas", que se construyen a sí mismas en torno a diversos sistemas de creencias basadas en relatos que describen como los antepasados (dioses, héroes,...) crearon el mundo, y con él a los seres humanos identificados como los miembros de dicha cultura.

La obra de Levi-Strauss , las Mitológicas, resume toda una época de estudios antropológicos, iniciados por E.B. Taylor, destinada a entender el sistema de vida y de conocimiento de dichas culturas. " Les societes que nous appelons "primitives" ne le sont en aucune facon, mais elles se voudraient telles. Elles se revent primitives , car leur ideal serait de rester dans l'état ou les dieux ou les ancetres les ont cree a l'origine des temps" (Levi-Strauss. 1990:175). "La creación del mundo" será la forma de construcción común a este tipo de culturas, y el modelo de formación de sus sistemas de conocimiento y sus jerarquías internas.

Los mitos cosmogónicos proporcionar un modelo de explicación y de conducta social a este tipo de culturas. A modo de ejemplo tomemos uno de los clanes winnébago estudiado por P. Radin: " En el principio, el Hacedor-de-la-tierra estaba sentado en el espacio. Cuando cobró conciencia, nada había en ninguna parte. Comenzó a pensar qué haría, y por último se puso a llorar y de sus ojos fluían lágrimas y caían por debajo de él. Luego de un rato miró bajo de sí y vió algo brillante. El objeto brillante por debajo de él representaba sus lágrimas. Al caer, habían formado las aguas presentes. Cuando las lágrimas fluyeron hacia abajo, se convirtieron en los mares tal como son ahora. El Hacedor-de-la-tierra comenzó a pensar otra vez. Pensó: "Es así: si quiero algo, se hará como yo quiero, tal como mis lágrimas se han hecho mares". Así pensó. Entonces deseó la luz y se hizo la luz. Luego penso: " Es como yo pensaba: las cosas que he deseado han venido a la existencia como yo deseé". (Radin, 191968:209). A esta forma de pensar se le ha denominado "realismo conceptual" (Buxó, 1984:44), según la cual los deseos subjetivos son considerados una realidad tanto o más poderosa que la existencia real. En realidad, la objetividad y la subjetividad no forman aún una dicotomía. Curiosamente elementos de este modo de pensar los volvemos a encontrar en la "computer culture" y su creencia en que las construcciones formales matemáticas son ya tecnología en sí misma. "La creacion-destruccion o muerte" será el núcleo común temático de chamanes, artistas, sacerdotes y guerreros de dichas comunidades postfigurativas. Son clásicos los estudios sobre los movimientos de revitalización religiosa de comunidades como los Seneca, estudiados por A.F.C. Wallace (1972), el papel del chamán, de los estados alterados de conciencia, de los códigos morales transmitidos a través de ellos, etc. Estos movimientos se constituyen en el patrón de innovación cultural más profundo de dichos modos culturales. Su

renovación pasa por una "vuelta al origen" , al momento de la creación, de donde extraen las enseñanzas para cambiar sus culturas presentes.

Una utilización de este modo de innovación cultural lo podemos apreciar también en culturas como la japonesa. Al inicio de la Restauración Meiji en 1868, las reformas económicas y políticas se hicieron bajo la consigna de "Sonno joi" : "Restauraremos al emperador y expulsemos a los bárbaros". (Ruth Benedict. 1974:74). Todo el sistema de deudas (on), y de deberes (giri) ante la sociedad y ante sí mismo de la cultura clásica japonesa derivan de esa misma concepción de la cultura como creación de los antepasados, simbolizados en la figura del Emperador. "Isshin": restaurar, volver al pasado, fue la consigna de una reforma que abrió paradójicamente al Japón al mundo industrial.

Este modo de construcción cultural se repite en momentos de crisis de la moderna cultura como estamos viendo con el auge del fundamentalismo islámico en Oriente Medio, en especial en países como el Iran shiita o la Argelia integrista.

Así también, se aprecia en Estados Unidos en el pensamiento neoconservador de escritores como Irvin Kristol o el mismo Daniel Bell: "El principio de la cultura es el de un constante retorno -no en sus formas, sino en sus preocupaciones- a las modalidades esenciales que derivan de la finitud de la condición humana...¿Cuáles son, pues, las normas para la conducta humana?. No pueden estar en la naturaleza... No puede ser la historia...Queda, por ende, la respuesta anticuada y tradicional: la religión...como concepción trascendental que está fuera del hombre, pero relaciona al hombre con algo que está más allá, fuera de él" (1977:160).

2.6. LAS LOGO-CULTURAS.

Coincidiendo con la Depresión de los años 30s y sobre todo tras la II Guerra Mundial , la antropología inició una serie de estudios sobre las culturas urbano-industriales (antropología industrial, antropología urbana, de la salud,...), fundamentalmente en Estados Unidos . Con ellos la antropología ampliaba su campo de estudio pro primera vez a las áreas urbano-industriales. En los años 60s, algunos antropólogos no occidentales espezaron a estudiar la propia cultura occidental desde patrones culturales distintos a los nacidos en Europa. Se empezó a considerar a Occidente no como la civilización sino como una cultura entre otras, tal como había hecho el siglo pasado la antropología colonial con las sociedades primitivas.

Estudios de diversos antropólogos como Magoroh Maruyama (1978), reunidos en torno a ideas de una "antropología del futuro", en gran parte influenciada por el fenómeno denominado "contracultura", proporcionaron a los occidentales elementos interesantes para un análisis de nuestra propia cultura desde el exterior de la misma. Para Magoroh Maruyama, desde el punto de vista del sistema de conocimiento, estas eran sociedades basadas en la lógica aristotélica (1978: xxi). Margaret Mead denominó este tipo de cultura, "sociedades configurativas" (1977). En estas "el modelo prevaleciente para los miembros de la sociedad reside en la conducta de sus contemporáneos" (ib:65).

Podríamos denominar logoculturas al conjunto de sociedades urbano-industriales, cuyo programa cultural se basa en una cosmovisión por la cual el mundo se considera un "orden natural universal", descubierto por medio de la razón humana. Este orden universal o "logos" habría evolucionado hasta culminar en el "animal racional y social" (Aristóteles). Y de esa comprensión racional del mundo se extrae el código de valores de estas culturas.

"El punto de partida para alcanzar esta cosmovisión se inicia con la ideación griega del entendimiento de las cosas en cuanto estas son" (M. J. Buxo. 1989: 215). Como señaló M. Mead, en este tipo de culturas el tiempo verbal culturalmente significativo es el presente, no el pasado propio de "sociedades post-figurativas", ni el futuro, característico de "culturas prefigurativas".

El modo de conocimiento dominante de estas culturas sería para María J. Buxo, el formalismo abstracto: "La abstracción no es lo mismo que el formalismo abstracto...Lo que sí es nuevo y propio de las sociedades urbano-industriales es que esta lógica formal aporta la base estructural de las relaciones sociales y económicas, y dentro del pensamiento ha dominado otras formas de operar mentalmente hasta proveer el paradigma cosmológico único, el cual determina las nociones de realidad y verdad" (M. J. Buxo. 1984: 38).

No obstante, la ciencia moderna se diferencia de la griega es que este formalismo matemático lo ha aplicado a la comprensión de fenómenos empíricos. Como ha señalado A. Koyre, la gran obra de Galileo fue la "matematización del universo".

La modernidad ha prescindido de una explicación del mundo basada en la revelación, para confiarla a la razón y los sentidos. Explicación de donde está excluida no sólo la voluntad de los dioses sino también el libre arbitrio humano. (Salvador Giner, 1987:275). Por ello ciencia y libertad humana son dos

valores conflictivos en el actual modo cultural, que en parte se reflejan en la oposición conocimiento científico, conocimiento tecnológico.

La ciencia, basada en la matemática, y en la lógica, sería el tipo de conocimiento que explicaría este mundo, de cuya evolución, y por selección natural, surgirían todas las especies incluido la especie humana. La naturaleza humana sería en esencia constante, y regida por principios universales, como el propio conocimiento humano.

Las culturas urbano-industriales en que vivimos basan fundamentalmente su legitimidad, como la ciencia natural, en leyes extraídas del libro de la naturaleza, mediante "experiencias sensibles y demostraciones necesarias" (Galileo), y no en la autoridad de los antepasados, o en diseños culturales de futuro.

Pero este tipo de visión del mundo es no solo descriptiva sino fundamentalmente normativa. Lo que es debe de ser. En palabras de Hegel: "La libertad es la conciencia de la necesidad". Este patrón cultural constituye el programa cultural de la ciencia como saber unificador del presente modo de vida. Y conlleva como ha señalado R. Merton un ethos científico, basado principalmente en el valor del "universalismo". Y es la Universidad, la institución que ha terminado siendo internacionalmente el centro del saber científico de la moderna civilización, su organización de conocimiento más compleja, desplazando a la Iglesia o las sectas reformadas, como centros de elaboración de los patrones de la cultura.

El modelo de innovación que este tipo de culturas sigue, preferentemente, es el modelo de la evolución natural. No son "los dioses" los que generan el cambio, mediante "una creación", sino la propia Naturaleza en su progresiva evolución. Los seres humanos ayudan a este cambio, ante todo, adaptándose al mismo, analizando y descubriendo sus tendencias y preveyendo sus futuros cursos. Estos descubrimientos son a su vez utilizados por la tecnología para producir inventos útiles para la sociedad. El ciclo termina en la industria que se encarga de producirlos y colocarlos en el mercado, a fin de satisfacer las necesidades del consumidor. Este ciclo de innovación se ha acabado sintetizando de esta forma: CIENCIA-TECNOLOGIA-INDUSTRIA.

Una fase más reciente de este modo cultural es aquella iniciada en Estados Unidos, a mediados del siglo XIX, donde la TECNO-LOGIA, la creciente conversión de la técnica en un procedimiento científico formalizado, se ha ido convirtiendo

en factor decisivo del cambio social. Este modelo ha dado un paso decisivo con las "computer cultures" en los años 80s, con la "high tech culture".

2.7. TECNOLOGIA Y "COMPUTER CULTURES".

Durante y tras la II Guerra Mundial, en Estados Unidos se desarrolló un tipo de cultura tecnológica basada en una transformación de la ciencia y los científicos tradicionales (matemáticos, físicos, psicólogos, antropólogos...) en científicos diseñadores o científicos-ingenieros, utilizados para fines de interés nacional.

El compromiso de una generación entera de científicos en el esfuerzo bélico y postbélico (Guerra Fría) ha orientado su investigación a objetivos aplicados de tipo estratégico. Este ha sido el origen de la "computer science and technology".

Este programa de investigación contiene, sin embargo, elementos contradictorios. Por una lado es la obra de unos "científico-diseñadores" con características particulares, distintas a los científicos e ingenieros tradicionales. En lugar de tener separado el descubrimiento científico y la invención ingeniera, se ha producido un proceso de síntesis de ambas actividades, lideradas, en la cultura norteamericana, por el avance tecnológico.

Sin embargo, por otra parte, este programa de investigación se sigue reclamando científico, y no ingeniero.

Ha sido Norteamérica, el país que ha aportado a la cultura occidental el concepto de "technology". El historiador Daniel J. Boorstein, director de Biblioteca del Congreso, define a Estados Unidos como "The Republic of Technology" (1978).

Según D. Noble, el que inventó este término fue el médico norteamericano Jacob Bigelow en 1829: "I have adopted the general name of Technology ...Under this title is attempted to include an account...of principles, processes, and nomenclatures of the more conspicuous arts, particularly those which involve applications of science, and which may be considered useful, by promoting the benefit of society, together with the emolument of those who pursue them" (1829:). Este fue el programa cultural en el que se basó en 1865 el Massachusetts Institute of Technology, instituto fundado como institución separada de Harvard, ante la negativa de esta institución a aceptar este tipo de estudios.

Fue en este mismo instituto donde varias generaciones más tarde nacería la "Cibernética", disciplina dedicada al estudio de los mecanismos de control en máquinas y animales, nacida de estudios sobre aparatos y circuitos eléctricos no lineales, que usaban el mecanismo de la retroalimentación ("feedback"). La máquina se transformaba en un autómatas, capaz de cierto gobierno de sus propios procesos mecánicos. (kibernetes). Su utilidad: el perfeccionamiento de la artillería antiaérea norteamericana en la II Guerra Mundial.

Con el desarrollo del ordenador y su extensión al conjunto de actividades intelectuales, Herbert Simon de Carnegie Mellon apuntó la idea de "mundos artificiales" (1969) o mundos diseñados por el propio ser humano, diferentes de los "mundos naturales". El concepto de artificialidad en este profesor, no nació de su contacto con los ordenadores, sino de sus estudios de Administración y su experiencia práctica en la misma. De ésta, y de su "model of man", el "administrative man" cuya actividad central es la "decision-making" extendió esta conducta al terreno de las computadoras, entendiendo finalmente por artefacto todo "complex information processing system", sistema que significa tanto un ordenador como un proceso mental humano.

Las "computer cultures" son culturas tecnológicas-científicas que tienen como centro la actividad de invención de sistemas informáticos. La ingeniería, las ciencias naturales y sociales constituyen disciplinas que sirven de ayuda para la construcción de dichos sistemas, y enriquecen a la propia computer science. El centro de toda esta cultura es el sistema informático, el ordenador. Por eso son "computer cultures", "artificial worlds".

Autores humanistas como el arquitecto Lewis Mumford han criticado lo que denominan "The Machine Age", considerando que lo característico de Occidente ha sido la mecanización del conjunto de los comportamientos sociales. (1934). Sin embargo, no distingue el diferente significado que "mechanization" tiene para un científico europeo tradicional o para un inventor norteamericano.

Mecanicismo en la cultura científica tradicional europea significa ante todo una concepción del mundo, según la cual el Universo se considera como un gigantesco artefacto sometido a las leyes inexorables de la física newtoniana. La Mecánica se considera, en primer lugar, una rama de la Física, no de la Ingeniería. Según esta tradición, no es la máquina industrial el modelo cultural que da significado al Universo, sino al revés. El Universo mecánico de Newton (1642-1727) antecede en varias generaciones a la Revolución Industrial de

Newcombe y James Watt.(1736-1819).

Pero la máquina no se introduce en Estados Unidos a través de la física teórica sino a través de la obra de "mechanics" y científicos aplicados como Ben Franklin, que más tarde se convertirían en ingenieros y sólo recientemente en físicos teóricos. Lo que ha aportado Estados Unidos a Occidente es la consideración de la máquina como una invención técnica, esto es, humana, cuyo valor se lo da el hecho de ser útil para servir a propósitos humanos, no su referencia primaria a un universo mecánico.

Representantes históricos de la cultura científica europea como Bertrand Russell consideraban a la máquina como "hideous, and loathed because they impose slavery", seres horribles y odiosos porque imponen la esclavitud. De aquí se deriva que "la cultura", y por consiguiente la "liberal education" en Europa, en especial en las universidades tradicionales (Oxford, Paris, Lovaina) se considere lo opuesto a la máquina y a las artes mecánicas.

Sin embargo, para el norteamericano medio, la máquina es una obra, ante todo, humana. "Only Man can make a Machine". (Boorstein. 1978:44). " The Machine is the great witness to man's power. The land was there at the Creation. But every machine is the work of man. The power of the Machine is man's power to remake his world, to master it to his own ends. ...It may tempt us to overlook our limitations and put ourselves in the place of God" (ibid:91).

El termino "creator", "creating", "visionary" son de uso común en Carnegie Mellon para definir la actividad de diseño de sistemas informáticos. Los más partidarios defensores del diseño como actividad propia del ingeniero distinta al científico son los "mechanical engineering": "The basic premise of the new design program at CMU is that there is a different set of requirements in a program centered around synthesizing as contrasted to analyzing. The designer is someone who is directed toward producing a result that is in some sense creative" (D. Baumann. Profesor de Mechanical Engineering en CMU. Focus March 19. 1973).

Como afirmó Daniel J. Boorstein : "The Machine brought endless novelty into the world". (:46).

Ello explica que el problema tecnológico, el "problem solving", se considera en este contexto el tipo de problema humano por excelencia: " We Americans have tended to take the technological problem-the soluble problem- as the prototype of the problems of our nation, and then too, of all mankind". (1978:33).

Carnegie Mellon es una expresión culminante de esta cultura en la que se considera al ser humano un "problem solver", y la institución como un "laboratory". Y este desde la tecnología que la cultura norteamericana ve, en gran medida, la política y la sociedad.

El fenómeno actual de la "high-tech" como cultura de invención tecnológica significa avanzar este proceso hasta el punto de inventar una máquina de propósito general, "general-purpose machine", el "computer", apta para todo tipo de conocimiento y actividad humana orientada a fines, a "goals". Y los límites en este objetivo dado que el ordenador no lo ha conseguido, es lo que está ahora impulsando iniciativas en CMU en ir más allá de esta máquina incluso.

La "computer culture", tan sólo el inicio de nuevas tecnoculturas, no es sólo la invención de una herramienta más (el computer), o el surgimiento de una nueva industria más a gestionar. Se está convirtiendo en un sistema tecnológico-social-económico, en suma cultural, cada vez más complejo, cada vez más internacional y global, que tiende a abarcar no una industria sino al sistema industrial en su conjunto, no un país, sino crecientemente al sistema internacional. El mundo se ve como un "artificial world". Y el artificial world se considera una obra humana. Cultura y tecnología se unifican. Si por cultura se entiende la definición de Hershkovits "Man and his works". La tecnología informática abre la posibilidad de considerar por primera vez el mundo como una "human work", como una cultura, un "man-made world".

El proceso la "computer science and technology" para Alan Perlis empieza en 1947 con la invención del primer ordenador digital ENIAC, por el equipo dirigido por el físico John Mauchly y el ingeniero electrónico John Presper Eckert.

(1977:113). A continuación Von Neumann y otros inventaron un ordenador con un programa almacenado: EDVAC. La herramienta empezaba a funcionar con programas, como las instituciones humanas. Se crearon, por tanto, programas para ella. En 1955-56 un equipo de un científico social-un matemático aplicado- y un programador, Simon-Newell-Shaw inventaron el Logic Theorist el primer programa de Inteligencia Artificial. En 1960, otro equipo de matemáticos, entre los que estaba Alan Perlis, diseñó un lenguaje de programación: ALGOL 60. En 1969, Robert Kahn, promovió el equipo de DARPA, que puso en marcha la red ARPANET, hoy la red de conexión de ordenadores más extensa de todo el mundo.

Este proceso conllevó el sintetizar un conjunto de disciplinas científicas, técnicas y humanísticas antes separadas (matemáticas, ingeniería, psicología,

lingüística,...) y formar otras nuevas útiles para la construcción del nuevo sistema. De esta síntesis, están surgiendo nuevas disciplinas, ya especialidades de la Computer Science (Inteligencia Artificial, Programming Systems,), a su vez, subespecialidades de éstas (Machine Learning, Expert Systems...) y combinaciones de Computer Science con ramas de ingeniería para formar otras áreas como Robotics, con ciencias sociales (Computational Linguistics,...) etc. Conforme el sistema se va complejizando nuevas disciplinas empiezan a colaborar con el computer: el management, la música, el diseño industrial, las ciencias sociales, formando nuevas síntesis.

Lo mismo ocurre a nivel organizacional. Lo que era un pequeño departamento, se convierte en una escuela que influencia a toda la universidad. Lo que era una pequeña industria (IBM) influye al conjunto del sistema industrial de un país. Lo que antes era una tecnología de un solo país, se ha convertido en un fenómeno internacional.

Por último, estas nuevas tecnologías se fusionan con otras como la televisión, las telecomunicaciones, etc. formando una red tecnológica global, más y más compleja.

No obstante, el ordenador sigue siendo considerado por la cultura científica mayoritaria como un "tool", una herramienta mecánica al servicio de un conocimiento más elevado. La R&D en Estados Unidos sigue dirigida por las ciencias naturales (física, química, biología, matemáticas), a cuyas investigaciones ha de servir el computer scientist.

Ello conlleva que se considera al ordenador una "calculadora", más sofisticada que hace 40 años pero de la misma naturaleza.

Los primeros ordenadores se inventaron para el cálculo de trayectorias balísticas. "Computation" significa en inglés "cálculo". Los físicos los utilizaban como herramientas para la solución de ecuaciones. Los matemáticos especializados en Análisis Numérico, como G. Forsythe, fueron de los primeros directores de departamentos de Computer Science . En cualquier caso, una herramienta de cálculo para la ciencia y la ingeniería. Esta visión del ordenador aún pervive en los programas de la National Science Foundation basados en supercomputadores. (Grands Challenges: High Performance Computing and Communications. 1992. Office of Science and Technology Policy)

Por su parte, el mundo de la empresa considera aún hoy el ordenador como prolongación de las máquinas calculadoras y las cajas registradoras de hace 40

años, utilizándolo ante todo como un instrumento de "proceso de datos" para llevar la contabilidad y la facturación de la empresa.

Por su parte, los científicos sociales que se han incorporado a la "computer science" también lo consideran una útil herramienta para la "simulación cognitiva" o para el "cooperative work".

Durante estos primeros 40s años el ordenador no ha cambiado la clásica visión de las *technai* establecida por Platón en La República según la cual la educación más elevada no era la de las *technai* que "se orientan hacia las opiniones y deseos humanos o tienen que ver con la creación y fabricación" (533b) sino aquél saber que consiste en la contemplación de la verdad. (cit. Mitcham: 1989: 16)

En cualquier caso, dos tendencias se pueden apreciar: 1. La absorción dentro del ordenador, y por tanto de una "problem solving mentality" del conjunto de disciplinas e instituciones sociales. Ello implica la instrumentación del "computer scientist" del conjunto del saber social en función del "computer". 2. La resistencia de las ciencias y humanidades a ser integradas en esta tendencia, manteniendo al ordenador como una herramienta para realizar sus tradicionales trabajos.

2.8. TECNOLOGÍA, CULTURA, TECNOCULTURA.

Margaret Mead, en los años 70s, detectó éste fenómeno cultural de cambio tecnológico acelerado denominándolo , "sociedades prefigurativas", : "La invención de la computadora, la desintegración efectiva del átomo...el descubrimiento de la bioquímica de la célula viva, la exploración de la superficie del planeta, la extraordinaria aceleración del crecimiento demográfico,...la quiebra de la organización urbana, la destrucción del entorno natural, la interconexión de todas las comarcas del globo mediante los vuelos de retropropulsión y las imágenes televisivas, los preparativos para la construcción de satélites y los primeros pasos que se han dado en el espacio, las posibilidades recién descubiertas de obtener energía ilimitada y materias primas sintéticas...he aquí los factores que sumados, han culminado en una división drástica e irreversible entre las generaciones." (1977:92).

"Los asertos contemporáneos sobre el drama del hombre o, a la inversa, sobre las nuevas oportunidades que se le presentan al ser humano, no contemplan la aparición de nuevos mecanismos de cambio y transmisión cultural que difieren fundamentalmente de los mecanismos postfigurativos y cofigurativos con los que

estamos familiarizados. Sin embargo pienso que está surgiendo una nueva forma cultural y la he denominado prefiguración". (1977:91).

Margaret Mead detecta elementos tecnológicos nuevos, pero los pone en el mismo plano que descubrimientos científicos, y considera el conjunto de innovaciones como un proceso espontáneo.

Esta confusión de saberes ha tenido repercusiones estratégicas 20 años más tarde. Un país como Estados Unidos con más ciencia básica que ningún otro, está siendo superado en áreas industriales claves por otro como Japón, con un nivel inferior de ciencia, pero con una tecnología cada vez más avanzada, y con una creciente voluntad de construirse como "technological country" (Hajime Karatsu. 1986), y con teóricos importantes de la "information society" como Yoneji Masuda (1980).

Por su parte en Europa, cuna de la ciencia, existe un potencial de crecimiento tecnológico en países como Alemania, Italia o España. (El caso español es materia de otro estudio, pero es interesante resaltar que la primer plan de I+D nace del impulso dado por parte de sociólogos de las nuevas tecnologías como Manuel Castells (1986), y con una clara orientación hacia la investigación "mission oriented".

La ciencia no es un saber "prefigurativo". Su aportación histórica, tal como Occidente la ha concebido de forma mayoritaria, ha sido el poder explicar, en palabras de Kuhn, "qué existe y qué es"(1987:97). "El descubrimiento de un pauta parcialmente oculta"(Herb Simon, 1973: 16). Es tarea de la tecnología, no de la ciencia, el diseñar como podrían ser éstas. El saber prefigurativo es el diseño. Los programas de investigación científicos para Lakatos pueden ser deductivos o inductivos, basados en conjeturas y refutaciones (1975) pero no programas de diseño de nuevos sistemas, tal como se practican en CMU.

Esa tarea le ha correspondido a la tecnología. Esta, de momento, se ha concebido, por la moderna cultura, como mera aplicación de una ciencia concebida a su vez con criterios de objetividad, lo cual ha sido un paso intelectual importante en la historia de la cultura humana. Una nueva visión de la misma puede concebirla como el campo de diseño e invención humano del conjunto de nuestra realidad cultural.

Denomino tecnoculturas o "design cultures" a las comunidades que consideran el diseño y la invención como su práctica cultural preferente, actividad encaminada no sólo a la construcción de máquinas informáticas y sus programas sino del

conjunto de su vida organizada, incluyendo su propia tecnología. Entiendo que este modelo cultural es una hipótesis no tanto sobre una estructura cultural existente sino sobre una posible. Al fin y al cabo, la invención de momento en las "computer cultures" está limitada al sistema mecánico principalmente. Salvador Giner viene utilizando también el término "tecnocultura" desde 1984 ("Tecnoculturas", La Vanguardia, 4 de febrero 1984). para definir una realidad social emergente producida por estas nuevas tecnologías. La incesante capacidad de innovación, no ya de objetos, sino de los propios conocimientos, la considera una de sus características principales. Es una realidad cultural que considera poco estudiada y que ya no puede entenderse con los tradicionales estudios sobre "los impactos sociales de la tecnología". Su relación con las técnicas por un lado, y el saber científico por otro es problemática.

Por otra parte, Karl Popper ya sostuvo en los años 30s, y en nombre de la libertad contra el historicismo, la posibilidad de una "ciencia social tecnológica", una metodología para "un estudio de las leyes generales de la vida social, cuyo fin sería el de descubrir todos aquellos hechos que habría que tomar en cuenta todo el que quisiera reformar las instituciones sociales". (1987:60).

En antropología, los intentos de una antropología aplicada han sido conflictivos. Es conocida la historia de críticas que sufrió el proyecto Vicos de la Cornell University, dirigido por el antropólogo Allan Holmberg (1955,1956,1958), realizado en plena Guerra Fría y acusado de proyecto "colonialista" (Stavenhagen, 1971). A su vez, todo plan de reforma social era visto por el otro bando, de hacer el juego al comunismo.

En el nuevo periodo abierto tras la Guerra Fría, hay nuevas posibilidades de situar a la ciencia social al nivel tecnológico alcanzado por otras disciplinas y abrir una metodología tecnológica y de diseño en campos como la antropología o la sociología, que incluya un posible diseño de nuevos valores, instituciones y pautas de actuación de los nuevos sistemas culturales en nacimiento. En parte, esta línea ya está iniciada (Lasswell, 1971).

En los años 40s, el antropólogo Melville Herskovits propuso un giro en su libro "El hombre y sus obras", publicado por primera vez en 1948, para entender la mentalidad inventora que nos parece que ayuda a ver de forma distinta esta nueva realidad cultural: "Es menester un giro absoluto en la mentalidad euroamericana de hoy si hemos de considerar que el proceso de invención actúa sobre la

totalidad de la cultura, y no sólo sobre sus elementos tangibles. Con la mayor facilidad propendemos a descuidar el papel del inventor de nuevas ideas y de nuevos conceptos, descuidando la función que desempeña en los cambios que señalan el desarrollo histórico de toda cultura..." (Ed. cast.1981:536)

Siguiendo a Roland Dixon, profesor de antropología en Harvard, ("The Building of Cultures" 1928), Herskovits entendía por invención la "creación deliberada de alguna cosa radicalmente nueva", actividad distinta del descubrimiento , el hallazgo accidental de algo que no se había observado previamente.

"El uso común de la palabra 'inventor' ilustra muy bien el prejuicio euroamericano. Un 'inventor' es una persona que 'inventa' una nueva máquina, o un nuevo método mecánico. Quien desarrolla ideas para un nuevo sistema económico, o imagina un nuevo esquema político, o elabora una nueva concepción del universo, no es, para nosotros, en modo alguno, un inventor. Podemos llamarlo un teórico, un filósofo, un visionario o , más despectivamente, un revolucionario.

Sin embargo, las ideas son ciertamente no menos poderosas que las cosas en la formación de las vidas de los hombres. Sería difícil sostener que los 'inventores'-y la palabra se usa aquí en su sentido etnológico y funcional- que imaginaron contar la descendencia por un lado de la familia, o quienes más tarde desarrollaron sistemas de clasificación de la terminología del parentesco, tuvieron menos influencia en el curso de la cultura humana que tuvo el inventor de la tienda de pieles, o de la canoa batanga, o de la bomba y los fuelles usados en el trabajo del hierro". (1981:536).

Por su parte, Margaret Mead dentro de los trabajos por una Anthropology for the Future, afirmaba: "We are in a period of invention, of dealing with many concepts that we never had to consider before, and recognizing that one of the tasks of anthropology is culture building". (1978: 272)

En las "computer cultures' el mundo se empieza a ver como un "artificio colectivo" (H. Simon. 1973: 17) , como un "man-made world" (entendiendo por "man" ser humano genérico). Antropólogos como Clifford Geertz, poco proclives al estudio de la tecnología, ya habla de la ideación como un "cultural artifact" ("Local Knowledge. 1983:152), tomando el lenguaje propio de los investigadores en Inteligencia Artificial.

El invento en esas culturas ya no se reduce a la máquina. Como afirma Herb Simon: " Moreover for most of us--the white -collared ones-- the significant

part of the environment consists mostly of strings of artifacts called 'symbols' that we received through eyes and ears in the form of written and spoken language and that we pour out into the environment..The law that govern these string of symbols, the laws that govern the occasions on which we emit and receive them , the determinants of their content are all consequences of our collective artifice" (1981: 5).

Las "computer cultures" nos permiten apreciar que el fenómeno de invención mecánica es una invención de símbolos y programas de símbolos. Las tecnoculturas desarrollan y complejizan las "computer cultures" al no reducir la invención al sistema informático, o a un sistema simbólico lógico-matemático. El diseño no tiene por que reducirse al diseño sólo de máquinas o sistemas mecánicos o su lenguaje de programación. Puede incluir también el de instituciones humanas que los usan (familiares, escolares, de entretenimiento,...) , los nuevos sistemas de conocimientos que los desarrollan, los nuevos ritos y prácticas sociales que rodean al ordenador y que no son codificables en lenguaje máquina y procesables por el ordenador, pero sin las cuales el ordenador no hubiera nacido.

Una tecnocultura es una cultura de invención de nuevos conocimientos, ante todo, de nuevas disciplinas tecnológicas. Así como se concibió la Inteligencia Artificial, como una síntesis de matemáticas, ingeniería, psicología cognitiva,etc. concebida para hacer más inteligentes a las máquinas, el profesor Frederik Reif de CMU concibe una "human cognitive engineering" (1980: 44) que pueda mejorar las capacidades de los humanos. De hecho, éste es el objetivo de la educación. Y es en la educación donde en CMU se están desarrollando modelos más avanzados de diseño. Las tecnoculturas son culturas de invención general, y esta invención considera la propia educación humana su campo privilegiado de acción.

Basándome en este enfoque conceptual, analizaré en este estudio, los proyectos de la School of Computer Science de CMU como procesos de formación de una "computer culture".

3. DESCRIPCION DEL OBJETO DE ESTUDIO.

3.1. CMU COMO CULTURA DE INNOVACION.

Los dos motivos iniciales para la elección de CMU como base de la investigación fueron: 1. El constituir un primer experimento de "computer-intensive campus". 2. El ser un centro de investigación generador de la idea de "The Sciences of the Artificial", libro escrito por Herb Simon en 1969 , donde introduce la idea de mundos "naturales" y "artificiales" como mundos distintos, basados los últimos, en el diseño humano. Estas dos características son únicas de CMU como "computer culture".

Segun Herb Simon, profesor de sicología cognitiva y computer science, existe un tipo de ciencias, distintas de las naturales, que denomina "sciences of the artificial" que se basan en la idea de diseño. A diferencia de la física o la química dedicadas al estudio de qué son y cómo funcionan los "natural worlds", las ciencias de lo artificial, como la ciencia de la administración, la economía, la sicología y las profesiones como la ingeniería tienen un terreno distinto: "Engineering, medicine, business, architecture, and painting are concerned not with the necessary but with the contingent--not with how things are but with how they might be--in short with design". (1981:xi).

La computer science es una expresión de este tipo de ciencia. Estudiar la cultura de CMU implica ante todo analizar que es este tipo de conocimiento y la gente que lo ha producido.

Esta distinción entre "natural worlds" y "artificial worlds" tiene relación directa con la cultura de CMU como cultura de innovación.

Tradicionalmente, en la cultura occidental el mundo de la ciencia natural ha sido el mundo de los fenómenos necesarios.

Yehuda Elkana, en su "Sciences and Cultures"(1981), ha puesto de manifiesto la conexión entre la ciencia griega , su visión de lo inevitable y su relación con el teatro trágico: " Greek drama is a depiction of the inevitable. Fate is immutable, and man can influence minor details of his own destiny...It is an old Western cultural tradition to view the growth of knowledge-knowledge of all kinds, even scientific knowledge- as the subject of Greek drama: the unfolding of the inevitable". (1981:66).

Esta visión concuerda con la doctrina de la predestinación del calvinismo protestante y permite explicar, como hizo Robert Merton, el auge de la ciencia, en especial la ciencia natural, entre el protestantismo ascético, tanto en Inglaterra como en las primeras colonias norteamericanas.

El método de estudio clásico de la ciencia natural sería una combinación de racionalismo y empirismo: "predicciones empíricamente confirmadas y lógicamente coherentes" (Merton, 1942: 67).

Daniel Bell ha culminado esta visión afirmando que la "racionalidad funcional" es el principio axial del orden tecnoeconómico contemporáneo. (1976:24).

Pero esta visión tradicional del mundo, como "orden natural", explicado racionalmente por la ciencia, esta siendo modificada por un "orden artificial", inventado por la tecnología de forma incesante.

El presente paradigma científico ha admitido la tecnología y su cultura de invención como un complemento secundario de la actividad científica. Es el modelo "Science and Technology" que forma, con la industria, el denominado "unique innovation engine" (Dertouzos.1988: 22).

Según este esquema es la Ciencia la que inicia el ciclo de innovación, lo aplica la Tecnología y lo produce la industria. Pero en este esquema hay una anomalía. La llamada "innovation" no ha sido un valor cultural principal de la ciencia sino de la tecnología. Para Merton, el valor principal de la ciencia es el "universalismo", para Bell, el principio axial del orden tecnoeconómico es la "racionalidad funcional". Ninguno de ellos asocia ciencia, ni siquiera tecnología, a innovación. Más antiguamente, la escuela de Frankfurt habló de "racionalidad instrumental", como la "ideología" del "unidimensional man"(Marcuse). Veían al tecnólogo como un burócrata controlador de procesos naturales. Su valor cultural clave era el "control" (Beniger, 1986), y así veían a los ordenadores, como las herramientas del burócrata.

Según este esquema, la secuencia normal de innovación en las culturas urbano-industriales o "sociedades complejas", partiría del "descubrimiento científico" desde las clásicas ciencias naturales, realizado en la Universidad, continuaría con la invención tecnológica a cargo principalmente de los laboratorios industriales y terminaría en la producción de objetos manufacturados para el mercado realizado en las industrias. Es interesante destacar que el conjunto del proceso se denomina "innovation cycle" y en él la ciencia, aunque se considera que inicia el proceso, actúa dentro de un sistema

cultural presidido por un valor cultural ajeno a ella.

En realidad este modelo, éste "innovation cycle" no empezó en una Universidad sino en un instituto de tecnología: el Massachusetts Institute of Technology. En este instituto nació el concepto de TECNO-LOGIA, por el cual la ciencia debía servir para aplicaciones útiles, y la técnica se convertía a cambio en un procedimiento científico, sistemático y formalizado.

En realidad el M.I.T. nació buscando independizar la tecnología de la ciencia. Empezó como un centro investigación tecnológica básica. Más que como una ciencia aplicada, el modelo empezó dotando a la tecnología de estatus académico, esto es, convirtiéndola en "básica", elevándola de categoría intelectual. El M.I.T pasaba a competir con Harvard, el centro espiritual de la cultura liberal de Estados Unidos.

Fue la cultura tecnológica del M.I.T la que produjo el fenómeno de la cibernética en los años 40s, reconocido por Simon y Newell (1972. :878) como la base intelectual, el Zeitgeist, donde se desarrollaría el resto de la cultura informática, y de la propia ciencia cognitiva. Esto es sólo parcialmente cierto.

Una aportación clave provino del grupo de matemáticos de Princeton: Church, Turing y Von Neumann.

De la cibernética se pasó al "information processing". Dos científicos de Carnegie Tech se adelantaron en 1955 a los del M.I.T como Minsky y McCarthy, inventando el primer sistema artificial inteligente. Este ha sido, hasta el momento, el acmé o cénit de CMU: en la invención de la tecnología más avanzada en el campo informático: la Inteligencia Artificial se pusieron por delante de la primera institución tecnológica de Estados Unidos, el M.I.T.

El modelo de CMU es un modelo nacido en un Instituto de Tecnología, como una teoría dentro de un Zeitgeist denominado "cybernetics revolution" (Newell, Simon,1972) con un carácter estructural tecnológico.

No obstante CMU ha aportado a esa cultura rasgos propios:

1. La "computer culture" de CMU es un fenómeno tecnológico, que se apoya en una de las ramas de la computer science, la Artificial Intelligence, donde tiene un liderazgo destacado un científico social, H. Simon. La Computer Science, y en particular la AI, se ha considerado para algunos líderes de la comunidad como una aplicación de una disciplina más amplia denominada "Complex Information Processing", que abarcaría tanto ordenadores como mentes humanas. El modelo del ordenador era un "model of man" definido por H. Simon como de "racionalidad

limitada". (bounded rationality). Este modelo fue elaborado por un científico político a partir de una reflexión sobre el "comportamiento administrativo" iniciada en los 40s. Este papel decisivo jugado por la AI dentro del Computer Science de CMU , y dentro de ella el de un científico político, más tarde convertido en sicólogo, es un rasgo específico de esta cultura, que la distingue de otras como el MIT o Stanford.

2. Al pivotar sobre la Artificial Intelligence esta cultura se ha apoyado principalmente en la única institución que durante décadas ha financiado una investigación básica en AI: el Defense Advance Research Project Agency (DARPA). La misión estratégica de esta institución era doble : promover la investigación avanzada para la defensa nacional de Estados Unidos y secundariamente y transferir a la industria dichas tecnologías para aumentar la productividad económica nacional. La investigación en Computer Science de CMU está basada en el modelo de un laboratorio estratégico de la Defensa, en particular "a DARPA Artificial Intelligence Laboratory", y por lo tanto al servicio de intereses estratégicos de Estados Unidos. Ello le ha permitido una mayor profundidad a su investigación básica de carácter tecnológico en áreas como la AI, en las que ni siquiera las grandes corporaciones (IBM, Apple...) se han atrevido a abordar, que la National Science Foundation solo ha financiado tardíamente.

3. La School of Computer Science (SCS) de CMU ha sido un centro principal de investigación de esta universidad, y de donde han salido innovaciones principales de esta "computer culture".

Al mismo tiempo, es uno de los centros generadores de la "computer culture" en Estados Unidos en el terreno de la investigación básica. Es uno de los tres centros ("hubs") estratégicos en el país para el desarrollo de la investigación básica y aplicada en el campo de los ordenadores, y particularmente de la Inteligencia Artificial.

Los otros dos históricamente han sido el Massachussets Institute of Technology, (MIT), y la Stanford University de California.

Durante los últimos 25 años, DARPA ha sostenido ininterrumpidamente la investigación básica en ese departamento en áreas estratégicas como la Inteligencia Artificial, Programming Systems, Computer Systems , que están teniendo aplicaciones y relaciones con otras áreas y disciplinas (ciencias, management, humanidades, ingenierias, educacion, etc.).

3.2. LA INVESTIGACION DEL COMPUTER SCIENTIST DE CMU: "OUR SUCCESS IS INNOVATION".

"We don't have 300 years like Harvard. Our success is in innovation". Estas palabras de Herb Simon (Entrevista, 1 Mayo 1990) marcan el patrón cultural en el que desarrolla la actividad de investigación del "computer scientist" de CMU.

La innovación en esta "computer culture" parte, fundamentalmente, aunque no de forma exclusiva, de lo que antes se denominó Computer Science Department y hoy School of Computer Science de CMU y de su investigación. Esta escuela parece tener características culturales particulares, en parte heredadas de la institución a la que pertenece, Carnegie Mellon, y en parte, propias y adoptadas posteriormente por el conjunto de la universidad.

Una de ellas es el haber nacido no de un departamento universitario o de una disciplina científica tradicional, sino de la confluencia de tres factores: En primer lugar, una actividad de invención en 1955 de un programa de Inteligencia Artificial, denominado LOGIC THEORIST (LT) y un lenguaje de programación nuevo INFORMATION PROCESSING LANGUAGE (IPL), por el equipo interdisciplinar formado por profesores como H. Simon, estudiantes de postgrado como Allen Newell y programadores como Cliff Shaw, equipo con base en dos instituciones, la propia universidad, que en 1955 era aún Carnegie Institute of Technology, y la Research and Development Co (RAND), el original "think tank" inventado en Estados Unidos y financiado con fondos de la Defensa. Esta invención formaba parte de un nuevo programa de investigación de este equipo, iniciado en febrero de 1952, destinado a comprender lo que denominaban "complex information processing systems" (A. Newell, H. Simon "The Logic Theory Machine". 1956).

En segundo lugar, la apertura un Computation Center en dicha universidad en 1956, formado por Alan Perlis, director del Departamento de Matemáticas, y basado en una máquina, una IBM 650. Este centro fue concebido como servicio al conjunto del campus, como un lugar abierto de encuentro de matemáticos, ingenieros, físicos, estudiantes de management de todo el campus, que utilizaba el ordenador para resolver problemas de su especialidad.

En tercer lugar, la formación en 1962 de un programa interdisciplinar de doctorado, denominado Systems and Communications Sciences, con participación de cuatro departamentos: Mathematics, Psychology, Electrical Engineering y Graduate School of Industrial Administration, precursor de lo que en 1965 será el programa de doctorado en Computer Science.

Estos rasgos responden a una cultura académica, la de Carnegie Mellon University, entonces llamada Carnegie Tech, que se había distinguido por dos rasgos: 1. Su "problem solving mentality", su mentalidad pragmática, propia de una escuela técnica, expresada en el lema de su fundador Andrew Carnegie en 1900: "Handication versus Headication". Y 2. Su concepción de la educación para los ingenieros de carácter liberal, que debía integrar también inseparablemente el estudio de humanidades y la ciencia básica. Ambos rasgos culminarían en el llamado Carnegie Plan, en los años 40s, un plan para formar ingenieros con mentalidad interdisciplinar científico-humanística, y al mismo tiempo con dominio de los métodos de "problem solving", propios de la ingeniería. Desde su fundación en 1965, el Departamento de Computer Science de CMU ha propiciado una actividad de investigación interdisciplinar basada en un modelo que al parecer se comparte como valor común de la escuela: el enfoque pragmático y diseñador del ingeniero, y la formación científica y humanística, complemento del anterior. Este modelo cultural se ha definido como "scientist of design" (Herb Simon). El modelo educativo se denomina: "liberal-professional education". Ello conlleva un tipo de investigación específica en la que se combina la innovación tecnológica, que está en el puesto de mando, con una dimensión social y humanística, que le sirve de complemento.

A estos dos rasgos se unió un tercero, no existente en la antigua Carnegie Tech: la investigación estratégica, introducido por la cultura de la RAND Co. Carnegie Tech se convertirá en un centro de investigación para los intereses estratégicos nacionales norteamericanos en materia de tecnología informática. De centro de educación pasará a ser principalmente una unidad de investigación, y de servir a la industria local o nacional, pasará a servir, principalmente, intereses estratégicos de Estados Unidos en el campo de la investigación informática. Este pensamiento estratégico, retomado a través del management, le servirá para constituirse en los años 80s, mediante un ejercicio de "strategic planning" en la primera "computer-intensive university" de Estados Unidos.

Esta combinación de rasgos constituye el patrón cultural hegemónico, el modelo de pensar y de actuar dominante en esta comunidad, y le proporciona un conjunto de ventajas comparativas sobre otras "computer cultures".

Este patrón en CMU ha ido unido a una doble actividad: 1. La tendencia a sintetizar los dos primeros procesos de innovación tradicionales, el descubrimiento científico y la invención tecnológica, en una única e inseparable

actividad, liderada por el avance tecnológico. Y 2. La introducción del conjunto de toda la comunidad universitaria, no sólo a un sector de la misma en la actividad de invención tecnológica, concretada en la producción de sistemas informáticos, en particular, software. Sus líderes, incluso provenientes de ciencias sociales, se dedican a producir tales sistemas sin los cuales no serían reconocidos como tales en una comunidad de base ingeniera como ésta.

Si consideráramos Carnegie Mellon como un prototipo de una "tecnopolis" apreciaríamos que su modelo de innovación es más complejo que el descrito por el tradicional ciclo de CIENCIA-TECNOLOGIA-INDUSTRIA. En este ciclo, que podríamos denominar "simple", la innovación la inicia sólo un sector de la población investigadora: los científicos y técnicos, dejando fuera al resto de disciplinas (ciencias sociales, humanidades,...) y de conocimientos (arte, management,...). Y la culmina sólo un sector de la sociedad: la industria. Se deja fuera al resto de instituciones: la Administración, la enseñanza, la sanidad, las administraciones locales,... simples usuarios de las anteriores invenciones tecnológicas.

Por contra en el ciclo "complejo" de CMU, en tanto que COMPUTER CULTURE, la invención tecnológica trata de comprometer al conjunto de disciplinas, a las que se trata de embarcar en una actividad de diseño de programas informáticos, propia de la tecnología ("ciencias del diseño") y es el conjunto de la institución la que se pretende que participe en el proceso de innovación, entendido como un proceso de planificación estratégica.

El iniciador de la innovación es una combinación de político-computer scientist (Herb Simon, A. Newell, A. Jordan), más que un mero científico -industrial.

Según el modelo de CMU, la investigación ha de ir unida al diseño y construcción de sistemas informáticos, no sólo a la mera explicación de un fenómeno.

La teoría se elabora construyendo los programas y sistemas informáticos. Una construcción matemática de un hipotético sistema, por ejemplo, tendrá tanto más valor en la SCS de CMU si va unida a la construcción de nuevo programa informático que funcione, que "valide" dicha teoría. La "computer science" se sirve así del conocimiento matemático para construir sistemas computacionales.

Pero no se reduce a matemáticas, ni básicas ni aplicadas, ni a otra ciencia clásica. Se considera una disciplina distinta, en palabras de Mary Shaw, veterana profesora de la SCS : "a separate discipline" Great Ideas, april 1991) .

Este particular enfoque de la SCS hace que la clásica relación entre ciencia y tecnología, entre "investigación básica" e "investigación aplicada" aparezca invertida y unificada. En documentos fundacionales del centro, se afirma así que es de la segunda de donde salen principalmente las nuevas ideas. "The new ideas and insights that move it towards being a science arise mostly from applications rather than from any self-generative character of the pure science of information processing"(A. Newell, A. Perlis, Schatz, E. 1964: 3).

Los fundadores de la disciplina en esta institución (Perlis, Simon, Newell) han operado así desde el principio, construyendo sistemas, en particular "programas" (Logic Theorist, GPS, lenguajes informáticos...), que validaban su teoría del "information processing", asentando un hábito cultural característico, un "particular approach" de esta escuela : "the pragmatic approach". (ib:13).

La computer science nació en CMU en los años 60s como "the attempt to produce a science in an area that not long ago was a purely technology" (ib : 1). " The scientific knowledge is contained, not in a formal theory, but in the knowledge, however expressed, of the structure of a class of information processing tasks and the programs that can do that". (ib: 7).

En esta escuela, la tarea teórica del científico se indentifica con la de hacer programas. Un ejemplo de ello es SOAR el último programa informático de Allen Newell , resultado de casi 30 años de trabajos en Inteligencia Artificial, considerado no un mero vehículo para validar una teoría sobre la cognición humana sino la propia teoría en sí misma: "The effort internally (at home, so to speak, was to convert Soar to a cognitive theory" (A.Newell.1990:x).

Por tanto, las dos primeras partes del ciclo Ciencia-Tecnología aparecen sintetizadas en una nueva disciplina: "Computer Science", en la que se reconoce que la tecnología va por delante en innovación respecto a la propia ciencia . Allen Newell explicaba en el reciente XXV Aniversario del Departamento, celebrado en septiembre de 1990, esta paradoja. A diferencia de como otras ciencias como la microbiología o la genética habían nacido , a partir del descubrimiento de algún fenómeno natural (micro-organismos o genes), el Computer Science nació de una tecnología : "Its emergence has been especially confusing because of the computer's origin as an engineered and manufactured device " (Newell,1990 :41).

Esta primacía de la invención tecnológica sobre el descubrimiento científico da lugar a un proceso de innovación distinto, en el que las iniciativas provienen

de las nuevas invenciones tecnológicas que son teorizadas científicamente a posteriori o simultáneamente, formando ambos aspectos un conjunto inseparable, pero liderado por el avance tecnológico. Este rasgo constituye uno de los fenómenos característicos de la "computer culture" en CMU, institución que fue primero una escuela técnica (1900), luego un instituto de tecnología (1912) y sólo recientemente (1967) se convirtió en un tipo particular de universidad, un "computer-intensive campus" (1980s). Y su futuro está abierto.

Este proceso intelectual de síntesis entre investigación básica y aplicada ha precisado a su vez de un marco organizativo nuevo: el Computer Science Department, hoy ya School of Computer Science (SCS).

Como la propia disciplina, este es un departamento universitario distinto a los restantes. Inicialmente fue concebido como un "open center" (Newell, Perlis, Schatz. 1964: 25), un centro abierto a todas las disciplinas y departamentos de la universidad, con la idea de que "A very substantial segment of Carnegie Tech should become the laboratory for the study of information processing" (ibid. :26).

Esta institución se ha ido constituyendo como una nueva institución experimental, y con un estilo de organización informal, síntesis de diferentes disciplinas (matemáticas, ingeniería, psicología, ...) y de profesores provenientes de estos departamentos, siendo a su vez distinto a éstos. En 1965, se fundó éste fuera tanto del departamento de ingeniería eléctrica como del de matemáticas dentro del College of Engineering and Science, siendo el primero que lo hizo en Estados Unidos. Una vez que el College of Science en CMU se separó del de Ingeniería, el departamento se quedó en el colegio de ciencias, de donde se separó en 1985 y en 1988 se constituyó como escuela con rango equivalente a los otros colegios de CMU.

Por último, el modelo de relación de esta institución con el mundo industrial ha sido más compleja que las de la clásica de las escuelas de ingeniería, y descrita en el ciclo Ciencia-Tecnología-Industria. La industria del ordenador ha jugado y juega aun un papel secundario en el apoyo de la investigación en la SCS.

Secundario pero significativo.

La entidad que ha sostenido y sostiene aún fundamentalmente estas investigaciones, de "alto riesgo" (high risk) dado que su resultado no está garantizado y es a medio o largo plazo, en CMU es una institución pública de carácter gubernamental: la Defense Advanced Research Projects Agency (DARPA).

Esta es una institución particular. A pesar de ser pública, su carácter desde el principio fue financiar proyectos tecnológicos avanzados, teniendo un gran carácter innovador.

Gracias a su prolongada y sostenida inversión y apoyo durante los últimos 30 años, ese tipo de investigación nueva mitad ciencia-mitad ingeniería y con objetivos públicos no privados al servicio directo de la empresa, se ha podido mantener . En 1988, la proporción entre la investigación esponsorizada por el gobierno y la sostenida por la industria en el Departamento era 85%:15%.

La SCS forma parte del pequeño grupo de seleccionados departamentos que recibe cerca de 10 veces el promedio de financiación en comparación con otros departamentos de similar tamaño en Estados Unidos. (Goals and Strategic Plans. CMU. 1988).

Ello ha permitido realizar en la SCS un tipo de investigación estratégica en áreas que no han tenido aplicación comercial inmediata, y que la industria no se ha atrevido a sufragar dado su alto riesgo : Inteligencia Artificial, Vision y Reconocimiento de Voz, redes electrónicas, circuitos integrados, robots autónomos, ... Una vez que han producido resultados las propias industrias, no sólo norteamericanas sino japonesas y europeas, se ha beneficiado de ellos. Igualmente se han aprovechado las otras universidades, la Administración, y todo el complejo mundo formado en torno al ordenador, y sobre todo, el propio Departamento de Defensa.

No obstante, es preciso reconocer que fue una industria, la IBM , la que patrocinó el Proyecto Andrew, que fue la base de conversión de CMU en una "computer-intensive university". Y que fue la fundación de un banquero, Richard K. Mellon, la que aportó 5 millones de dolares iniciales para la constitución del Computer Science Department en 1965.

3.3. LA SCHOOL OF COMPUTER SCIENCE , CENTRO DE LA INVESTIGACION DE ESTA "COMPUTER-INTENSIVE UNIVERSITY".

Esta escuela es una escuela de postgrado (graduate students), su objetivo es formar científicos del ordenador de alto nivel investigador (highly-qualified research computer scientists.). Pero por encima de todo, su meta es liderar la investigación en "computer science" a escala mundial: "The overall goal of the Computer Science Department is to maintain its worldwide leadership role in computer science research" (Goals and Strategic Plans. CMU. Feb. 1988: 28). El

primer objetivo de la institución es, pues, la investigación ("research") y subordinado a éste la educación ("education") de investigadores en computer science. Al menos así ha sido hasta ahora.

La estructura oficial de la escuela esta formada por un Programa de Doctorado en Computer Science ("Doctoral Program in Computer Science"); un programa de Master para formar ingenieros de software ("Software Engineering Master's Program"), realizado en colaboración con el Software Engineering Institute. Y por último, tres institutos: Robotics Institute, (que tiene su propio programa de doctorado), el Machine Translation y el Information Technology Center, nacidos de la propia investigación de la escuela y con una estructura organizativa independiente. Todo ello bajo la dirección del decano ("Dean") de la SCS, Nico Habermann, que en 1991, pasó a un nuevo cargo en la National Science Foundation. En 1989, había 193 estudiantes graduados en la School (77 en computer science y 17 en robotics). Ese curso se doctoraron 14 estudiantes de un total de 350 Ph. D. en todo el país. La admisión de estudiantes en la escuela es altamente selectiva. En 1987, se recibieron 900 solicitudes de todo el mundo, siendo admitidos unos 30 estudiantes, de los cuales un tercio aproximadamente estudiantes extranjeros.

A efectos educativos, el programa de doctorado en Computer Science se divide en cuatro áreas ("areas") : Teoría, ("Theory"), Sistemas de Programación ("Programming Systems"), Sistemas de ordenadores ("Computer Systems") e Inteligencia Artificial ("Artificial Intelligence"). El estudiante deberá realizar un examen escrito en cada una de ellas "core qualifiers" para poder realizar la tesis doctoral.

Esta estructura docente colabora con la estructura de investigación que constituye la actividad principal de la SCS. La valoración en esta escuela está en función directa de la capacidad investigativa del computer scientist. "The potential for outstanding ability in research is always the primary criterion" (Graduate Studies in CS. CMU.)

La escuela cuenta con 108 profesores e investigadores que forman el núcleo investigador clave de la misma. En 1989 eran 77 de Computer Science, 27 en Robotics y 3 en el Machine Translation. Del total, 36 eran "regular faculty" o profesores con derecho a plaza, 35 "research faculty" y 37 "special faculty" (investigadores especiales, profesores visitantes,...). Un figura clave para entender un aspecto que distingue esta escuela de otras es el papel de los

"programmers". En 1990-91, la escuela contaba con 91 progradores, de ellos 68 en el area de Computer Science. Ellos son los responsables de escribir los programas informáticos para el ordenador.

En total se calcula que en 1991 la School abarcaba entre profesores, estudiantes de doctorado, personal de investigación, personal técnico de apoyo y personal administrativo, unas 700 personas (XXV Anniversary.:vii)

Cada profesor trabaja según diferentes "intereses" ("interests") en las diferentes áreas arriba indicadas, y en determinados "proyectos de investigación" ("projects"). Las principales innovaciones de la escuela se han producido en la investigación . Esta es la razón para seleccionar " los proyectos de investigación" como unidad de análisis etnográfico básica .

Los proyectos se agrupan en grandes programas de investigación que son los esponsorizados por DARPA (Defense Avanced Research Projects Agency). En 1990 son cuatro: "Basic", "Strategic", "Speech" y "ERGO". Estos programas forman la columna vertebral de la investigación de la Escuela. Agrupan a sus mejores equipos y los mayores presupuestos.

Los proyectos, al mismo tiempo, son el campo de trabajo del estudiante de la SCS , dado que durante los 4-6 años que dura su doctorado, la mayor parte transcurra trabajando en algún proyecto de investigación dirigido por su director de tesis ("advisor"). Muchas de las tesis doctorales son productos del trabajo en el proyecto.

He seleccionado cuatro áreas, "Artificial Intelligence", "Programming Systems", "Computer Systems" y "Theory", y distintos proyectos en ellas, que reúnen algunas de las características que considero valores culturales propios de la School:

1.) la "construcción de sistemas" ("building systems"), como expresión de la investigación en "information processing systems", teoría básica de la computer science en CMU ; 2,) un enfoque holista a hora de construir dichos sistemas , ("total system effort") abierto al trabajo interdisciplinar y de innovación entre distintas áreas de la computer science y entre esta disciplina y otros departamentos de CMU considerados como laboratorios para el estudio del " information processing"(open center) ; 3.) un estilo de trabajo que promueve la formación de equipos (teamwork), y 4) una estructura informal en la escuela con menor compartamentalización que en instituciones de similar tamaño y organizada en redes (network).

El objetivo central de esta investigación es describir el o los patrones

culturales de los profesores y equipos que forman estos proyectos de investigación, y de las metas de finalidad de los propios proyectos como unidades de estudio. Este patrón o modelo etnográfico estará compuesto por el sistema de valores de dichas personas, sus relaciones jerárquicas, sus relaciones con otras instituciones en la comunidad universitaria de CMU y en el resto de organizaciones del país, tanto industriales como político-administrativas, y sobre todo un análisis de como se están construyendo estos patrones a través de los proyectos de investigación de dicha escuela.

Siguiendo el análisis del ethos de las comunidades científicas establecido por Robert K. Merton (1942), y que para Daniel Bell dicho "ethos" de la ciencia, constituye "el ethos emergente de la sociedad postindustrial" (1973), la pregunta es: ¿es éste el ethos de una high-tech community como la de computer scientists de CMU, en concreto de los investigadores en estudio? ¿En qué coincide y en qué difiere? ¿Qué valores o metas de finalidad tienen estos, cuál es el sistema de valores de los profesores que construyen los ordenadores, los programas, las aplicaciones a través de estos proyectos seleccionados?. En suma, como cualquier modelo etnográfico, el estudio ha de responder a estas cuestiones básicas: "Cómo son los individuos, qué es lo que quieren ser y cuáles son los caminos institucionalizados que deben seguir para lograrlo" (Claudio Esteva. 1978: 234).

4. HIPOTESIS DEL ESTUDIO .

4.1. PRIMERA HIPOTESIS: "KNOWLEDGE IS DESIGN" EN LA SCHOOL OF COMPUTER SCIENCE.

Supongamos que hemos de hacer un modelo etnográfico del saber de un "computer scientist" de CMU. ¿Qué programa cultural rige a los que hacen programas informáticos, a los "computer scientists" de CMU?.

Afirmamos que la principal meta de finalidad de estos "computer scientists" parece ser el aumentar su conocimiento sobre qué sistemas informáticos pueden construir y cómo hacerlo. Su meta es el diseño como sistema de conocimiento cultural básico de esta comunidad. Esta comunidad no persigue principalmente entender o explicar una realidad dada, ni siquiera un sistema tecnológico dado,

sino conocer qué nuevos sistemas informáticos pueden hacer. Por otra parte, el hecho de que acabe sabiendo cómo hacerlo, es la única vía para cubrir su primer objetivo. Para aumentar su conocimiento sobre qué nuevos sistemas tecnológicos pueden hacer, aprenden a hacerlos, adquieren lo que denominan know-how, o saber-como-hacerlos.

El hecho de que acabe construyendo prototipos, o demostraciones ("demo") de sistemas es un medio para lograr su objetivo central : la creación de nuevo conocimiento sobre nuevos sistemas informáticos posibles.

Por "sistema informático" se entiende en la SCS los sistemas construidos en Artificial Intelligence, Programming Systems, Computer Systems, y Robotics . La Theory es la área encargada de estudiar los posibles sistemas informáticos a construir, a nivel formal.

El proceso de innovación en los programas estratégicos de la Escuela, Vision, Speech, Supercomputers,... parece empezar no por el descubrimiento de un objeto empírico sino por el diseño de un nuevo conjunto de objetivos a conseguir ("goals"), que describen el nuevo "sistema", "programa", o robot, a construir.

La lógica del diseñador de "computer science" de la SCS , tal y como se expresa en los documentos de los proyectos ("proposals"), indica que lo primero son los "goals", los objetivos que identifican el nuevo sistema que el equipo de computer scientists quiere construir y que aún no existe. Y que es imposible por tanto descubrir a priori y estudiar como objeto empírico.

Estos objetivos, sin embargo, no se establecen "ex nihilo". Se apoyan en elementos tecnológicos y científicos ya existentes. Pero constituyen una realidad cultural nueva, existente tan sólo como "proyecto".

En el apartado denominado "technical rationale" de sus "proposals" , documentos de base de sus proyectos, es donde exponen los objetivos nuevos en forma de nuevos sistemas que se pretenden construir: Nectar, Mach, Soar...

En esto podrían asemejarse a los ingenieros y a los tecnólogos tradicionales. Como ellos pretenden construir, diseñar nuevos sistemas. Pero estos sistemas son instrumentales para su finalidad principal: aumentar su saber a cerca de los mismos. Su fin principal es el nuevo conocimiento, no sólo el nuevo sistema. Ese rasgo cultural les diferencia de la ingeniería tradicional : no se consideran aplicadores de un conocimiento básico encarnado en las ciencias clásicas (matemáticas, física, química, biología), sino que se consideran los elaboradores del nuevo conocimiento que ellos mismos aplican. Son también

teóricos, construyen su propia teoría mediante la construcción de sus proyectos.

No se trata sólo de tecnología, sino "high technology", donde el término "high" designa el peso del componente teórico o "científico" de la misma.

Su programa cultural tiene una doble característica: la finalidad principal es la construcción de una teoría del nuevo artefacto informático posible. El medio es su propia construcción. Así, por ejemplo, se está formando ahora en esta escuela una nueva disciplina informática, Robotics Science, al calor de la construcción de sistemas como Ambler o Navlab, robots autónomos. Esta nueva disciplina está formada por la síntesis de Machine Intelligence, Mechanical Engineering y un nuevo campo denominado "Computer Vision", que estudia los sistemas de identificación de la imagen del robot.

¿ En qué se diferencian, pues, de un científico tradicional?. En que las nuevas disciplinas nacen no del descubrimiento de un objeto real preexistente a la propia observación científica, mediante la observación empírica en combinación con el razonamiento, sino que surgen de una invención tecnológica que realiza el propio computer scientist y que una vez construida, la toma como objeto empírico de estudio para seguir construyendo nuevas realidades artificiales. Estas disciplinas son "conocimientos artificiales", sintetizados, nacen de la actividad de construir artificios y sirven para producir nuevos artificios, nuevos sistemas.

Como la ciencia clásica, el "computer scientist" busca crear conocimiento, pero a diferencia de ella, su conocimiento es inseparable de la construcción de artefactos pues se trata del conocimiento de esos mismos ingenios mecánicos.

Son "nuevos conocimientos", basados en una lógica de integración, de "putting it all together", la misma lógica con la que construyen sus máquinas.

En este sentido podríamos formular esta primera hipótesis: en la SCS de CMU, el objetivo fundamental del computer scientist es la creación de nuevos conocimientos sobre nuevos sistemas informáticos posibles. Actividad que exige inseparablemente la construcción de dichos sistemas, que se toman como objeto de estudio e investigación. En este sentido, en la SCS, "knowledge is design".

De ello se derivan dos corolarios fundamentales: 1. La inversión de conocimientos, subordinando el estudio empírico de los sistemas a su diseño y 2. La tendencia hacia la síntesis de nuevos conocimientos de diseño.

Desarrollémoslos.

4.1.1. PRIMER COROLARIO: LA INVERSION DE CONOCIMIENTOS.

El primer corolario podría formularse así: el estudio científico empírico de estos nuevos mundos artificiales del ordenador, tal y como se ve en estos proyectos de la SCS, no es ni puede ser previo a su diseño sino posterior. Ha de secundar con elementos analíticos el proyecto de sistema informático que se pretende construir, siendo inseparable de su construcción práctica. De haber seguido una lógica inversa, anteponiendo el proceso de descubrimiento científico de esos artefactos a su diseño e invención tecnológica, estos hoy no existirían. Este "científico" de CMU diseña, se propone construir realidades artificiales, en las que cuentan sus preferencias. El diseño parece ser su meta principal, su "core knowledge" .

Para construir sus artefactos o "desired states", realiza análisis empíricos, trata de encontrar patrones y modelos en las realidades que construye ("initial state") . Pero el análisis empírico de realidades artificiales en el ámbito del ordenador no puede ser previo a su construcción, sino posterior. Este es un fenómeno que no se ajusta al patrón clásico de la ciencia occidental, ni a la mentalidad cultural del "animal racional" aristotélico. Si nos basamos en Levi-Strauss podríamos compararlo inicialmente con el artista: " El arte se inserta, a mitad de camino, entre el conocimiento científico y el pensamiento mítico o mágico; pues todo el mundo sabe que el artista, a la vez, tiene algo del sabio y del bricoleur: con medios artesanales, confecciona un objeto material que es al mismo tiempo objeto de conocimiento". (1984:43).

Construye "modelos reducidos", como las pinturas de la Capilla Sixtina respecto al universo. Es algo "man made": "En la medida en que el modelo es artificial, se torna posible comprender como esta hecho" (ib:46).

El "computer scientist" se sirve de análisis empíricos de realidades que son previas a la existencia del ordenador, como la mente humana. Pero esos análisis son instrumentales al servicio de la construcción de una inteligencia artificial distinta a la del ser humano. Igualmente se sirve de sistemas formales, creados por los matemáticos o lógicos, antes de la existencia del ordenador, pero esos sistemas son también herramientas para construir lenguajes de programación distintos a aquellos sistemas matemáticos, de la misma forma que el computer

scientist se expresa a través del lenguaje simbólico natural sin por ello convertirse en poeta, en creador de este lenguaje.

Estos "científicos del diseño" han abierto las puertas , a través de la computer science, a un cambio de la clásica relación descubrimiento-invencción-descubrimiento en otra nueva, invertida : invención-descubrimiento-invencción. Esto es un fenómeno cultural nuevo , distinto a la clásica relación de saberes en la cultural occidental. A este fenómeno detectado en los proyectos analizados lo denomino: " una inversión cultural de conocimientos". Esto hace de la "Computer Science" de la SCS una "separate discipline", y una institución independiente.

La denominada "basic research " es en la "computer science" de CMU el estudio de un invento y de sus propiedades a fin de producir más inventos. El descubrimiento empírico es una función del diseño, no al revés.

Lo que se denomina investigación "básica" en este campo tiene un significado distinta a lo que se entiende por ella en la física u otras ciencias naturales.

Su propósito principal no es el descubrimiento de leyes en la naturaleza o la sociedad, sino el saber que nuevos sistemas computacionales son posibles y como poder construirlos. No es tampoco la estrecha investigación aplicada propia de la ingeniería tradicional: su finalidad no es meramente aplicada, de construcción de artefactos o "building systems", sino destinada a conocer cuáles se pueden hacer y luego hacerlos, como medio para poder ampliar el horizonte de los que se pueden hacer.

Lo que nos interesa es señalar un proceso que es el que vemos en la base del Computer Science, y que en CMU se ha reconocido en la figura del "designer": la tendencia de algunos científicos a hacerse diseñadores.

Pero el científico tradicional una vez convertido en computer scientist entra en conflicto con su ethos anterior : se pone en función del diseño de ordenadores, sus lenguajes, sus mecanismos, etc, en lugar de ser un científico dedicada a probar teoremas o a probar hipótesis sobre el mundo natural. Se ha pasado a mundo del diseño. Tiende a convertirse en un nuevo tipo de ingeniero.

De momento, algunos líderes de la generación de estos "científicos de lo artificial" definen a este tipo humano como un "Renaissance Man", una especie de ingeniero-científico-artista, todo al mismo tiempo, como Leonardo da Vinci (H. Simon. Entrevista Personal. 30/11/90).

Este fenómeno cultural se ve de momento como la síntesis de los patrones de la

cultura occidental . Por ejemplo, Herbert Simon, como científico de lo artificial se sigue viendo a sí mismo, ante todo, como un científico, un "animal de racionalidad limitada" y como tal con valores básicos universales . En esto sería un científico, siguiendo el patrón cultural de R. Merton. Pero al mismo tiempo, Simon ve al científico como un "problem solver". (1989:373). Esta síntesis es clara en el Dr. Dana Scott, lógico, matemático, y filósofo, y figura en la área de Theory en la SCS, internacionalesmente reconocida.

No obstante, Leonardo da Vinci no era fundamentalmente un científico como Galileo, sino un ingeniero renacentista, que gran parte de su esfuerzo intelectual lo dedicó a construir ingenios militares. La cultura de problem solving de CMU surgió de la ingeniería, más que de la ciencia o la matemática. Sin embargo, los computer scientists de CMU proceden crecientemente de culturas de matriz no occidental. Como es sabido, cada vez es más reducido el porcentaje de estudiantes norteamericanos en carreras técnicas y más los que proceden de Asia. H.T. Kung procede de Taiwan y no se define como científico sino como "problem solver". Takeo Kanade proviene de Japón. Otros líderes de la SCS proceden de la India, como Raj Reddy o de países europeos recién incorporados al tren tecnológico como España (A. Jordan) o de América Latina (J. Carbonell). Los profesores de la School of Computer Science unos pertenecen a la National Academy of Engineering como Raj Reddy y Angel Jordan , como lo fueron A. Perlis y J. Traub , antiguos directores del Computer Science Department. Otros, como H. Simon y D. Scott son de la National Academy of Sciences. Y sólo Allen Newell es de ambas.

4.1.2. SEGUNDO COROLARIO: LA TENDENCIA A LA " SINTESIS DE NUEVOS CONOCIMIENTOS DE DISEÑO".

Dicho mundo artificial avanza unido a la construcción simultánea de disciplinas nuevas, nuevas "ciencias artificiales" que lo estudian y lo complejizan. Este fue el proceso de construcción del "information processing" en la propia escuela en los años 60s, y de subdisciplinas como el "Speech", "Vision", el "Machine Learning" y ahora Robotics.

El fenómeno citado de "inversión" constituye una innovación en la cultura de una institución universitaria. Su explicación sólo es posible si estudiamos con más detenimiento la cultura de Carnegie Tech como instituto tecnológico, y su

utilización tradicional de la ciencia como parte instrumental en la educación del ingeniero. En Carnegie Mellon, el "problem solving" instrumentaliza a la ciencia, hasta convertir al científico en un "problem solver" (Herb Simon). La citada inversión de conocimientos tiende a priorizar como es propio del diseño la síntesis, sobre el análisis. El "computer scientist" recaba para ello todo tipo de conocimientos y disciplinas, en una dinámica conocida en CMU como "Putting It All Together" (Allen Newell). Busca una "síntesis de conocimientos". Forma una nueva interdisciplinariedad.

El segundo corolario de la hipótesis "conocimiento es diseño" podríamos formularlo así: La inversión de conocimientos antes citada precisa de una nueva interdisciplinariedad que se genera en torno al ordenador en la SCS, y tiende a la formación de síntesis de conocimientos nuevos igualmente basados en el diseño, acordes con los sistemas artificiales en construcción. La investigación en la SCS está abierta a una "collaboration across boundaries" o trabajo interdisciplinar entre distintas disciplinas del campus.

El antiguo inventor pragmático, también era experto en la síntesis de los más variados componentes, a través de una interminable serie de experimentos de laboratorio. Este método de "cut and try" se ejercía sobre los más diversos materiales y procedimientos. Proceso no muy distinto de lo que Levi-Strauss denomina el pensamiento "bricoleur", propio de la magia. Popularmente se conocía a Edison con el apodo del "Wizard of Menlo Park" (El Mago de Menlo Park).

La construcción de la Inteligencia Artificial se ha servido a su vez de distintas disciplinas (sicología cognitiva, lingüística,...), aprovechando de ellas aquellos enfoques y herramientas conceptuales y metodológicas aptas para la construcción del nuevo artefacto.

La SCS no se reduce en su investigación a una sólo escuela sino que tendencialmente atrae a su esfera de influencia progresivamente al conjunto de disciplinas del campus, hasta formar una "computer-intensive university" , proceso iniciado en los años 80s y aún no culminado. El College of Humanities and Social Sciences forma disciplinas, como Computational Linguistics, Information Processing Psychology, o programas de doctorado como el Information and Decision Systems. Igualmente otras escuelas.

Este fenómeno tiene base en la cultura de una institución que desde los años 40s con el Plan Doherty, introdujo el concepto de "liberal-professional education", por el cual la educación del ingeniero debía ser al mismo tiempo profesional y

liberal, ésto es, basada en el conocimiento de humanidades y ciencias. Y a su vez, a los de "liberal arts" se les educaba en la mentalidad de problem solving del ingeniero.

Este fenómeno ya fue detectado por Herb Simon en 1969: "Los que hemos vivido en íntimo contacto con el desarrollo de la computadora moderna, durante su gestación y su infancia, hemos pasado ante una gran variedad de campos profesionales, entre los cuales se encuentra la música. Hemos advertido la creciente comunicación entre disciplinas intelectuales que se efectúa en torno a la computadora. La hemos acogido con agrado porque nos ha puesto en contacto con nuevos mundos de conocimientos, porque nos ha ayudado a combatir nuestro aislamiento multicultural" (1973: 123).

Pero este fenómeno tiende a cambiar el propio patrón cultural de una universidad, en sus fines y en sus jerarquías y formas de organización y hábitos internos.

El sistema cultural tradicional de la Universidad occidental, como institución especializada en la producción de saber, está basado en el supuesto de que el valor del conocimiento aumenta en proporción directa a su grado de formalización lógico-matemática. Se le ha denominado "el ethos de la ciencia pura". (Barnes, Merton, Kuhn, et al. 1980). En palabras de Simon, la cultura general de la universidad implica un conjunto de normas que tienden a priorizar la formalización de las materias: " De acuerdo a las normas vigentes, la respetabilidad académica exige aquellas materias que son, intelectualmente, arduas, analíticas, susceptibles de formalizarse o de enseñarse". (1973: 88-89).

La ciencia (matemáticas, física, química, biología) es el conocimiento más valorado, y se considera a la tecnología (ingenierías) como una aplicación de sus conocimientos. Por ello, para el ingeniero tradicional considerarse "científico" significa aumentar de estatus cultural. Esta dinámica quedaría expresada en la tradicional relación Harvard-M.I.T. en Estados Unidos, en la que, simplificando constatamos que la primera institución es la que aporta los Premios Nobel, los Presidentes del país y los Premios Pulitzer y la segunda, los ingenieros, los jefes de empresa, y también algún Nobel.

La tendencia hacia la unificación de conocimientos y creación de nuevas disciplinas de diseño, como factor cultural propio de la SCS, bajo una idea de "problem solving", constituye una innovación en la cultura de una institución universitaria, incluso de la propia CMU, que originalmente fue un instituto

tecnológico.

Al priorizar el conocimiento y construcción de sistemas posibles, la síntesis, sobre su análisis, el "computer scientist", como el ingeniero, recaba para ello todo tipo de conocimientos y disciplinas: "Putting It All Together" (Allen Newell). Busca "síntesis de conocimientos". El tema de la "synthesis" (Art Westerberg, 1979, 1988) se convierte en clave de la nueva ingeniería, basada en el diseño.

La computer science de CMU avanza sobre la base de síntesis con otros conocimientos y disciplinas formando nuevos conocimientos artificiales, nuevos conocimientos de diseño. En este sentido se le puede considerar una matemática de las tecnoculturas.

La Artificial Intelligence de CMU de Simon-Newell-Shaw fue posible gracias a una síntesis entre matemática aplicada, problem solving, métodos de introspección psicológica y técnicas de programación informática.

El invento de Ed Feigenbaum de los sistemas expertos, fue otro avance al unir el ingeniero de conocimiento con el médico o el experto y trabajar juntos para poner en marcha un producto realizado por los dos.

Este es el fenómeno de intento de síntesis con el "complex information processing" de Simon-Newell, una disciplina o campo de investigación unitario entre psicología y AI.

La línea de avance de la Computer Science es la abierta con la experiencia Andrew: abrirse a la colaboración con otras disciplinas, hacer nuevas síntesis de conocimientos artificiales como la propio Computer Science.

Pero esta síntesis de conocimientos tiene sentido sólo si se pretende construir nuevos sistemas artificiales, más y más complejos. Y aquí llegamos a los límites del "computer" como sistema para conseguir avanzar en dichas síntesis.

4.2. SEGUNDA HIPOTESIS: "KOWLEDGE IS LOGIC" Y "KOWLEDGE IS SEARCH" en la SCS.

Pero que el diseño haya sido hasta ahora la forma de conocimiento que predomina en la escuela no garantiza que lo siga siendo. El conflicto con una concepción de la Computer Science como ciencia empírica o como una rama de la lógica o las matemáticas está presente.

Respecto a concebir la Computer Science como ciencia empírica ya hemos visto los efectos que implica. Analizemos ahora qué implica concebirla como una parte de

la lógica o las matemáticas.

El ordenador nació como una máquina de calcular hecha por matemáticos para realizar con más rapidez y ausencia de errores sus operaciones aritméticas. Dado que la matemática es el lenguaje de la ciencia, el ordenador tiende a convertirse en la máquina por excelencia de la universidad, la institución principal de la ciencia, como sistema de conocimiento cultural.

Ello conduce a que tiendan a ser matemáticos y lógicos los que dirijan la evolución teórica de la máquina más específicamente suya. Y así ha ocurrido en CMU, y en el resto de la comunidad de computer scientists de Estados Unidos. Ello implica considerar subordinada la labor del ingeniero, del constructor de instrumentos, a la del matemático aplicado, que diseña la construcción formal de los ordenadores.

El reciente proyecto dirigido por Peter J. Denning de la Association for Computing Machinery (1989) sobre un "Core of Computer Science" pretende hacer oficial esta orientación académica . Propone un "nuevo" paradigma para la computer science que claramente coloca a "los matemáticos" a la cabeza de la disciplina, dándoles el mando de la "theory". Más abajo coloca a los "computer scientists", considerados como aplicadores de sus teorías en forma de "modelos", según el patrón cultural de los científicos naturales. Y finalmente, se coloca a los "ingenieros" como los encargados del diseño . Por último, se define la "computer science" como "el estudio sistemático del proceso algorítmico, que describe y transforma información" (P. Deening. 1989), siendo los algoritmos una área de trabajo propia de la matemática.

Esta definición, de aceptarse en CMU , puede dejar fuera de la escuela a la mayor parte de la Inteligencia Artificial, y a gran parte de la área del software y de hardware . Paradójicamente dejarían de ser "computer scientists" dos de los fundadores de la disciplina en CMU : Herbert Simon y Allen Newell, que no son partidarios de esa definición de la computer science. Quedaría fuera también parte de la área de Robotics, Takeo Kanade, Red Whittaker,... Se rompería el pacto de 1964.

La computer science así concebida se diferencia radicalmente de la práctica tradicional del ingeniero al creer que construcción formal de un sistema tecnológico es ya el sistema tecnológico. Los ingenieros tradicionales asociaban inseparablemente el proyecto de ingeniería con su realización práctica. Eran considerados "system builders". Trabajaban, normalmente al servicio de empresas

que les planteaban problemas concretos del mundo real a resolver ("real world problems"). Eran profesionales, el centro de su educación era la preparación para la solución de los llamados "real world problems", era una educación práctica, quizá en exceso.

El actual computer scientist, y también el ingeniero cada vez más, trabaja cada vez más con simulaciones. La cultura universitaria tradicional presiona al "computer scientist" en dirección hacia la Theory, y dentro de ella a sus partes más lógico matemáticas, en lugar de las más operativas. La escuela tiende, en tanto que parte de una cultura universitaria, a la propia lógica de su investigación y la docencia, en lugar de a los problemas del llamado "real world".

La construcción de la máquina informática formal por A. Turing ha tenido un doble efecto en este sentido: Por un lado, ha permitido construir modelos formales de computación liberados de las constricciones de la ingeniería tradicional, pero a cambio, las máquinas resultantes son meramente formales, no tecnológicamente relevantes, lo cual las coloca por detrás de las que la ingeniería ha construido tradicionalmente.

Conforme el Computer Science va adquiriendo autoridad universitaria, va adaptándose a la cultura de la misma, presidida por la necesidad de formalización y matematización de los conocimientos. Así, la área de Theory, tardíamente iniciada a principios de los 80s, va cobrando fuerza conforme se afianza la School, desplazando los métodos más pragmáticos del inicio. Dana Scott, lógico, matemático y filósofo, desplaza en la School a Herbert Simon o Allen Newell, como autoridad en Computer Science en la área de Theory. La "racionalidad limitada" no puede competir con la racionalidad que funda la matriz del concepto de "animal racional", la razón lógico-matemática, base a su vez de la filosofía clásica griega.

Pero esta tendencia a cambio tiende a concebir los sistemas informáticos como sistemas formales, y por lo tanto no regidos por el criterio de viabilidad en el "real world". El constructivismo matemático no es equivalente a la "computer science", al diseño de sistemas informáticos. Un sistema lógico puede ser consistente, pero inviable para construir un programa de ordenador que funcione. Esta creencia está favorecida por la separación entre "computer scientists" y "electrical and computer engineers". Escisión iniciada con Von Neumann y su intento de establecer una disciplina universitaria científica, separada y por

encima de la ingeniería de Eckert y Mauchly, los ingenieros constructores de ENIAC, el primer ordenador digital. (ASPRAY, 1980:120).

Ello produce una tendencia a a identificar sistemas informáticos con sistemas formales matemáticos, y Theory con teoría matemática.

Ello tiende a congelar el diseño en Computer Science en un diseño formal, un diseño de sistemas formales, válido en si mismo, sin necesidad de implementación práctica. Este diseño puede consistir también en un DISEÑO SIMULADO por ordenador. Reduce la computer science a matemáticas.

Pero la Computer Science nació de la tendencia contraria de los científicos y matemáticos a hacerse diseñadores ingenieros, a tratar con "real world problems", a hacerse aplicados (Von Neumann).

La Computer Science avanza a través de la ingeniería, y de una teoría y práctica de diseño de sistemas informáticos.

Existen límites a la computer science considerada como disciplina basada en las matemáticas aplicadas.

Esta hipótesis la podríamos formular así: El diseño formal de sistemas informáticos no implica su realización. Si estos sistemas no se realizan existen sólo como sistemas formales o simulados, pero inoperativos. La lógica y la matemática son instrumentos de la Computer Science pero no pueden ser su teoría, pues están en función de la construcción de un sistema informático sin el cual no tienen sentido. "Mathematical Logic is mostly used as a tool, not as a foundation". (Giuseppe Longo. Some Aspects of Impredicativity. 1988. CMU-88-135).

La formalización de una teoría de la Computer Science como teoría de los sistemas formales, computacionales en un sentido formal, transforma a esta disciplina en una rama de la matemática y la escinde en sus dos componentes iniciales: matemáticas e ingeniería. Este fenómeno constituye "una regresión de conocimiento". En este sentido el Computer Scientist en lugar de una nueva figura, suma de científico e ingeniero se queda en un ingeniero formal, o un matemático aplicado.

El no haber podido encontrar un estatus nuevo a la Computer Science como nuevo tipo de conocimiento provoca una tendencia a definirlo simplemente como matemática aplicada en su Theory. Desde los años 80s, han aumentado las presiones académicas, no precisamente de los departamentos principales del país, para que la "Computer Science" se definiera y se organizara siguiendo el modelo

clásico de la Academia presidido por la matemática, "knowledge is logic" o de la ciencia natural, "knowledge is search", acorde con los patrones mayoritarios de la institución universitaria moderna (P. Denning Report. 1989).

Esta presión académica, que comparten los propios departamentos de Computer Science del país, es tanto más fuerte cuanto la Universidad es, en Estados Unidos, la institución cultural de mayor prestigio internacional. Como institución de investigación básica, su sistema de promoción sigue estando regido por el patrón de la ciencia tradicional, cuya base es la producción de descubrimientos científicos significativos.

Esta orientación de la computer science presiona en CMU para dar más fuerza a una "professor mentality", ligada a la consecución de "tenured" (puesto fijo similar al titular universitario europeo), y una equiparación con el sistema de ascenso del mundo universitario clásico, basado en la redacción de trabajos ("papers") individuales, publicados en forma de artículo científico en revistas reconocidas por la comunidad de la especialidad en cuestión.

Todo ello hace que Computer Science puede entenderse como una disciplina transicional, en inestable equilibrio entre una cultura tecnológica basada en el diseño y una cultura científica basada en el descubrimiento ("search") o el razonamiento ("logic"). Ha indicado el camino futuro de la tecnología, el diseño, pero aún es una tecnología subordinada al sistema de conocimiento científico, presidido por la lógica y la matemática. Y ello tiende a normalizar su situación como una ciencia más al lado de las restantes.

Es concebible que la "computer science" en los próximos años se vea superada por un nuevo tipo de disciplina directamente con base en el diseño como nuevo tipo de saber cultural, con base directa en la ingeniería, y un nuevo tipo de máquina, que podríamos llamar, ya no "computer" sino "designer", con una función directamente diseñadora, inventada por ingenieros provenientes de diversos campos, incluidas las ciencias sociales, y por "computer scientists", jugando un papel complementario.

4.3. TERCERA HIPOTESIS: REDES ESTRATEGICAS NACIONALES VS. CULTURA ACADEMICA EN LA SCS, DOS SISTEMAS ORGANIZATIVOS EN CONFLICTO.

Este conflicto entre sistemas de conocimiento distintos en Computer Science está relacionado con sistemas de organización a su vez, diferentes.

En la formación de Computer Science en CMU como sistema cultural basado en el diseño, entra en juego la particular relación establecida desde principios de los 60s entre el Computer Science Department de CMU y el Departamento de Defensa, en concreto su Information Science and Technology Office (ISTO) de la Defense Advance Research Project Agency (DARPA), posiblemente la principal institución de investigación de Estados Unidos, en esta área.

Durante tres décadas, computer scientists destacados de CMU, como Allan Newell, Raj Reddy, Takeo Kanade, y últimamente Angel Jordan han jugado un papel iniciador en la propuesta de programas tecnológicos de carácter estratégico : Inteligencia Artificial, Speech, Vision, Software, Supercomputers , Sistemas Operativos y ahora HDTV (Television de Alta Definicion).

Los propios proyectos analizados (NECTAR, MACH, GANDALF) son, ante todo, proyectos tecnológicos estratégicos para la década de los 90s, encabezados por científicos de la segunda generación (R.Rashid, H.T. KUNG, R. Whittaker).

Y es en torno a este tipo de programas, con un marcado acento tecnológico, como científicos como A. Newell o H. Simon han desarrollado sus teorías. Para H.

Simon, el ordenador ha sido simplemente un medio para avanzar en el descubrimiento de las bases del comportamiento inteligente, ante todo, humano.

Por eso su trabajo se ha desarrollado preferentemente en el Departamento de Psicología, financiado por los Institute of National Health. Pero visto desde dichos programas, y desde Computer Science, H. Simon y sus teorías han sido un medio para avanzar en la construcción de sistemas informáticos inteligentes.

La estructura para llevar adelante el tradicional esquema "descubrimiento -invención" era "Universidad-laboratorio industrial". Al iniciarse en CMU la inversión y unificación de conocimiento, la estructura del laboratorio industrial moderno, basada en el modelo de la "invention factory" desarrollado por Edison, y más tarde el "think tank" como laboratorio de la Defensa, según el modelo de la RAND Co se ha introducido dentro de la propia universidad.

" Se refieren a él a menudo, junto con los laboratorios de Stanford, M.I.T. y SRI, como uno de los cuatro Laboratorios de Inteligencia Artificial de ARPA (Advanced Research Projects Agency). (Proposal for Continuation of Research in...1975:8).

Desde la formación del equipo Newell-Simon-Shaw, en los años 50s, constructores del primer programa de Inteligencia Artificial, la estructura de base de este tipo de trabajo no ha sido el clásico departamento universitario de profesores y

estudiantes sino el equipo interdisciplinar y operativo, llamado también "mission oriented".

Este modelo de organización es similar a un "task force": un equipo teórico-práctico, universitario y extrauniversitario, de profesores, investigadores, estudiantes, programadores, staff administrativo, disponiendo del equipo tecnológico más avanzado y que lleva a cabo proyectos que exceden las fuerzas de un clásico departamento universitario. Cliff Shaw fue el programador de la RAND Corporation sin cuya ayuda Simon y Newell no hubieran podido materializar su programa. Este equipo complejo les dio ventaja respecto a Minsky y MacCarthy, meros teóricos de la Inteligencia Artificial, en la famosa reunión de Darmouth en 1956. Estos últimos presentaron a la reunión los clásicos "papers", Simon y Newell llevaron un programa inteligente que podía funcionar como se demostró semanas más tarde.

Lo mismo se puede decir de los actuales equipos de R. Rashid con Mach o H.T. Kung con Nectar.

Este modelo organizativo de equipos interdisciplinarios altamente cualificados y operativos es la base estructural del Computer Science Department de CMU desde su fundación en 1965, y de sus proyectos como Nectar, Mach o Ambler.

Este modelo se complejizó con una innovación decisiva : la formación a finales de los 60s, de ARPANET, la primera red electrónica nacional, organizada a instancias de Robert E. Kahn, de DARPA. Los principales centros de computer scientists de Estados Unidos, incluido el de CMU, fueron los primeros en experimentar un tipo de tecnología nueva: la network informática, que se ha demostrado útil para su tipo de trabajo.

Ello ha introducido un nuevo modelo de organización: la red de proyecto, académica, empresarial y política, estructura básica de realización de los programas de investigación más importantes de CMU.

Los equipos citados forman redes de investigación estratégica con otros equipos todo Estados Unidos que participan en los programas de DARPA, y que se sirven de las redes electrónicas para sus comunicaciones. Estos equipos son los más avanzados en dicha materia, y trabajan ya en centros universitarios ya en laboratorios industriales, formando una red de programa , que constituye una comunidad de referencia principal de estos equipos científicos. Estas son las redes estratégicas de la "computer science" en Estados Unidos.

La SCS es uno de los centros claves de esa organización de redes estratégicas

que son las encargadas de llevar adelante los programas fundamentales de la computer science en Estados Unidos.

Esta estructura de proyectos de la SCS participa de una nueva estructura informal suprauniversitaria basadas en tres instituciones: los propios equipos de investigación universitarios interdisciplinarios, las agencias de investigación federales, y los laboratorios industriales de grandes empresas o de empresas que comienzan ("start up") con personal cualificado para dicho tipo de investigación.

Estas redes son clave en la investigación estratégica en la área del ordenador.

Se han ido formando en los últimos 30 años, en torno a diversos programas (Artificial Intelligence, Speech, Vision, Planetary Robots, y ahora HDTV...).

Este modelo de redes se ha intentado extender al campus.

Por un lado observamos con experimentos como Andrew una tendencia es convertir todo el campus en una gran red o "network", al modelo de SOAR.

Paradójicamente, la SCS no considera a Andrew como su red de trabajo. Porque sus redes son extraacadémicas y supraacadémicas. Ello sitúa a la School en cierta medida de espaldas al campus. La SCS no considera a Carnegie Mellon como su lugar de trabajo, sino a sí misma y sus redes complejas, como lugar de trabajo principal.

CMU ha alcanzado prestigio nacional como universidad de investigación basándose en la SCS, y ésta, a su vez, se apoya en las redes y proyectos estratégicos que la vertebran. Son estos proyectos los que han hecho al Departamento y a la propia Escuela, que es la estructura que da cabida al conjunto de dichos proyectos. Un ejemplo de red es Soar, una red interdisciplinaria de computer scientists de todo Estados Unidos, y con presencia en Europa, organizada en torno al programa de Inteligencia Artificial del mismo nombre por Allen Newell.

Pero dichas redes estratégicas no se ajustan con la estructura de la comunidad académica normal, ni con sus sistemas de valoración ni con sus sistemas de recompensas y ascensos. Ni con su composición, pues no tienen estudiantes de licenciatura. Son redes de investigación no de docencia. Por otra parte, el funcionamiento en red es distinto al de un departamento universitario tradicional, basado en la especialización y la rígida jerarquía de promociones y cargos.

La función de las redes estratégicas es llevar adelante los programas de investigación básica y aplicada en computer science y tecnología en función de

los intereses nacionales de Estados Unidos.

Pero este tipo de trabajo choca con la estructura clásica de una universidad, base de una cultura que como la científica se considera en teoría "universal", y donde sus profesores colaboran en revistas internacionales, realizando publicaciones que basan su ascenso en la escala académica. Y donde han de atender diariamente a sus estudiantes en clases numerosas y con escaso nivel académico, en su mayor parte.

Ambas culturas, la que se basa en redes estratégicas nacionales de carácter tecnológico, y la que se basa en las estructuras universitarias científicas con vocación universal, están en conflicto, y este no es sino un reflejo de dos culturas en pugna en CMU, la "liberal" y la "professional culture", con dos tipos de conocimientos, el científico y el tecnológico.

PARTE SEGUNDA.

CARNEGIE MELLON : DE INSTITUTO TECNOLÓGICO A "COMPUTER - INTENSIVE UNIVERSITY".

1. "EXCELLENCE AND INNOVATION".

El estudio de la investigación en una "computer-intensive university", a partir de los programas de investigación de la School of Computer Science se explica por el hecho de que estamos hablando de una comunidad con una doble característica: 1. Ser una "research university", esto es, uno de los primeros centros universitarios de investigación en Estados Unidos y 2. El ser el primer "computer-intensive campus" del país, el primer experimento de universidad donde cada estudiante, profesor o staff tiene acceso, a través de la red Andrew, a la comunicación informática con el resto del campus, el resto de país y el resto del mundo.

De hecho, en una encuesta realizada por la Administración de CMU entre los estudiantes para conocer las razones de su elección de CMU, las principales eran ante todo la "research reputation", y las "computer facilities". (D. SCOTT. Search for President 1989:21-22).

1.1. LA PRIMERA "COMPUTER-INTENSIVE UNIVERSITY".

Carnegie Mellon University es una institución universitaria norteamericana situada en la ciudad de Pittsburgh, en el Estado de Pennsylvania, al noreste del país. Esta institución es el resultado de la fusión en 1967 de dos instituciones de enseñanza superior e investigación: el Carnegie Institute of Technology y el Mellon Institute for Industrial Research.

Dentro del rango de universidades nacionales, esta catalogada entre los 25 primeros centros de Estados Unidos, de un conjunto de más de 3000 universidades y "colleges" que imparten grados. Se considera una "research university" o

universidad de investigación.

La ciudad de Pittsburgh, donde se encuentra, fue a principios de siglo la capital del acero de Estados Unidos. A lo largo de su historia ha generado dos movimientos ciudadanos de revitalización, los llamados Renaissance, que han tratado de renovar su base industrial, su medio urbano y ecológico. Hoy día, es una ciudad de algo más de 375.230 habitantes en 1988 (Pittsburgh Facts 1990-91), con un área metropolitana de 2.322.246 habitantes previstos para 1990, reconvertida en un centro de alta tecnología. Cuenta con 170 laboratorios de R&D y más de 25.000 científicos e ingenieros, destacando en especial por su investigación en informática (CMU) y en biotecnología (University of Pittsburgh).

En el documento "The Search for the President 1989", preparado por el Faculty Presidential Search Committee, dirigido por John G. Fetkovich y redactado por Dana S. Scott, podemos encontrar los objetivos generales de esta institución: "Carnegie Mellon University is an academic community whose purpose is the creation and dissemination of knowledge, a purpose which can only be achieved with good will and collegiality among its administration, faculty, staff, and student body. As a major research institution, Carnegie Mellon has many unique qualities which gave it both special strengths and special problems." (1989:1).

Esta universidad está compuesta por siete escuelas o facultades: Carnegie Institute of Technology, College of Fine Arts, Mellon College of Science, Graduate School of Industrial Administration, College of Humanities and Social Sciences, School of Urban and Public Affairs y School of Computer Science. Cuenta a su vez con más de 60 centros de investigación.

Según Herbert A. Simon, Richard K. Mellon University Professor of Computer Science and Psychology en esta institución y miembro del Board of Trustees de la misma, los dos objetivos fundamentales de la política de Carnegie Mellon han de ser: "excellence and innovation". (Models of My Life.1991:251).

Según el documento the Search for the President una de las "unique qualities" de esta institución es la posibilidad de colaboración entre distintas disciplinas y colleges lo que resulta más difícil, sino imposible, en otras instituciones: "With its long-standing emphasis on the fine arts, on management professions, on the sciences and engineering, and its newer ventures in interdisciplinary research in areas of the humanities and social sciences, Carnegie Mellon provides both a wide spectrum of opportunities for a liberal-professional

education and a site for exciting graduate and post-doctoral study" (1989:1).

El cuerpo universitario está formado por cinco grupos de personas: el "Board of Trustees", la "University Administration", el "Faculty". los "Students" y el "Staff".

El Board of Trustees, órgano que tiene la decisión última de la universidad en materia de políticas, organización, finanzas y gobierno, si bien no está envuelto en la administración cotidiana de la Universidad que corresponde a la University Administration, al President y al Provost de la misma. No obstante es significativa su figura dado que es propia de una universidad como ésta, (y de una gran parte de las norteamericanas), que son entidades privadas. La mayoría del Board de Trustees son presidentes o directores de empresas, principalmente de Pittsburgh (Westinghouse, USX Co, PPG, Mellon Bank...) aunque también del resto del país.

El Presidente de CMU es el "chief executive officer" (C.E.O) de la Universidad. Esta denominación CEO, es la que se utiliza en la empresa para denominar al director ejecutivo de la misma. Es el encargado principal de diseñar la estrategia de la institución.

Para la dirección operativa de la Universidad el Presidente delega su responsabilidad en el Provost, la figura encargada de la gestión de la investigación y los programas de estudio, varios Vice-Presidents, los Deans o decanos de facultad, Directors y Head of Departments o responsable de los departamentos.

En 1989, el total de profesores o "faculty" era alrededor de 500, divididos en cuatro categorías: "university professors", "professors", "associate professors", y "assistant professors". Los estudiantes de grado o "undergraduate" eran 4200. Había 2000 estudiantes graduados realizando el doctorado, y el staff estaba compuesto por unas 2800 personas: 365 administrativos, 85 instructores, 500 personal de investigación (incluyendo "special faculty", y "research scientists"), 185 administrativos profesionales y 860 personal de apoyo. En resumen, un total de cerca de 10.000 personas formaban esta comunidad.

El estudiantado se distribuía equilibradamente entre un 30% en ingeniería, un 20% en ciencias, un 30% en humanidades y un 20% en "fine arts".(Goals &Strategic Plans: 1988).

Vista a partir de estos datos, ésta universidad parece no tener una especial

diferencia con otras universidades de tamaño y calidad de investigación parecidos en Estados Unidos. Pero en los años 80s, Carnegie Mellon cobró relevancia en todo el país, y a nivel internacional, por ser la primera experiencia mundial en construir una red informática, Andrew. Esta red constituía el primer experimento piloto en la informatización de una universidad, iniciando lo que se denominó un "computer intensive- campus". Este experimento levantó una serie de críticas a cerca de lo que consideraban un paso perverso en la transformación de una universidad en una "Computer U." Ocho años después de iniciado el proyecto, Carnegie Mellon tenía instalados en 1990, aproximadamente 9000 ordenadores personales ("Carnegie Mellon Facts 1990", editado por la oficina de University Planning). De estos 342 estaban a libre disposición de cualquier miembro de CMU, instalados en claustros públicos en las diferentes escuelas. Los principales tipos de ordenadores eran Unix workstations, Apple Macintoshes, IBM PCs, y VMS workstations. Estos están conectados en una red denominada Andrew, instalada en la década de los 80s, que cubre todo el campus, cada edificio y cada despacho. Hay 12.500 conexiones a la red. Esta universidad es un centro ("hub") de la National Science Foundation Net.

Esta red proporciona los siguientes servicios: archivos de 100 gigabytes de memoria a través del "Andrew file system", correo electrónico, más de 3000 boletines electrónicos tanto del campus como de todo el mundo, un servicio de información de las bibliotecas ("Library Information Service"), y conexiones con el Pittsburgh Supercomputer Center, uno de los cinco centros nacionales de la NSF para la investigación científica, dotado de los ordenadores con mayor potencia de cálculo actualmente existentes.

En resumen, según el citado informe: "Carnegie Mellon has probably the finest computing environment of any university in the United States, built around the general theme of personal computers with a network which provides access to shared information." (ibid.: 39).

Su entonces Presidente Richard Cyert afirmaba: "I expect that this will be looked back upon in 25 o 30 years as the most significant move in the 20th century as far as higher education goes". ("Personal Computing" 1983:37). " "It is going to be the prototype for every major university in the country" (Cyert 1982:6).

Por todo ello, esta comunidad universitaria constituye un modelo de lo que puede

considerarse la cultura de una "computer-intensive university", o una universidad informatizada.

Este medio informático es el resultado de toda una cultura tecnológica que está en el origen de esta institución formada por el Carnegie Institute of Technology, y por el Mellon Institute of Industrial Research, instituciones dedicadas la primera a la educación técnica y la segunda a la investigación aplicada a la industria.

Esta "computer university" es el resultado de una "new idea of education" que está en el origen de una institución fundada en 1900 por el industrial Andrew Carnegie.

1.2."INSTITUTES OF TECHNOLOGY" y "LIBERAL-ARTS UNIVERSITIES" : DOS CULTURAS ACADEMICAS DISTINTAS

Para entender el tipo de investigación que se hace en CMU, y en particular en la School of Computer Science, es preciso primero analizar la comunidad en la que dicha investigación tiene lugar. Existe el peligro para un antropólogo europeo, que analiza una comunidad de investigación norteamericana, de creer, que por el hecho de utilizar palabras similares en uno y otro continente, como "universidad", "ciencia" o "investigación", éstas tienen los mismos significados. Existe una importante diferencia de culturas académicas que un análisis más en profundidad puede desvelar.

El nombre de universidad asociado a CMU tiende a confundir si pretendemos identificar esta institución con lo que se entiende en Europa por Universidad. Mientras que las tradicionales Universidades europeas (Oxford, Paris, Lovaina) nacen en la Alta Edad Media, en un medio urbano con una creciente actividad comercial, como creaciones de la Iglesia destinadas al estudio de la teología y la filosofía natural, basada en los clásicos griegos, Carnegie Mellon, a diferencia de éstas, fue inicialmente una institución tecnológica con fines educativos fundada por un industrial en 1900, en el contexto histórico de la gran expansión industrial de Estados Unidos.

Actualmente asistimos en Europa a distintos esfuerzos por adecuar la Universidad a las demandas de una sociedad cada vez más tecnológica. Este proceso de modernización en marcha, que incluye como uno de sus aspectos claves el aumento de la I+D, se realiza, sin embargo, conservando en lo fundamental una triple característica de la universidad tradicional europea: 1. La división de dos

tipos de instituciones universitarias, las Científico-Humanísticas y las Politécnicas. 2. La consideración en ambas de la Ciencia como el conocimiento por excelencia dentro del mundo académico, lo que conlleva el predominio intelectual del primer tipo de universidad sobre la segunda, considerada ésta como la aplicación de los conocimientos producidos por la primera.

Carnegie Mellon University tiene unas características distintas a ese modelo de universidad: 1. Como la mayoría de las universidades norteamericanas, es una institución académica que ha admitido la ingeniería en su seno, así como las disciplinas derivadas de ésta como el management y la informática. 2. A diferencia de éstas, en CMU la cultura tecnológica ha sido predominante sobre la científico-humanística, dado que su propósito inicial fue la educación profesional y no la científico-humanística. 3. Esta relación, expresada en el término "liberal-professional education", es no obstante conflictiva.

Este conflicto se inició a principios de la expansión industrial de Estados Unidos después de la Guerra Civil, con el nacimiento del Massachusetts Institute of Technology en 1861, ante la resistencia de Harvard a los planes de industrial Abbott Lawrence, a introducir una educación científica para "practical purposes". (Noble 1972:22).

Tanto el MIT como Carnegie Tech han evolucionado en dirección hacia la cultura académica tradicional, introduciendo mayores dosis de educación científica y liberal, pero intentando conservar su matriz original como escuelas tecnológicas.

En los años 30s el denominado Carnegie Plan, introdujo en la educación ingenieria patrones académicos de la cultura científico-humanística al modo europeo, la llamada "liberal-arts". Estos, no obstante, fueron concebidos como complementos de su modelo educativo tecnológico predominante, cuyo colegio principal se denominaba College of Engineering and Science, situación inversa al modelo Science and Technology que se generalizará tras la II Guerra Mundial en los campus norteamericanos.

En diferentes momentos de su historia esta institución ha tratado de elaborar ese estilo de pensar y de educar propio que pretende encarnar. Este modelo que se ha denominado a lo largo de su historia de forma distinta: "problem solving", "engineering method", o en la era del computer, "programming" o "science of design". En cualquier caso se trata de un tipo de mentalidad distinta a la clásica de una universidad tradicional. Su objetivo es tratar de resolver "real

world problems", no tanto dedicarse a la contemplación de la verdad al modo académico clásico.

En los años 60s, esta institución paso de ser fundamentalmente educativa a ser investigadora. Se transformó en una "research university". La calidad de su investigación le ha permitido ascender al rango de las 25 primeras instituciones universitarias del país, grupo de élite que representa lo fundamental de la investigación que se realiza en Estados Unidos en cualquier campo de la Ciencia, la Tecnología o las Humanidades.

Esta investigación, basada principalmente en la informática, no obstante está cortada en CMU según un patrón más propio de un laboratorio industrial que el de una universidad de tipo europeo tradicional. En la School of Computer Science, su centro neurálgico, sus mayores logros consisten en programas y sistemas informáticos, como MACH O NECTAR, que implican investigación y desarrollo, R&D. La teoría y la práctica forman un conjunto inseparable, dentro de una mentalidad orientada a conseguir un sistema que funcione, "it-works- mentality".

La "computer culture" en Estados Unidos bajo su forma de departamentos de computer science nació en universidades marcadas por un patrón académico tecnológico. Los tres centros que hoy protagonizan los programas mas avanzados en Inteligencia Artificial, el Massachusetts Institute of Technology, la Stanford University y la Carnegie Mellon University, forman parte, de forma desigual, de esta cultura academica tecnologica norteamericana.

Pero de estas tres, la primera institución que se ha diseñado a sí misma como "computer-intensive university", unificando en una misma red informática tanto a las escuelas de ingeniería como a las de "liberal-arts" ha sido Carnegie Mellon. Fue también ésta la primera que produjo un programa de Inteligencia Artificial, un Instituto de Robótica, un sistema operativo, MACH, que trabaja en cualquier soporte informático, y está trabajando en la primera red informatica de alta velocidad (NECTAR).

Como veremos a continuación, un rasgo que distingue este patrón universitario es la unión de la ingeniería, la ciencia, y las artes en un mismo tipo de institución, en el que la primera disciplina, la ingeniería como profesión tiene un rango igual o incluso superior al de la ciencia al haber adoptado crecientemente sus métodos y conocimientos y en el que la ciencia adopta actitudes más pragmáticas y orientadas a la aplicación propias de la ingeniería. El resultado de ésta tendencia a la unión de la ciencia y la ingeniería lo

podemos ver en el Computer Science, disciplina que aún se discute si su carácter es científico, tecnológico, una nueva disciplina o una síntesis de las anteriores. Por su parte, las "Fines Arts" han influido notablemente en la formación de la cultura de éste campus, en el que en un principio la "technical education" y la educación artística eran complementarias.

Mientras que en Europa, la cultura científico-humanística y la cultura tecnológica forman dos mundos separados, presididos por la primera, en Estados Unidos, la universidad clásica ha admitido en su seno a la ingeniería, al estar aquella más orientada hacia la "purely practical part of science" como ya detectó Tocqueville. Ello ha producido una cultura particular centrada en la idea de "problem solving", un tipo de problemas propios de la tecnología, aquellos a resolver.

Margared Mead en su estudio sobre el carácter cultural norteamericano, "And keep your powder dry" (1942) también detectó la relación entre la mentalidad del inventor y el "problem solving": "For our willingness to work on inventions, our belief that problems are to be solved by purposeful thought and experimentation, is another aspect of this type of character structure... Only in those societies which shifted success from heaven to earth, and so put the whole impact of religion back of efficiency, could we have a type of character in which it became a virtue to do the kind of thinking which lies back of invention, a virtue to set problems and solve them". (1942:207).

Carnegie Mellon es un caso extremo en este modelo, con un peso aún mayor de la mentalidad profesional y orientada al "problem solving", al haber nacido no ya como una universidad, como Stanford, ni siquiera como un Instituto de Tecnología con rango universitario como el MIT, sino como una escuela técnica de grado medio con un particular interés en la educación técnica o "technical education".

Fue la creación de Andrew Carnegie, un industrial que pretendía fundar una institución distinta a las universidades americanas de la época (Harvard, Princeton, Yale...). Este empresario fue una de las principales voces que en la década de 1890 se levantaron contra la inutilidad ("uselessness") de la educación universitaria en Estados Unidos (L. Veysey, 1965:13), y construyó un tipo de institución no sólo distinta sino en abierto rechazo de la tradicional "college education" norteamericana, basada en la cultura académica de la "liberal-arts". Con todo, también diferenció ésta institución de las hasta

entonces escuelas de aprendices de las empresas, al servicio de sus inmediatas necesidades.

La comparación de Carnegie Mellon con universidades clásicas europeas tipo Cambridge, Oxford o París está, por tanto, fuera de contexto histórico. La llamada educación liberal propia de estas universidades y cimentada en los modelos educativos nacidos en la cultura clásica griega, ha sido, en la cultura académica tecnológica de CMU un elemento instrumental al servicio de un patrón cultural diferente, de un tipo humano distinto al de dichas culturas académicas: el "professional".

La idea de la educación en Carnegie Mellon es formar, ante todo, profesionales, esto es, en terminología de esta institución: "problem-solvers", personas capaces de aplicar sus conocimientos a la resolución de los problemas que se plantean en la vida económica, social o cultural. Es en función de esta tarea que se le proporcionan conocimiento sobre la historia, el lenguaje, o las matemáticas, a modo de complemento.

Este modelo, a los ojos de un europeo, conlleva un paradójico planteamiento de la educación y de la investigación, que podríamos comparar con el de un mundo al revés.

Lo que en Europa, y en particular en países como España, es considerado un tipo de educación de carácter aplicado, y por tanto con menor contenido intelectual, esto es, la llamada "formación profesional", en esta cultura académica de CMU es lo contrario: es la educación básica. Y viceversa, lo que para la universidad clásica europea es el modelo ideal, la educación científico y humanística, la impartida en las Universidades tradicionales, en esta cultura tecnológica inicialmente nació como un complemento subordinado a la formación profesional. Este cambio de perspectiva tiene repercusiones a todos los niveles.

Respecto a la investigación en CMU ello influye en una consideración de la investigación aplicada, o "mission oriented" como la principal en CMU. Como veremos al analizar la School of Computer Science, por investigación se entiende, ante todo, el conocimiento para construir sistemas, o "building systems". Y estos son considerados a su vez medios para producir un saber a cerca de la manera de producir más sistemas, el "know-how".

Sir Charles Percy Snow, profesor universitario inglés, afirmaba en los años 60s que existía un conflicto entre "dos culturas" universitarias, por un lado la científica y por otro, la humanística (1978). Pero visto el mundo académico

desde América, y en particular desde CMU, el conflicto tiene tres contendientes pues existe una tercera cultura académica distinta a las dos anteriores : la cultura académica tecnológica. Esta cultura, como veremos, se alimenta tanto de la ciencia como de las humanidades pero sin confundirse con ellas. Es distinta a ellas, y como vemos en CMU, en su actual grado de desarrollo, aún incipiente, puede mediante un producto suyo, el ordenador, unificar a las restantes en un mismo campus universitario.

En esta cultura tecnológica, el "professional" tuvo como primera versión el "engineer". Su cenit en Estados Unidos se produjo entre el último tercio de siglo XIX y las décadas anteriores a la Gran Depresión de 1929. Para afrontar una creciente problemática situación en la empresa, el professional se transformó en "manager", una creación cultural típica norteamericana, basada originalmente en el ingeniero convertido en jefe de empresa y conocedor de "social relations", figura que dominó la escena de los años 30s y 40s. Tras la segunda guerra mundial, y el auge de la ciencia en Norteamérica, y la invención de la computadora, surgió el computer scientist, que domina el panorama tecnológico desde entonces. Y todos ellos tienen con la ciencia y las humanidades una relación instrumental.

Así, mientras que la ingeniería, por ejemplo, precisa conocer principios de la ciencia (física, química o biología) sin confundirse con ésta, el management, como vemos en esta universidad, se alimenta de las ciencias sociales (economía, ciencia política,...) siendo una disciplina profesional distinta a aquellas.

Igualmente, la Computer Science utiliza las matemáticas aplicadas siendo distinta a éstas, y la Inteligencia Artificial hace lo mismo con la Psicología Cognitiva o la Lingüística sin confundirse con éstas. Ello no significa que la relación sea estable y sin conflictos.

Los elementos de liberal-arts tienden a convertir a los profesionales en su instrumento, el ingeniero tiende a convertirse en científico, el manager en profesor de economía y el computer scientist en matemático. Pero esa propia tendencia indica que la relación es distinta a la existente en una cultura universitaria normal, como la europea. Esta tendencia a revertir el modelo se apreció en los años 80s con la denominada batalla del "core curriculum" en CMU, por la cual se pretendió sentar una base de cursos de ciencias y humanidades principalmente como modelo unitario del campus, complementado con el uso de ordenadores a través de la red ANDREW. Pero este modelo no ha prosperado.

Ante la crisis ahora de este modelo, y si la historia sirve de lección, la tendencia lógica sería a utilizar una nueva versión ingenieril de las ciencias sociales para dar un carácter más complejo a la solución de la misma.

Complementar la "high tech" con una "high policy" que considere los aspectos sociales y culturales de la misma.

El carácter de esta institución, Carnegie Mellon, como una cultura académica inicialmente con un origen distinto a las de las liberal-arts universities" la explica acertadamente Herb Simon, uno de los miembros veteranos de esta institución: " In the United States, institutes of technology...have had rather different traditions from the liberal arts colleges, and even from the nonprofessional divisions of research universities.

The liberal arts tradition...is one of extreme decentralization. In its limiting form, a faculty is a collection of individuals...organized in departments along disciplinary lines,...and the chairman of the department is primus inter pares...

At the time I came to Carnegie (1949), institutes of technology lived much more closer to traditions that came out of industrial practice...The administrative hierarchy was clearly defined...It was not unthinkable for research areas to be defined by a department for its members, or for faculty members to be urged or persuaded to aim their efforts in particular directions.

Carnegie Tech, and GSIA within it, clearly belonged to the tradition of institutes of technology, not the liberal arts tradition." (Models of My Life. 1991:157-158).

Esta cultura tecnológica en Estados Unidos se empezó a desarrollar en instituciones como el MIT (1865) y como Carnegie Tech. Los primeros institutos tecnológicos norteamericanos fueron el Rensselaer Polytechnic Institute (1824) y el Polytechnic Institute of Brooklyn, ambos en el Estado de New York. George Bugliarello, presidente de la Polytechnic University de New York, y antiguo profesor de CMU, ha señalado que el modelo de estas primeras instituciones es l'Ecole Polytechnique, fundada por la Convención francesa, en 1794, en el cénit del periodo revolucionario. (1991:34).

A lo largo de su historia, este modelo académico ha ido produciendo un estilo de pensamiento que se ha pretendido codificar como un modo de pensamiento propio y para algunos profesores un modelo claramente distinto al de la ciencia , al menos en esta universidad de CMU. Así Richard Teare, decano del College of

Ingeniería, afirmaba en 1963: " It is clear, I think, that what distinguishes the engineer from scientist, apart from education, is that while the scientist's goal is to add knowledge, discover new phenomena, establish new principles, the engineer is concerned with devising and designing systems which not only work, but which meet often conflicting criteria of cost, efficiency, life, manufacturability, reliability, safety and so forth...Courses will be needed that are different in content and point of view than those we have now. And just as Carnegie Tech has been a leader in the developments in engineering education that took place in the last two decades, we want to continue to lead in the future". (citado en P. Covey Ed. 1988:13).

El llamado Carnegie Plan introdujo en los años 30s un mayor peso de la liberal-arts culture en el curriculum ingeniero, pasando a denominar su modelo educativo la "liberal-professional education". Este ideal educativo que aún se mantiene en CMU conserva la manera de pensar del ingeniero o profesional como "core knowledge" siendo la educación liberal su complemento. La relación por tanto entre la cultura científico-humanística y la tecnológica en CMU se halla invertida.

A este modo cultural de pensar y de actuar, se le denomina en esta universidad: "problem solving", y más recientemente "design", y constituye el sistema de conocimiento clave de esta cultura tecnológica, de la misma forma que la lógica o la ciencia empírica y experimental constituye el "método" de la cultura científica europea, nacida de la filosofía griega. Ambos sistemas de conocimiento están unidos conflictivamente en esta institución y su tensión no está resuelta.

En los años 80s, proyectos como Andrew, y la batalla del "core curriculum" intentaron cambiar esta situación. Se pretendía poner en el centro un modelo de "liberal education", complementado con medios técnicos avanzados como una red informática. Se intentó unificar la antigua escisión entre Harvard y el MIT, pero bajo la dirección de Harvard. Era la llamada "revolution in education", que venía a culminar la "cognitive revolution" iniciada en los 60s.

La idea era una "síntesis de conocimientos", liderada por la ciencia cognitiva, y complementada con la tecnología informática.

Pero este modelo a finales de los 80s dejaba un problema nacional sin resolver.

El informe del MIT, Made in America, de 1989 alertaba que Estados Unidos estaba perdiendo su liderazgo industrial respecto a Japón, país con un nivel científico

inferior al norteamericano, pero con una producción de 73.000 ingenieros al año, casi la misma cifra que Estados Unidos con la mitad de la población.

Por otro lado, el fin de la Guerra Fría basado en el poderío nuclear, ponía en crisis el modelo de investigación basado en el predominio de la ciencia sobre la tecnología, y dentro de la ciencia de la física sobre las restantes disciplinas.

La "sociedad del conocimiento", liderada por la ciencia, no se muestra capaz de mantener el papel líder de Estados Unidos en el mundo.

Esta crisis del modelo de "research university" replantea el papel de las grandes universidades del país, y en particular a los grandes laboratorios nacionales, que en gran medida viven de la investigación financiada con dinero público.

Las universidades de la Ivy League, con Harvard, Yale o Princeton a la cabeza, marcan el patrón académico mayoritario, según el cual, la educación científico-humanística, propia de la cultura occidental, es la que marca el modelo dominante y la educación de la clase dirigente. De ellas salen los Presidentes de la nación, la mayoría de Premios Nobels en el campo de la ciencia, y los Premios Pulitzer, en el campo de la literatura y las artes.

Pero este modelo, como ahora la competencia con Japón demuestra, ya no garantiza, que los Estados Unidos continúen en su papel de líder internacional.

Está planteado en toda la "higher education" en América la necesidad de una reforma profunda.

CMU es una institución joven que intenta buscar una ventaja competitiva con otras universidades mediante la adopción de un valor, que es propio de la tecnología: la "innovation".

Carnegie Mellon atraviesa tras la experiencia Andrew, una etapa de transición, como le sucede al conjunto de la universidad en Estados Unidos. En 1990 eligió a un nuevo Presidente, el Dr. Robert Mehrabian, anteriormente decano de la School of Engineering en la University of California at Santa Barbara. En un momento de crisis e incertidumbre en el conjunto del país, los ingenieros vuelven a la presidencia de CMU.

El modelo dominante de la anterior administración intentó hacer del diseño una ciencia, del computer una ciencia, del "problem solving" una ciencia. En suma, trató de convertir a la ingeniería en una ciencia más. De ahí que estuviera presidido por la idea de "The Sciences of the Artificial".

El problema en el próximo periodo parece ser otro: como formar una nueva

ingeniería ante todo una ingeniería cultural propia de las "computer cultures", una ingeniería de diseño de computer cultures, ante todo de sus aspectos de organización.

Una necesidad similar de impulsar la cultura tecnológica se planteó a principios de siglo, en el momento de la fundación de esta institución.

2. "A NEW IDEA OF EDUCATION" (ANDREW CARNEGIE).

2.1. LAS CARNEGIE TECHNICAL SCHOOLS.

Existe una historia particular que explica como ha llegado a conformarse esta nueva "computer-intensive university", como se ha formado su actual patrón cultural como "research university", que puede dar luz a cerca de cómo resolver sus actuales problemas.

El origen de esta "problem solving culture" en Carnegie Mellon hay que buscarlo en la fundación en 1900 de las Carnegie Technical Schools por Andrew Carnegie (1835-1919).

En la década de 1890, Carnegie Steel constituía la más concentrada y moderna industria del mundo, gracias a los métodos de Andrew Carnegie, basados en la combinación de la más avanzada tecnología Bessemer de fundición, las economías de escala y una política de reducción sistemática de costos. Este tipo de industria revolucionó la estructura del resto de sectores productivos y estableció el modelo que a continuación imitaron el petróleo, el caucho, los automóviles, el equipamiento eléctrico, y más recientemente la propia industria del "computer". Pittsburgh era conocida entonces como "the Workshop of the World".

Andrew Carnegie, un emigrante escocés, fue el artífice de esta industria.

Empezó trabajando a los 13 años de aprendiz en una fábrica de algodón, continuó como superintendente de los ferrocarriles de Pensilvania a los 24 años y en su madurez se convertiría en el industrial más rico del globo, con unos beneficios anuales en 1900 de 40 millones de dólares.

En la cumbre de su poderío industrial, este empresario aportó un millón de dólares para crear ese mismo año las Carnegie Technical Schools, origen de la actual Carnegie Mellon University. Esta institución pretendía ser la cristalización de lo que Andrew Carnegie denominó: "A new idea of education". ("The Idea of Education at Carnegie-Mellon". Focus. 1979)

Este industrial, cuya educación formal no pasó de la escuela primaria, donó el 80% de su fortuna personal, considerada en su época una de las más grandes del mundo, a fines educativos (Swetnam:149). En 1889, en un discurso de fundación de una biblioteca para obreros del metal en el barrio de Braddock, Andrew Carnegie resumía así su filosofía educativa:

" We occasionally find traces even at this day of the old prejudice which existed against educating the masses of the people...Men have wasted their precious years trying to extract education from an ignorant past whose chief province is to teach us, not what to adopt, but what to avoid...

They have in no sense received instruction. On the contrary, what they have obtained has served to imbue them with false ideas and to give them a distaste for practical life. I do not wonder that a prejudice has arisen and still exists against such education. In my own experience, I can say that I have known few young men intended for business who were not injured by a collegiate education. Had they gone into active work during the years spent at college they would have been better educated men in every true sense of that term. The fire and energy have been stamped out of them, and how to manage a life of idleness and not a life of usefulness, has become the chief question with them. But a new idea of education is now upon us". (Focus 79:)

Esta filosofía pragmática, democrática y basada en la idea de utilidad, es la que condujo a la construcción de la Carnegie Technical School en 1900, bajo el lema de Andrew Carnegie: "My heart is in the work". Esta idea no es un mero deseo sentimental o formal, resume un tipo de cultura que cuando inventa o produce algo, sus autores lo primero que afirman es "it works", "funciona", "trabaja".

Este tipo de pensamiento no fue patrimonio exclusivo de Andrew Carnegie, sino un patrimonio ideológico común de la denominada Progressive Era en la Norteamérica de principios de siglo, la era dorada del "professional engineer", de las grandes corporaciones como General Electric o General Motors, del Taylorism o "scientific management" y del Massachusetts Institute of Technology. La América de los Edison, los Ford, los Taylor.

David F. Noble, en "America by Design"(1977) explica el surgimiento de la "technical education" en Estados Unidos en el periodo que siguió a la Guerra Civil: "Technical education in the United States, developed in struggle with the classical colleges, both inside and outside of them. One form of this

development was the gradual growth of technological studies within the classical colleges, resulting from the reorientation of natural philosophy toward the empirical, experimental, scientific research for truth and from the pressures of some scientists and powerful industrialists for practical instruction; the other was the rise of technical colleges and institutes outside of the traditional colleges in response to the demands of internal improvement projects like canal-building, railroads, manufactures, and, eventually, science-based industry". (1977:20).

La Stanford University, fundada en 1891 gracias a la donación de la familia Stanford, podría considerarse como expresión del primer modelo, iniciado por Ezra Cornell al fundar la Cornell University, en 1868. Este constituye un modelo de reforma de la clásica universidad europea desde dentro, experimento que más adelante analizaremos.

Pero el modelo central de este periodo lo constituyó el Massachusetts Institute of Technology (MIT), construido en Boston en 1865 independientemente de Harvard University ante las resistencias de esta Universidad a adoptar la "technical education". El MIT tomó por lema: "Mens et Manus", un patrón cultural diferente al de Harvard: "Veritas". El MIT fue la cuna del concepto "technology" propuesto por primera vez por el médico Jacob Bigelow, indicando con ello que el estudio de la ciencia iría dirigido a conseguir fines prácticos.

El MIT es la institución que instituyó el patrón del ingeniero profesional en Estados Unidos, formando a la élite dirigente de las corporaciones industriales de Estados Unidos durante la época dorada de las General Electric, General Motors, y la Ford. (Noble:43).

Las Carnegie Technical School, fundadas más tarde, constituyeron una versión aún más radical,(y también más local) de este modelo. Esta versión constituye también una versión más típicamente americana y menos cosmopolita,esto es, europea, de este segundo modelo de "technical education". Fueron concebidas en su inicio como una institución más democrática que los institutos de ingeniería de rango universitario al modo MIT. Su finalidad fue formar masas de especialistas y profesionales para la industria y los negocios locales de Pittsburgh, en un momento en que esta ciudad estaba considerada "the workshop of the world" por su industria siderúrgica floreciente, que era la primera industria moderna de Estados Unidos, y la primera con una facturación de miles de millones de dólares.

"This is the age of the specialist: therefore you who have to make your living in this world should resolve to know the art which gives you support: to know that thoroughly and well, to be an expert in your speciality". (Andrew Carnegie . Focus. 1979).

Inicialmente la institución se diseñó para atender las necesidades locales, industriales, económicas y educativas de la ciudad: "The type of Industrial Education ...adapted for the Carnegie Technical Schools of Pittsburgh has been specially designed to meet the local, industrial, economic, and educational needs which are found to exist..." (Preliminary Report, 1902:15).

"Education is a direct preparation for life, and modern civilization demands distinct and defined vocations. And that education is most lasting and effective, which is directly related to the lifework of the individual...Therefore:-You should recognize that SPECIALIZATION in any one particular line of work, whether in the arts, sciences, or in any field of human endeavor, is the path which leads most directly to success. Specialization means thoroughness in a particular calling". (Preliminary Report, 1902: 16).

2.2."TECHNICAL EDUCATION": JAMES WATT COMO MODELO.

Los documentos fundacionales de esta institución fueron la carta escrita por Andrew Carnegie el 15 de Noviembre de 1900 al alcalde de Pittsburgh, en donde le comunica su intención de fundar estas escuelas, y la "proposal" formulada en 1903 para desarrollar las finalidades, estructura y tareas de las mismas, basada en un Preliminary Report realizado el año anterior.

En la primera de ellas podemos apreciar que su intención era proporcionar "useful education for the sons and daughters of working men". Estas escuelas se basarían en el modelo de institutos técnicos como los de Boston, Worcester en Massachusetts y Drexdel Institute en Philadelphia. En particular, tras una visita a Gran Bretaña al Keithley Institute y al Halifax Institute, lo que más impresionó a Andrew Carnegie fue la combinación que realizaban estos jóvenes entre trabajo y estudio: "workers during the day, improving themselves in various studies pertaining to their crafts in the evening classes of these institutions. I told these students that this impresses me more than anything else".(. Plan and Scope,1903:4).

Esta orientación hacia la "vocational training" para las jóvenes generaciones hijos de los trabajadores es un rasgo específico de Carnegie Tech. Ni Stanford

University ni MIT tenían en torno suyo una población de trabajadores como Pittsburgh necesitada de formación profesional de grado medio. La gran expansión de actividades comerciales e industriales se había adelantado en esta ciudad a su oferta educativa, produciendo anualmente 120.000 jóvenes buscando un sistema de aprendizaje que les prepara directamente para el empleo, "to their life of work".

El lema que Andrew Carnegie eligió para explicar el sentido de estas escuelas lo tomó de su abuelo, zapatero de profesión: "Handication versus Headication", esto es, dedicación manual versus dedicación intelectual. Este lema resume su modelo educativo, basado más en la figura del artesano que la del moderno obrero industrial. Su lugar de trabajo era el taller, "the workshop".

El modelo educativo elegido para las Carnegie Technical Schools estaba inspirado en la figura de Andrew Watt, compatriota de Andrew Carnegie, y de quien escribiría su autobiografía, al que llamaba "the creator of the most potent instrument of mechanical force known to man", "the inventor of the greatest of all inventions" (Swetman, G. & Smith, H., 1989 :129)

En esa biografía relata la triple formación de Watt: "While Watt was still in boyhood, his wise father not only taught him writing and arithmetic, but also provided a set of small tools for him in the shop among the workmen- a wise and epoch-making gift, for young Watt soon revealed such wonderful manual dexterity, and could do such astonishing things, that the verdict of one of the workmen, "Jamie has a fortune at his finger-ends" became a common saying among them." (1905:11).

Esta visión de Watt es con todo unilateral. La máquina de vapor no la inventó Watt en un taller artesano sino en un taller de la Universidad de Glasgow, en Escocia. Esta era una universidad con características particulares como ser la primera en establecer una escuela de ingeniería y un profesorado en esta materia, y según Andrew Carnegie "American universities have, fortunately, followed the Glasgow model". (1905:35). Un grupo de científicos y filósofos entre los que se encontraban Adam Smith, Lord Kelvin, y otros habían acogido en ella a Watt como mecánico, proporcionándole una pequeña habitación de 6 metros cuadrados.

Instituciones como el MIT tenían ya el lema "mens et manus". Pero el lema de Carnegie Tech : "My heart is in the work" es aún más radical, implica una conversión de la institución educativa en un taller, o "workshop". No quiso

organizar en Pittsburgh una universidad al estilo Glasgow sino una escuela de formación profesional de grado medio. Sólo más tarde se convertiría en un instituto tecnológico al estilo MIT.

El 4 de abril de 1910, en su segunda visita al campus, Andrew Carnegie explicaba así a los estudiantes su modelo educativo: " In the struggle for life some of you may place upon yourself the ambition to be a multi-millionaire. That is a low ambition. The engineering professions you have chosen, for instance, are much higher and more worthy. Your first duty is to be self-maintaining, because if you are not that you will not possess those qualities of independence, aggressiveness, and content that mark a true man.

Surplus wealth is a great trust and should be administered for the good of the community. It does not add to human happiness, and if you have sons it may lead them astray....The ignorant man has no ambition but millions. The man who is educated does not care for money. As we progress we will find the desire for money decrease and the desire for service increase. Just imagine you were an Edison, or a Graham Bell, or a Shakespeare, ...". (Tarbell, 1937: 39).

Lo cual conecta con la otra institución clave, aparte del MIT, para entender la cultura tecnológica norteamericana: el "industrial research laboratory" fundado por un contemporáneo de Andrew Carnegie: Thomas A. Edison.

Este abrió en 1876 en Menlo Park, el primer laboratorio de investigación industrial de América y del mundo. Pero el nombre con el que lo designaba no era el de laboratorio sino el de "invention factory". Su propósito, hacer investigación y desarrollo en "any and all inventions" (Josephson, 1959:133), convirtiendo de esta forma la invención en un proceso rutinario e industrial.

Las aportaciones principales de Edison fueron dos: 1. la construcción de un laboratorio, esto es, de un centro de investigación, dedicado exclusivamente a la tarea de invención, independizando esta función de la de la ciencia y los laboratorios de las universidades pero empleando conocimientos científicos, separado por tanto de la invención artesana, meramente práctica . y 2. la conversión del nuevo tipo de invención en un procedimiento rutinario industrial, y en este sentido, ingeniero.

Carnegie Technical Schools se construyó sobre un modelo anterior al laboratorio industrial, en el que no se precisaba aún esta separación de funciones, y donde Edison y Shakespeare, el inventor y el artista, aún no formaban tipos humanos claramente diferenciados, como igualmente ocurría con figuras como Leonardo,

tipo humano que aún hoy se considera el prototipo de CMU. Como certeramente apunta Pamela McCorduck, aún en los años 70s: " Carnegie-Mellon represents old-world craftsmanship" (1979:112).

Entre el modelo de Edison y el de Andrew Carnegie existía una importante diferencia: la que mediaba entre la industria eléctrica puesta en marcha por el laboratorio industrial, y la industria siderúrgica, anterior a esta. Mientras que la novedad de la primera era la separación entre laboratorio y la cadena de montaje, en la primera, el conjunto del proceso se realizaba en el taller.

La gran industria siderúrgica de Carnegie se basó en un invento técnico específico, el proceso Bessemer de producción de acero, con las técnicas de management de los ferrocarriles de Pensilvania, construyendo la primera industria integrada del país. Su modelo era el de producción en masa a bajo precio, y era anterior al uso sistemático de los conocimientos científicos para la producción de inventos industriales. No tenía diferenciada como funciones delimitadas el laboratorio del taller, la parte pensante o "research" y la manufactura o "manufacturing".

Sus beneficios provenían ante todo de las economías de escala, no del valor añadido proporcionado por sus inventos tecnológicos. Andrew Carnegie basó su modelo de gran industria, no en el textil, sino en los métodos de administración de la red de ferrocarriles del estado, la Pennsylvania Railroad, inspirada a su vez en la organización del ejército. (Livesay, 1975:33).

Ello se refleja en la concepción original de Carnegie Tech. Mientras que el MIT se organizó como instituto de tecnología, basado en la ingeniería, Carnegie Tech pretendía formar a la masa de especialistas, los que necesitaba su modelo de gran industria siderúrgica.

Los cuatro primeras escuelas de Carnegie Tech fueron: La School of Science and Technology, destinada a formar "engineers assistants", the School of Fine and Applied Arts, para preparar en profesiones relacionadas con el diseño y sus aplicaciones industriales. La School for Apprentices and Journeymen, destinada a formar mecánicos y especialistas para la manufactura y el comercio. Y la Margaret Morrison Carnegie School for Women, que preparaba en estudios de secretaría, diseño de vestidos, economía doméstica y formación general.

El punto en común a toda esta filosofía era la idea de una educación para la "real life", dotar al estudiante de "skills" (habilidades, destrezas) para que pudiera resolver los problemas que se le iban a plantear en el mundo real, "the

real world problems".

2.3. CARNEGIE TECHNICAL SCHOOLS Y FINE ARTS.

Si bien en este enfoque hacia la educación profesional Carnegie Tech era parte del esfuerzo nacional por construir una "technical education", el modelo de Carnegie Tech, guardaba una diferencia particular: una visión de la educación técnica ligada a las artes. La visión del técnico como "craftmanship", como poseedor de "skills" o destrezas, habilidades manuales. El especialista necesitaba solo "skills". Durante décadas lo que proporcionaba Carnegie Tech fueron "tools", herramientas, concibiendo la educación como un "learning by doing", como es propio del taller artesano, en el que aprendiz sigue los pasos de su maestro que mediante la experiencia de los años ha conseguido la maestría (el "master") en el oficio. Ochenta años más tarde, ANDREW, la nueva red informática de CMU, aún se concebirá como un nuevo "tool" para el estudiante. Y Simon proponía el aprendizaje por "inmersion".

El modelo inicial de Andrew Carnegie fue modificado en la década siguiente bajo la dirección de Arthur Hamerschlag (1903-1922) en el sentido de impartir una educación técnica de rango universitario. En 1912, Carnegie Tech se convirtió de escuela técnica en instituto tecnológico con derecho a conferir grados universitarios según el modelo del MIT, cambiando su nombre de Carnegie Technical School to Carnegie Institute of Technology. Con este paso Carnegie Tech pudo otorgar los títulos de Bachelor of Science in Chemical, Civil, Electrical, Mechanical, Metallurgical or Mining Engineering, y el Bachelor of Arts in Architecture o en Interior Decoration. Se iniciaba así el periodo del "engineer", con formación científica, iniciado por el MIT en 1865.

2.4. M.I.T vs. HARVARD: EL "INSTITUTE OF TECHNOLOGY" Y LA "UTILITARIAN UNIVERSITY".

El nacimiento del MIT en 1861 vino dado por el agotamiento del modelo universitario vigente dominado por las universidades tradicionales como Harvard, Yale o Princeton. Ello provocó un proceso de reforma interna de la universidad norteamericana.

En esa época, entre 1865-1910, según Laurence R. Vesey, ("The Emergence of the American University" 1965) quedó fijado el modelo actual de la cultura académica en Estados Unidos. En este periodo surgieron tres concepciones rivales de

Universidad, que este autor denomina: "Utility, Research, y Liberal Culture" respectivamente, modelos académicos que se ofrecieron como alternativos al del clero protestante, hasta entonces dominante en la universidad norteamericana. Hasta mediados del siglo XIX la universidad norteamericana estaba dirigida por clérigos y humanistas. En 1865 nueve de cada diez presidentes de Universidad eran clérigos. Harvard fue fundada en 1636 por el clero puritano como una escuela para formar ministros de las iglesias y era la universidad más antigua y mejor dotada del país.

La autoridad en ella de la Biblia era indiscutida. En 1876, Noah Porter, presidente de Yale, afirmaba: " To Christ belongs the supreme authority in heaven and earth and...the goings on of nature and the events of human history, including the developments of science and letters, of culture and art, are all in the interest of Christ's kingdom". (cit. Veysey: 26). Las materias de estudio básicas eran el Griego, el Latín, las matemáticas y en menor medida la filosofía moral. El método de enseñanza básico , la lectura de textos clásicos, su traducción y comentario. Método que aún pervive en la expresión "lecture". Las alternativas que se ofrecieron a este modelo fueron , ateniéndonos a la división de L. Veysey tres: el "practical service", el "research" y la "culture". El primer modelo , el de "utility", o "practical service", es el menos europeo de los tres y el más anclado en la cultura norteamericana de la "applied science" iniciada por Benjamin Franklin. Por contra, el modelo de "research" se importó de Alemania, y el de "culture" era el más tradicional en Europa, el dominante tanto en las universidades inglesas como francesas.

El modelo académico universitario denominado "utility" fue iniciado por la Cornell University en 1868, bajo la presidencia de Andrew D. White y culminado por Charles W. Eliot, presidente de Harvard.

Stanford University y las Universidades del Midwest, entre ellas la Universidad de Wisconsin, fueron sus más fieles seguidoras.

El mismo Andrew Carnegie fue trustee de Cornell, y admirador de su fundador Ezra Cornell, constructor de ferrocarriles y cuáquero de confesión religiosa.

Este industrial de Nueva York quiso fundar una institución basada en la idea de "democracy", un lugar "where any person can find instruction in any study". El dictum de Ezra Cornell fue " No object of human inquiry can be out of place in the programme of a real university." (Veysey: 90). De Cornell surgiría en los años 50s, el proyecto Vicos, el primer proyecto de ciencia social aplicada,

dirigido por antropólogos, concebido como un experimento de cambio social dirigido.

Cornell era una "useful university", una agencia dedicada a promover el "individual success", con una amplia variedad de cursos donde cada estudiante tenía la posibilidad de elegir ("choice") según sus propósitos y potencialidades. Este modelo de "free elective system" aún pervive. En lugar de una elitista institución al modo de Princeton o Yale, una universidad basada en la igualdad de trato o condición de todos los estudiantes.

Las escuelas de ingeniería, educación, y business entraron en los campus universitarios con igualdad de consideración que las restantes.

Las bases de dicha "utilitarian university" eran :

1. El concepto de "real life". La educación académica tenía que servir para la vida real. Una versión extrema de esta concepción es la que adoptaron algunas universidades del Midwest, como la Universidad de Wisconsin : la universidad es un "universal public service". Esta idea denominada la "Wisconsin idea", fue formulada por Charles K. Adams, presidente de la Cornell University en 1896: "The university is not a party separate from the State. It is a part of the State...as much (sic) as the brain and the hand are parts of the body" (Veysey:104).

2. La "Social Efficiency". El problema pasó de ser "qué-enseñar" a "cómo-enseñar". El pragmatismo de William James y J. Dewey devino una expresión de este modo de pensar. Este último filósofo, profesor de la Universidad de Chicago, donde años más tarde estudiaría Herb Simon, codificaría en 1910, en How to think, ésta "problem solving mentality", estilo de pensar basado en la cultura tecnológica ingeniera norteamericana.

David S. Jordan, presidente de la Stanford University, fue también uno de los principales portavoces de esta reforma universitaria, y de la unión de la ciencia con la ingeniería.

George E. Vincent de la Universidad de Chicago definía así en 1902, la "social efficiency" : "It carries a general idea, the ability to meet situations, to solve problems, whatever they may be. Efficiency is problem-solving, adequacy. There is need of efficient persons in a world of problems. Daily life is continuous series of situations to be dealt with, problems to be solved..." (Ibid: 117).

Este modelo quedó legitimado plenamente cuando Harvard lo adoptó bajo la

presidencia del unitarianista Charles W. Eliot (1869-1909), "the first citizen of the Republic", el fundador de la moderna Harvard.

Esta universidad destaca entre todas las universidades norteamericanas por ser la institución decana entre las instituciones educativas del país, la que en su orígenes contribuyó decisivamente a la construcción del carácter nacional norteamericano. Su presidente Eliot definía así su "model of man": "We seek to train doers, achievers, men whose successful careers are much subservient to the public good. We are not interested here in producing languid observers of the world, mere spectators in the game of life, or fastidious critics of others men's labors". (cit. R. N. Smith, 1986:29).

De él afirmó el presidente Theodore Roosevelt: " It unites the assurance of the scientist and the convictions of the moralist...He is the Man Doing". (Smith: 57). Su proyecto fue convertir a Harvard en el guardian de toda la educación americana, y de su cultura.

Esta "utilitarian university" fue el modelo mayoritario en esta reforma de la Universidad norteamericana, y se levantó en clara oposición respecto a otros dos modelos copiados de la tradición europea: la "research university" y la "liberal-arts university".

2.5."RESEARCH UNIVERSITY" E "INDUSTRIAL RESEARCH LABORATORY".

Cuando se afirma que Carnegie Mellon es una "research university", un europeo tiende a pensar en el modelo de universidad alemana, basado en la ciencia pura, no-utilitaria, y que produjo a los W. Wundt, los Einsteins, los Heisenbergs, los grandes científicos de la ciencia europea contemporánea. En España este modelo de investigación inspiró la fundación del Consejo Superior de Investigaciones Científicas.

Pero la "research university" norteamericana en realidad tiene dos modelos. Uno es el modelo europeo, en particular alemán, de investigación representado por la John Hopkins University en Baltimore.

La ciencia pura en Estados Unidos, al estilo de la universidad pública europea, en la que el catedrático o "Herr Professor", con puesto de funcionario del Estado, y por tanto sin interés comercial directo, se dedica en la soledad de su laboratorio a una investigación concienzuda y de por vida a la búsqueda del descubrimiento científico per se, se desarrolló en Estados Unidos como un modelo minoritario en universidades como la citada Johns Hopkins University de

Baltimore (1876) o la Clarke University. Y como un modelo en abierta oposición a la "technical education", y a la "utilitarian university".

La Johns Hopkins University oficialmente preconizaba una "advanced instruction, not professional, to properly qualified students, in various departments of literature and science" (Veysey: 149).

G. Stanley Hall, profesor de psicología fisiológica, y su nuevo presidente, llamaba al investigador el "knight of the Holy Spirit of truth", y la universidad era considerada un medio para mantener "the holy fervor of investigation". El centro organizador de la investigación era el laboratorio, en el caso de la química, la física y la biología, y el seminario, en el caso de las ciencias sociales o la historia.

Este modelo seguido posteriormente y de forma parcial por Harvard, Chicago y Columbia. Es el preconizado como "pure research" por Vanevar Bush en 1945. Pero el modelo de "research university" de CMU, no ha salido principalmente de esa tradición de hacer ciencia sino más bien de la tradición de la "applied science" fomentada por Benjamin Franklin y del laboratorio de investigación industrial, inventado por Thomas A. Edison en 1876 en Menlo Park, New Jersey. La aportación específica a la investigación de Norteamérica es considerar "research" lo que en Europa no era más que aplicación técnica sin valor intelectual alguno: no es tanto hacer la ciencia "aplicada" sino hacer científica la actividad inventiva. Edison no fue un científico que se decidió a aplicar sus teorías, sino un "craftman", que decidió hacer de la invención un procedimiento sistemático y rutinario.

Esta diferencia entre laboratorio científico y laboratorio industrial es clave dado que conduce a dos tipos de experimentación distintas, cuyos objetivos son diversos. Lo nuevo en América es el laboratorio de invenciones tecnológicas, cuya incesante actividad de experimentación tiene como objetivo no el probar hipótesis sobre el funcionamiento de la naturaleza sino la generación de forma sistemática, "científica", de nuevos artefactos, procedimientos o métodos a cerca de como realizar determinadas tareas.

Esta distinción es la que explica la diferencia entre la ciencia de Bacon o "empirical science" y la tecnología de Edison, guiada por fines prácticos. La primera descubre, la segunda inventa. Pero lo nuevo de Edison es que la invención, no sólo era una actividad fortuita y separada de la investigación. Construyó un tipo de investigación que se basaba en producir inventos. Como

afirma su biógrafo, Matthew Josephson: "No one had ever heard of a man setting up a center of research, a sort of 'scientific factory' in which investigation by a whole group or team would be organized and directed solely toward practical inventions". (Josephson,1959:133).

Como veremos más tarde, al definir la investigación en computer science como "experimental research" se están confundiendo ambos tipos de investigación. La "basic research" en computer science puede ser de dos tipos, la que sigue el modelo de Bacon, y la que continúa el modelo de Edison. Y ambas se denominan "basic". Hay dos tipos de "basic research", cuando se dice que Japón no tiene basic research, se está refiriendo a la investigación científica básica. Pero proyectos como el del Ordenador de la Quinta Generación o el de la Human Frontier, o los que financia DARPA en Estados Unidos,son otro tipo de investigación básica, la investigación tecnológica básica, o que también podríamos llamar investigación estratégica.

Este tipo de laboratorio inventado en Norteamérica cobró rango universitario en 1913 al fundarse el Mellon Institute of Industrial Research, el primer centro de investigación industrial de Estados Unidos asociado a una universidad, instituto patrocinado por los banqueros e industriales Andrew y Richard B. Mellon.

Este tipo de investigación denominada "mission oriented", u orientada a objetivos, culminará con la II Guerra Mundial pasando a denominarse Research and Development (R+D), inaugurada entre otras por la RAND Corporation en Estados Unidos, modelo de institución que jugará un papel importante a la hora de configurar el Computer Science Department de CMU, con Allen Newell, antiguo investigador de la RAND Co. entre otros a la cabeza.

En Carnegie Mellon, en particular en la School of Computer Science, por "basic research" se entiende investigación por objetivos . Estos pueden ser estratégicos y a largo plazo, pero su matriz es marcadamente tecnológico. Por "basic research" se entiende una actividad distinta a la del clásico laboratorio de experimentos científicos. Su finalidad no es tanto probar hipótesis empíricas sino producir un sistema que funcione, para lo cual puede planificarse a 20 años vista.

2.6."LIBERAL-ARTS EDUCATION" Y "LIBERAL-PROFESSIONAL EDUCATION".

Por último, la "liberal culture university" fue un modelo de universidad norteamericana que se planteó en conflicto con los otros dos, "utility" y

"research". Yale y Princeton University fueron sus paladines. El objetivo fue implantar los valores de la civilización occidental en la mente de la juventud americana. Princeton convirtió las "humanities" en su "supreme love", en abierta oposición a los cursos vocacionales y de formación profesional. Su presidente Woodrow Wilson afirmaba que en Princeton era "not a place of special but of general education, not a place where a lad finds his profession, but a place where he finds himself" (Veysey, 1965:242). Para conseguir esto las disciplinas preferentes eran: la literatura, la historia y la ciencia política. El filósofo idealista Josiah Royce era el pensador preferido de este modelo académico, en gran parte inspirado en las universidades humanísticas francesas e inglesas. La universidad ideal debía ser una "community", basada en los mismos ideales de "co-operation", "comradeship", "loyalty", al servicio de su país y del "public service".

Elementos de esta "liberal culture" se instrumentalizaron en Carnegie Tech con el Carnegie Plan en los años 30s, y más tarde en el MIT, para proporcionar algunos conocimientos de "general culture" a los ingenieros y estudiantes de formación profesional, pero sin llegar a constituir un "core knowledge" central como es propio del modelo de universidad de Princeton o Yale, o del viejo "trivium" y "quadrivium" de las universidades medievales europeas.

El hecho más significativo de este periodo en cualquier caso no es tanto el surgimiento de estos tres modelos de universidad sino el movimiento profundo que llevo a ello: la presión de una nueva cultura tecnológica en la educación y en la investigación en Norteamérica que tuvo distintas expresiones:

1. el surgimiento de institutos tecnológicos como el MIT (1865) al margen de la universidad patrón de las universidades en Norteamérica, Harvard.
2. El surgimiento del laboratorio de investigación industrial (1876) formado por Edison, al margen del tradicional laboratorio de investigación experimental universitario de escasa importancia en Estados Unidos.
3. El nacimiento de escuelas técnicas de grado medio, como las Carnegie Technical Schools, para formar a la gran masa de mecánicos y personal especializado que necesitaba la industria de este país.

En palabras de Dwight Baumann, profesor of Engineering Design en CMU: "I believe that the basic motivation force for the development of the profession engineering, stemmed from the era of the American inventor and in part from the main flow of the industrial revolution itself. The American inventors, Edison,

the Wright brothers, Eli Whitney, George Westinghouse, Bell and the rest became national heroes in their time..." (Focus. 1973. March 19).

Esta cultura académica tecnológica era vista por primera vez por sus protagonistas como distinta y superior a la "liberal education". Como afirmaba el Presidente del MIT en 1891: "Too long have we submitted to be considered as furnishing something which is, indeed, more immediately and practically useful than the so-called liberal education, but which is, after all, less noble and fine. Too long have our schools of applied science and technology been regarded as affording an inferior substitute for classical colleges. Too long have the graduates of such schools been spoken of as though they had acquired the arts of livelihood at some sacrifice of mental development, intellectual culture, and grace of life... I believe that in the schools of applied science and technology is to be found the perfection of education for young men". (Noble: 25)

Aún hoy día un ingeniero de CMU como el profesor Norman Chigier, William J. Brown Professor of Mechanical Engineering, considera que su profesión es más compleja aún que la propia ciencia: "I've tried to demonstrate to scientists and even to fine artists how creativity and imagination are extremely important in engineering design. Engineering is really more complex than science. It can't obey all the rules of science because it deals with systems that are less easy to analyze. Engineering have the challenge of working with practical systems --end-products" (Focus. 1983: Feb.).

Esta cultura tecnológica, expresión de cambios decisivos en el país tras la Guerra Civil y su industrialización masiva, es la que presionó a la universidad norteamericana para su transformación en una "utilitarian university" lo que logró en su mayor parte, consiguiendo la formación de escuelas profesionales como la ingeniería o el management en el interior de universidades tradicionales como Harvard o Yale.

Este modelo de universidad es específico norteamericano y permite entender el por qué, con algunas excepciones como el MIT o Carnegie Tech, no existe en Estados Unidos una doble estructura universitaria como en Europa: la Politécnica y la Científico-Humanística, si bien se puede distinguir una doble cultura académica dentro de las universidades, aquellas que proceden de una tradición de "liberal-arts colleges" como Yale o Harvard, de aquellas como CMU que proceden de un instituto tecnológico reconvertido en universidad.

Este modelo de universidades unificadas explica que un Instituto Tecnológico

como Carnegie Tech se pudiera transformar en 1967 en una universidad conservando su cultura tecnológica, y acabara construyendo una "computer university", una universidad donde la máquina herramienta del taller industrial, se ha transformado en el ordenador, concebido como un útil usado hoy en todas las facultades.

El modelo aún mayoritario de universidades norteamericanas son universidades que han incorporado elementos de una cultura tecnológica en su interior, pero de forma subordinada a lo fundamental de los patrones de la cultura occidental, los patrones de la concepción científica del mundo que permanece incambiable desde hace más de veinte siglos.

Los institutos de tecnología como el MIT o Carnegie Tech que se han adaptado al modelo universitario han tendido a priorizar la educación científica sobre la tecnológica, lo cual ha redundado en una pérdida de la sustancia teórica de la ingeniería, el diseño. Como se queja el informe Made in America del MIT: " The retreat of this activities has its origins partly in the wartime experience itself and especially in the dominant role played by physicists and other natural scientists in the development of nuclear weapons..But by now the pendulum has probably swung too far from real-world problem solving, especially as it relates to industrial production" (Dertouzos,1989:78-79).

Con todo, es en universidades como CMU que eran hasta hace poco un instituto de tecnología, con limitada tradición científico-humanística, donde podemos encontrar ese estilo profesional de pensar, propio de una cultura académica tecnológica, a los que están subordinadas formas de pensar científico-humanísticas fundamentales de la cultura occidental. Y es en este tipo de instituciones, como CMU, donde más se ha avanzado en transformar su propio campus en una "computer culture" donde se presiona a los profesores a transformar su conocimiento de la búsqueda de la verdad a la construcción de un programa informático que funcione.

Esta cultura tecnológica ha sufrido cambios importantes. En los años 30s, en plena Depresión, el modelo de "professional education" iniciado por Andrew Carnegie fue modificado con el llamado Plan Doherty (1936-1950). Este plan introdujo dos modificaciones educativas principales, 1. la introducción del problem solving como "engineering method" . 2. la " humanistic-social stem" o nuevo tipo de formación humanística para los ingenieros, doble cambio que puso a Carnegie Tech a la vanguardia de la educación tecnológica en este país.

3. "THE CARNEGIE PLAN OF PROFESSIONAL EDUCATION".

3.1. LA "HUMANISTIC-SOCIAL STEM": "THE LIBERAL CULTURE" EN UN INSTITUTO TECNOLÓGICO.

La Depresión económica de los años 30s significó el fin de la Progressive Era, del liberalismo económico, y de la autosuficiencia de la educación meramente técnica del ingeniero profesional. La invención de nuevos artefactos no garantizaba por sí sola la prosperidad económica. Había que tratar con el factor humano, crecientemente motivo de conflicto. Franklin D. Roosevelt iniciaba el New Deal. Se abría la era del "manager", del "planning", del ingeniero convertido en director de empresa y del planificador social (Hughes, 1989: 335).

Es la época del Hawthorne project de Elton Mayo sobre las actitudes y la productividad de los trabajadores en la Western Electric Company, del Tennessee Valley Project de Lewis Mumford, como experimento piloto de planificación regional, en suma de las "human relations" como nuevo tipo de "engineering problem".

En Carnegie Tech esta era estuvo dominada por el "Carnegie Plan of Professional Education" iniciado por el Presidente Doherty en 1936 y que constituyó una reforma de la educación ingeniera que sirvió de modelo para el resto de escuelas de ingeniería del país. Esta etapa culminó con la fundación de la Graduate School of Industrial Administration en 1949, auspiciada por el Provost Elliot D. Smith, y financiada por William Larimer Mellon, presidente de la Gulf Oil Company.

Herb Simon, uno de sus fundadores, experto en teoría de la organización, que había dirigido varios estudios sobre administración municipal ve así las aportaciones de este periodo: "Carnegie had pioneered in two important movements in engineering education: providing a substantial liberal arts component within the engineering curriculum, and shifting emphasis from teaching subject matter to teaching problem-solving skills" (Models of My Life: 264).

Estas dos innovaciones van a tener consecuencias en la formación, años más tarde, de una específica "computer culture" de CMU, donde, a diferencia de otras universidades o institutos tecnológicos como el MIT, esta cultura se inició, al menos en su 50%, por científicos sociales como Herb Simon que llegó a Carnegie Tech en 1949 para formar esa escuela de managers. Por otra parte, la "computer culture" de CMU hará suya la "problem solving mentality" ingeniera, asentada en

Carnegie Tech como un estilo de pensar propio de una escuela de formación profesional, tratando de convertirla en un "human problem solving" o un tipo de mentalidad no sólo ya del profesional sino del conjunto de disciplinas académicas.

El Carnegie Plan of Education fue el marco en el que se realizaron estas innovaciones. Este plan tuvo dos etapas distintas, separadas por la II Guerra Mundial.

En su primera etapa fue iniciado como un simple plan de reforma de la educación ingeniera, por el entonces Presidente Robert E. Doherty (1936-1950), ingeniero eléctrico en el laboratorio industrial de la General Electric durante 20 años , y decano de la Escuela de Ingeniería de Yale University de 1931 a 1936. Fue el denominado "Social Relations Program".

Su "proposal" sometida a la Maurice and Laura Falk Foundation en 1937 para obtener fondos para este programa lo justificaba así: "This country needs more professional men who can grapple intelligently with intricate social problems. Technology with all its blessings yet subverts traditional habits and attitudes of mind; narrow specialists multiply endlessly...Engineers who understand technology, must give more thoughts to consequences. They, in common with lawyers, economists and other responsible professional people, must consider a new kind of educational preparation for their oncoming generations..." (Cleeton, 1965. :99).

En el libro "The Development of Professional Education, The Principles which have guided the Reconstruction of Education at Carnegie Institute of Technology as Stated by Its President", editado en 1950, y que recoge una selección de artículos desde 1936, podemos encontrar en el primero de ellos los tres cambios fundamentales propuso en el curriculum de la School of Engineering and Sciences:

"First, a new philosophy and new outlook which will comprehend the human and social as well as the technical. Second, the development in all professional men of genuine competence in the professional way of thought,--a way of thought which embodies an analytical and creative power that is as effective in the human and social realm as that developed in engineering in the application of the physical sciences....Third, the development of the ability to learn from experience so that in the unfolding future they can continue to expand their fundamental knowledge, deepen their understanding, and improve their power as

professional men and women and as leading citizens". (1950:2-3).

El primer objetivo se concretó en una revisión del curriculum de la escuela de ingeniería introduciendo el "Social Relations Program"(1938), y estableciendo una cuarta parte del tiempo para estudios de humanidades y ciencias sociales. Este programa consistía en un conjunto de asignaturas de humanidades y ciencias sociales, como inglés, historia, economía, sicología industrial ("human relations in industry") literatura y arte, que dieran una visión más amplia al estudiante de ingeniería. El nombre del experimento se denominó: "Carnegie Plan of Professional Education in Engineering and Science". Como reconoce Glen Cleeton, esta reforma estaba inspirada por pensadores sociales de la tecnología de entonces como Lewis Mumford, Charles Beard, J.D. Bernal, Elton Mayo y John Dewey. Una idea común de toda esta reforma era de "education for social responsibility" (Doherty, 1950:13).

Se estableció la Maurice Falk Professorship of Social Relations ocupada por el economista Willard Hotchkiss. Y en 1941 se creó una División of Humanistic and Social Studies. Este programa era el más amplio existente en Estados Unidos para escuelas de ingeniería. Sólo a mediados de los 50s, el MIT comenzó una revisión del curriculum similar a la realizada cerca de 20 años antes por Carnegie Tech. Este rasgo tendrá más tarde repercusiones en la influencia de científicos sociales en el establecimiento de la "computer science" , y en particular, de la Inteligencia Artificial, en CMU, aventajando en este campo al M.I.T. y a Stanford.

Por primera vez desde su fundación, Carnegie Tech reconocía dos bandas o "stems" de estudio: "the technological and the social" (Cleeton:95). La creación de la Division of Humanistic and Social Studies favorecía que materias como la Historia, el Inglés o la Economía pasasen de ser meros "servicios" de la educación ingeniera a fines en sí mismos.

Este cambio, origen de la idea de "liberal-professional education" iniciará en Carnegie Tech un conflicto que aún pervive. Un reflejo de ella es la división por ejemplo de la Inteligencia Artificial en dos areas una en Computer Science y otra en el College of Humanities and Social Sciences. No obstante, la idea inicial del Carnegie Plan sólo habla de "professional education".

3.2. "THE PROFESSIONAL STYLE OF THOUGHT".

La Segunda Guerra Mundial abrió una segunda etapa al Carnegie Plan, donde los

científicos sociales adquirieron más relevancia, en especial los economistas.

Por primera vez en la historia de Carnegie Tech se elegía para el cargo de Provost a un abogado de profesión, Elliott Dunlap Smith (1945-58), hombre con larga experiencia en empresa y profesor de Social Relations en Yale. Su papel fue clave en la definición de la segunda etapa del Carnegie Plan, que coincidió con la construcción de la escuela que iba a encarnar parcialmente estos ideales: la Graduate School of Industrial Administration, fundada en 1949.

Discípulo de John Dewey, filósofo y educador pragmatista norteamericano, introdujo en la cultura académica de Carnegie Tech la idea que el "problem solving" podía proveer un lenguaje común tanto para el historiador como para el ingeniero, esta idea condujo a una particular forma de entender la interdisciplinariedad en Carnegie Tech.

Hasta entonces, el "problem solving" se había entendido como el "engineering method". Elliot D. Smith tratará de convertirlo en el estilo de pensar del profesional, y más en concreto del manager empresarial, tipo humano a formar en la Graduate School of Industrial Administration de Carnegie Tech.

En el artículo "Professional Discipline Versus General Education as a Preparation for Citizenship and Cultivated Living" (1948), Smith expresaba su filosofía particular: "If our country's most disciplined thinkers --its professional men-- to devote their moral energies and intellectual powers to solving the problems of our perilously free society, education must more effectively prepare them for civic responsibility that it has in the past...Hence proper humanistic and social, as well as technical, instruction is professional in fact, and can be made interrelated aspects of a common professional education...The solving of problems in machine design, in economics, or in English composition all involve a fundamental unity of approach, each contributes a special influence to developing well-rounded analytical and creative power". (1948:2).

Ya en 1946, en la colección denominada Carnegie Institute of Technology: Educational Papers, que resumía los trabajos en torno a la revisión del Carnegie Plan, Elliot D. Smith en su artículo "Professional Education in a Free Society" apuntaba la relación instrumental ("tool relationship") entre conocimientos que se había empezado a desarrollar en diversas áreas en la institución.

" Last year Carnegie Institute of Technology set out to integrate the apparently dispersed and separate courses in its program of general education by examining

each course in that program to see how far it could be useful in learning subsequent courses and also to see how far it could carry forward prior learning by using what was previous taught. Although it was hard, difficult work, the reward was atonishing great, for as we proceeded it became more and more clear that courses widely differing in field can be effectively brought together, not in surface particulars but at their roots..." (1946:6). Esta etapa de la institución es clave para la formación de un rasgo característico de Carnegie Mellon, y de la School of Computer Science: la interdisciplinariedad o "collaboration accross boundaries". Pero un cambio radical se producirá en la generación siguiente. En 1977, Herbert Simon daba una charla en CMU bajo el título : Liberal Education in A Technological Society" (1977. Focus, April 5). En ella también se planteaba esta colaboración pero bajo la dirección de la "liberal education". Justo el proyecto inverso del de Dunlap Smith.

Este intento de Smith tendrá repercusiones importantes, en el propio Herbert Simon, a la hora de entender los continuos esfuerzos realizados a partir de ese momento en Carnegie Tech por, extender "el problem solving" como "professional method" al conjunto de disciplinas. El "human problem solving" (1972) refleja esta filosofía. Mas tarde, "problem solving" se substituirá por el tema del "design", tratándolo de hacer una actividad propia no sólo de la ingeniería (Simon, 1973). De aquí nacera una filosofía del "Putting It All Together", para encontrar una vía de unificar, integrar y comunicar diversas disciplinas, especialidades y materias en un campus que se convirtió en la cuna de la Inteligencia Artificial nueve años más tarde, construyendo una específica combinación de conocimientos para resolver un problema significativo, el de como hacer pensar a una máquina.

No obstante, la generación siguiente a la del Carnegie Plan, entendió que esta unificación se podía hacer bajo el manto de la ciencia, que era posible hacer una "science of design". En lugar de una "professional education" con una "liberal education" de complemento, pretendió revertir el modelo hacia una "liberal-professional education", donde la segunda era un complemento de la primera.

Inicialmente la vía seguida por Dunlap Smith fue, en lugar de analizar conocimiento por conocimiento por separado , ponerlos en una relación instrumental entre sí con el objeto de resolver determinado problema significativo: "It is essential...not to teach even the fundamentals as ends in

themselves but to teach them as means of solving significant problems. For general as well as specialized education becomes cultural only when it becomes useful as a means of enlightening further learning. " (1946:5).

Así la Historia de la Civilización Occidental podía ser enseñada no tanto como un "fin en sí mismo" sino como medio de enseñar a los estudiantes los orígenes y el desarrollo de instituciones sociales con las que los estudiantes iban a tratar de forma distinta en economía. O las matemática y la ciencia "can be taught as basic tools available for all technical learning".

La conclusión a la que llegaron fue que " the most important and most general interrelation between fields of learning, however, was found to lie in the fact that a basic form of scholarly thought is common to all. After much study a committee representing scientists, historians, mathematicians, English scholars came to unanimous agreement as to what were the essential elements of well-ordered problem solving". (1946:7).

Colocando al estudiante ante problemas de la vida real a resolver, se trataba de provocar en él el deseo de diseñar algún plan o planes que sirvieran de guías para su solución, en lugar de lanzarse a ciegas. Este plan consistía en el diseño de los medios necesarios para conseguir los fines propuestos y la serie ordenada de pasos a seguir para lograrlo. Esta era un "style of thought" planteado tanto para los cursos profesionales (ingeniería) como para los generales (humanidades).

"If professional skill, by being freed from professional boundaries, is made available for the understanding of human and social problems and if in addition the student becomes accustomed to using that skill in his thinking about human and social relations whatever thereafter increases his capacity in his profession will increase his stature as a person and his value as a citizen". (1946:8).

El problem-solving era para Elliot Smith pues el punto de unión de la educación especializada y la educación general, el tipo de pensamiento capaz de preparar a los estudiantes para progresar en una sociedad en continuo y acelerado cambio: " A free society can exist only if its citizens can think well for themselves in solving its problems" (1946:11).

"Problem-solving" era pues para Smith el conocimiento básico de la "professional education in a free society", el núcleo del Carnegie Plan of Professional Education. Pero esta fue su lectura de dicho plan. Pero las tendencias

dominantes no la confirmarían.

Este momento de la historia de Carnegie Mellon no ha sido aún valorado con suficiente relevancia. El que por primera vez especialistas de materias tan distintas como la historia o las matemáticas puedan acordar un tipo de metodología común de enseñanza en torno al "problem-solving", y decidan acordar un interés en esta "basic form of scholarly thought", puede significar el punto de partida de la lo que podríamos denominar el "core knowledge" de una cultura tecnológica o tecnocultura.

3.3. LA GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION : DOS CULTURAS ACADEMICAS EN CONFLICTO.

El test del Carnegie Plan fue la Graduate School of Industrial Administration.

Elliot D. Smith fue su promotor principal. Iba a ser la escuela donde se produciría la síntesis entre la educación ingeniera , científica y la humanístico-social, bajo la metodología del "problem solving".

En 1948, el Carnegie Institute of Technology recibió una donación de 6 millones de dólares de William Larimer Mellon, fundador de la Gulf Oil Company e hijo del banquero Thomas Mellon, que inició la dinastía de una de las familias económicamente más poderosas de Estados Unidos. Según relata Herb Simon en su autobiografía, este empresario había llegado a la conclusión que las empresas modernas de alta tecnología necesitaban altos ejecutivos que fueran a la vez expertos en management y entendidos en ciencia y tecnología. (Models of My Life.1991:136). El provost Smith le presentó su programa de "industrial management" como base de lo que podría ser el centro y consiguió los fondos del empresario.

En su fundación, el Presidente Doherty afirmaba : "The new School will aim to produce men who have the fundamental qualifications for professional growth to leadership in the field of industrial administration and who will be active, intelligent citizens". (Cleeton: 88).

Pero al poco tiempo de su fundación se presentó un conflicto en la GSIA entre dos tipos de culturas académicas distintas: la cultura tecnológica y la "liberal-arts culture". Herbert Simon en su autobiografía denomina a este conflicto "la crisis de 1951", y a efectos de este estudio es muy significativa.

La GSIA fue creada inicialmente dentro de una "problem solving mentality". Así era vista por la revista Business Week, el 17 de septiembre de 1949: "Next

week...Pittsburgh's Carnegie Institute of Technology will begin an experiment aimed at turning out young, experienced executives...This means that the student will learn chiefly by solving a series of planned problems in the professional manner. That is, he must accurately define a problem after it has been presented to him in general form. He must then look for and organize data and information, and plan a way to solve the problem, test the solution, and make a generalization, about what he has learned in doing it. By thus working out the whole thing on his own hook, Carnegie believes a student learns best by hard, down-to-earth experience".

El punto de partida de la GSIA tenía como recuerda Simon (Models of My Life, 1991:138) una doble herencia: 1. El curriculum sobre "industrial management" (lease "industrial engineering") elaborado por Elliot Smith, y 2. El equipo de economistas del departamento de Economía de la División de Humanities y Social Sciences, dirigido por Lee Bach, nombrado a su vez decano de la nueva GSIA. La crisis citada nacerá de la diferencia entre fines y propósitos de ambas herencias.

Como relata Simon, a excepción de Elliot Smith que había sido jefe de personal para la Dennison Manufacturing Co., ninguno de los restantes fundadores de la escuela tenían experiencia en el mundo de la empresa o del management. Bill Cooper, Lee Bach y Herb Simon eran ,según este último: "social scientists". (Models:138).

Ambos tres provenían de la "liberal arts culture", en concreto de la Universidad de Chicago, pero, en palabras del propio Simon, conscientes de estar en el marco de una cultura distinta, una cultura académica tecnológica, habían aceptado "the premises of the other culture" (Models:159).

Su estrategia era transformar el vocacionalismo de la "business education" norteamericana, dominado por la Harvard Business School, en un "science-based professionalism, as medicine and engineering had been transformed a generation or two earlier" (Models:139). Este cambio consistía en un nuevo método de trabajo, la sustitución del "case study" propio de Harvard por métodos cuantitativos y formalizados: más en concreto, "the use of quantitative tools for managerial problem solving and decision making", herramientas como la llamada programación lineal o la investigación operativa.

Esta estrategia es muy significativa pues explica la línea inicial de trabajo de Herb Simon en CMU durante 40 años : tratar de transformar al "professional", de

un mero artesano o "craftman" en un profesional científico. A cambio, el científico se transformaba en un diseñador, en un "problem solver".

El conflicto de 1951 en el GSIA estuvo, según la versión de Simon, ocasionado por las resistencias de los profesores de economía de cambiar su investigación en una dirección de interés para una escuela de negocios. (Models:143).

Problemas surgidos de esta relación fueron: la decantación de estos profesores por las técnicas abstractas matemáticas, a veces con mayor énfasis en la matemática más que en sus aplicaciones para el management, la oposición entre la "ciencia pura" de la economía y las aplicaciones en el "real-world" propias de una escuela empresarial, y, por último, la oposición entre las teorías de la racionalidad neoclásica y las del "decision making", estas últimas adoptadas por Simon. (Models:146-47)

Lo interesante de este conflicto es que es visto claramente por uno de sus principales participantes como un conflicto de culturas académicas distintas:

"So part of what we were seeing, in the tension within GSIA, was a conflict between two academic cultures that had different definitions of the academic role and of academic freedom" (Models:159). Estas distintas culturas no eran otras que las del un "institute of technology" y las de la "liberal-arts tradition". El triunfo de la segunda cultura sobre la primera responde al giro científico de la cultura tecnológica norteamericana en los últimos 30 años, y este giro transformara este instituto tecnológico en una "university" en 1967.

Este conflicto no es distinto al que dió nacimiento a la propia Carnegie Technical Schools con Andrew Carnegie. Pero en inicialmente esta institución nació separándose de la "liberal education". Ahora el proceso se invertía.

El conflicto se resolvió con la victoria en los años 60s de los "economists faculty" y la derrota de los "problem solvers". Así explica Simon el final de la crisis: "By the early 1960s, the Golden Age of organization theory and the behavioral theory of the firm had ended at Carnegie Institute of Technology...GSIA became dominated by research on sophisticated mathematical techniques in operations research and economics and by neoclassical economic theory. The economists' aborted revolution of 1951 achieved a large measure of success in the 1960s" (Models:164).

Paradójicamente, lo que empezó siendo la institución destinada a encarnar los principios de una "professional education", base de la cultura tecnológica, acabó por adaptando patrones de la ciencia social por antonomasia, la economía

política.

Este fenómeno no es nuevo. Simon en su libro "The Sciences of the Artificial" detectó esta tendencia general de absorción de las escuelas profesionales en la "general culture of the university" (1981:130). En los años 30s, ya había ocurrido con las escuelas de ingeniería como el MIT, donde la reforma de Karl Taylor Compton, profesor de física y presidente de dicho instituto, se había basado en aumentar la enseñanza de las ciencias naturales. "Engineering schools have become schools of physics and mathematics; medical schools have become schools of biological science; business schools have become schools of finite mathematics" (ibid:130).

Sin embargo, Simon no advierte una tendencia que contrarresta la anterior y de la que él mismo, como representante de toda una generación de científicos norteamericanos: la tendencia a la transformación de las disciplinas científicas y humanísticas en disciplinas tecnológicas, basadas en el diseño, y con fines estratégico-políticos.

Las décadas inmediatamente posteriores a la II Guerra Mundial, a partir sobre todo del Proyecto Manhattan, en el denominado "Manhattan Engineering District (Hughes,1989:353), físicos, matemáticos e ingenieros trabajaron al unísono para fines aplicados, principalmente para la Defensa. Esta actividad se definió como Investigación y Desarrollo (R&D): la ciencia se aplicaba, no sólo ya para la industria, sino en todos los fines económicos, militares y políticos de una sociedad. Se inicia el diseño consciente de cambios culturales. Es la época del Proyecto Vicos (1955) dirigido por el antropólogo aplicado A. Holmerg de la Cornell University, un experimento muy debatido pero que marcará un hito en la ciencia social, al demostrar la posibilidad de un diseño planificado del cambio social.

En este contexto, nace el Computer Science, como nueva disciplina, gracias,entre otros, a los trabajos del matemático Von Neumann, colaborador en el Manhattan Project y partidario de la utilización de la matemática para fines prácticos. En sus artículos, con Burks y Goldstine sobre "Preliminary Discussion of the Logical Design of An Electronic Computing Instrument"(1943). Von Neumann, siguiendo el ejemplo de Turing, inició el diseño lógico de un nuevo tipo de máquinas denominadas ordenadores. En CMU se le considera, junto con Turing, uno de los fundadores del Computer Science. Ambos trabajaron para la Defensa durante la II Guerra Mundial. Y su investigación aplicada no se puede entender al margen

de su carácter estratégico.

4. CMU COMO "RESEARCH UNIVERSITY": "DEL COMPUTER SCIENCE DEPARTMENT" A LA "COMPUTER INTENSIVE UNIVERSITY".

4.1. CARNEGIE MELLON Y LA "CYBERNETIC REVOLUTION".

El Carnegie Plan introdujo una colaboración conflictiva entre culturas académicas distintas, la "profesional" y "la liberal-arts culture" en Carnegie Mellon que dura hasta hoy. Pero estas culturas no han permanecido estáticas, sino que en su mutuo contacto han cambiado, se han transferido conocimientos unas a otras, produciendo una paradójica y compleja situación.

Por un lado, tras la II guerra Mundial, ha avanzado la tendencia hacia la transformación de las antiguas disciplinas tecnológicas, como la ingeniería ("engineering science") y el management, en materias cada vez más científicas y formalizadas. Esta tendencia es clave para entender conceptos como el de "information" de C. Shannon, ingeniero eléctrico del M.I.T., que tienen una base tecnológica directa.

A esta evolución influyó el crecimiento durante y sobre todo después de la II Guerra Mundial, del prestigio de la ciencia en Estados Unidos, sobre todo de la física. Es la época de Vannevar Bush, asesor científico del Presidente Roosevelt, y su libro "Science: the Endless Frontier" (1945). Como recuerda Richard Teare, decano de la escuela de Engineering and Science de Carnegie Tech en ese tiempo: "World War Two pushed things in the direction of science and theory, and in a certain sense, you could say that engineering analysis was pulling back from theory as an end in itself" (Focus, Feb,5,1980). Esta tendencia hacia la formación de una teoría ingeniera en sí misma ha sido poco estudiada y es clave para el desarrollo de una teoría de las tecnoculturas.

Durante la II Guerra Mundial, también avanzó un fenómeno inverso: la progresiva transformación de la matemática, las ciencias naturales y sociales en Estados Unidos en áreas con una orientación hacia la "problem solving mentality", hacia la aplicación, y el diseño de ingenios mecánicos, ligados a objetivos estratégicos, dentro de una "mission oriented research".

En gran parte la colaboración en el esfuerzo militar de toda una generación de científicos como Wiener, Von Newman, Turing, favoreció el descubrimiento en sus trabajos de una "purposeful mentality". La cibernética de Wiener, nacida del

artículo de A. Rosenbluth, N. Wiener y J. Bigelow, "Behavior, Purpose and Teleology" (1943), la teoría de juegos y la *operations research*, el diseño de máquinas lógicas de Turing, la teoría de la información de Shannon basada en la ingeniería, la relectura de Simon y Newell de trabajos sobre *problem solving*, la obra del psicólogo George P. Miller, junto con E. Galanter y K.H. Pribram, "Plans and Structure of the Behavior" (1956), que inicia la llamada "revolución cognitiva", forman parte de una atmósfera cultural de una ciencia dedicada, sobre todo en Norteamérica, a fines prácticos, a una investigación por objetivos, planificada y con vistas más a fines políticos o militares que a la clásica misión de la Academia de descubrir la verdad.

Incluso las ciencias sociales no han escapado a esta tendencia. En la antropología, durante la II Guerra Mundial, la plana mayor de la disciplina en Estados Unidos, Mead, Bateson, Gorer, Benedict, Leighton, Kluchhohn, trabajó para el Department of War. (M. Mead. *The uses of anthropology in World War II and after*. 1979, Eddy, E. & Partridge, W. 1987.). El experimento Vicos en 1955 marcó esta época.

Para Herbert Simon estos fueron los años, en torno a la II Guerra Mundial, de formación de una nueva "scientific culture" - "an interlocking network of scientists with a real sense of community, which was almost independent of the special area in which each worked, and which ignored the diversity of their backgrounds and training. They came from physics, statistics, economics, biology, mathematics, engineering, philosophy, and even a few from psychology and political science" (Models. 1991:107). Él tuvo acceso a esta cultura a través de dos vías: su participación en la "engineering culture" del Illinois Institute of Technology donde era profesor y de la "econometric culture" de la Commission Cowles. (ibid: 107).

Según Simon, la cibernética ("cybernetics") se convirtió en la disciplina central de esta cultura, entendiendo por ella tres aspectos: la teoría de la información, la teoría de los sistemas de retroalimentación (feedback) y el ordenador digital. (1972. Newell & Simon: 878-879).

Pero lo que no dice Simon es que estos tres aportes fueron antes tecnológicos que científicos, nacieron en centros tecnológicos como los Bells Laboratories (Shannon), el M.I.T. (Bigelow, Wiener) y la Moore Engineering School (Eckert, Mauchly). La "Cybernetics Zeitgeist" fue inicialmente una cultura ingeniera teorizada matemáticamente.

La "mathematical theory of information" de Claude Shannon y Warren Weaver, nació en los Bell Laboratories, con los trabajos del ingeniero C. Shannon sobre problemas relacionados con la telegrafía de comunicaciones. Lo que este ingeniero hizo fue dar una nueva definición de información que la transformó en un parámetro físico capaz de ser cuantificable. Para ello separó el concepto de "information" (información) del de "meaning" (significado). Si nos quedamos aquí advertimos que esa es la operación puramente matemática, para lo que se requiere simplemente la lógica matemática. Pero Shannon no era un lógico matemático sino un ingeniero de comunicaciones: "This word information in communication theory relates not so much to what you do say, as to what you could say. That is, information is a measure of one's freedom of choice when one selects a message." (Shannon & Weaver: 1949:8-9).

Por su parte, el centro de la cibernética era la teoría del control basada en el mecanismo ingeniero del "feedback", o retroalimentación, estudiado en el M.I.T por el matemático N. Wiener y el ingeniero J. Bigelow para el diseño de mecanismo de disparo antiaéreo. De hecho el libro de Wiener "Cybernetics" tiene como subtítulo "control and communication in the animal and the machine". (1948).

Como explica el propio Wiener en su "Cibernética": "Al estallar el conflicto, el prestigio de la aviación alemana y el papel defensivo de la inglesa se tradujo en la atención de muchos científicos por el perfeccionamiento de la artillería antiaérea. Antes de la ruptura de hostilidades se había evidenciado que, debido a la rapidez del avión, todos los métodos clásicos de dirección de tiro quedaban anticuados,...Es de suma importancia disparar el misil, no contra el blanco sino de modo que misil y blanco coincidan en un determinado momento futuro en el espacio. Para ello había que encontrar un método para predecir la posición futura del avión..." (1985:27).

A través del estudio del comportamiento del artillero, el soldado acoplado al sistema de control de tiro, que luego inspirará el propio nombre de la disciplina ("cibernética", del griego, kibernetes, el timonel, el que controla el barco), Wiener y Bigelow llegaron al descubrimiento de la actividad voluntaria humana ("Behavior, Purpose and Teleology." 1943), y de ella, por un símil a lo que los ingenieros de control denominaban mecanismo de "feedback": " Bigelow y yo llegamos a la conclusión de que un factor sumamente importante en la actividad voluntaria es lo que los ingenieros de control denominan feedback

...Cuando deseamos que un movimiento siga un patrón determinado, la diferencia entre este patrón y el movimiento realmente efectuado se utiliza como nuevo impulso para que la parte regulada se mueva de tal modo que su movimiento se aproxime más al previsto por el patrón". (1985 :29). La cibernética tratará con este mecanismo mecanico-eléctrico de en palabras de Wiener: "suplantar una función específicamente humana...la predicción del futuro". (ibid.:29).

Finalmente el primer ordenador digital, el ENIAC, inventado en 1944, por el físico John M. Mauchl y por el ingeniero electrónico John Presper Eckert en la Moore School of Electrical Engineering de la Universidad de Pennsylvania, y bajo contrato del Army Ordonance Department, fue un producto de una investigación con fines militares (Stern, 1981). Su primera utilidad fuera el realizar complejos cálculos. Pero pronto se empezó a considerarlo un "general-purpose instrument", una máquina programable para cualquier propósito (Von Neumann).

El nacimiento del Computer Science como nuevo tipo de disciplina fue el resultado final de esta tendencia, en la que el matemático Von Neumann, jugo un papel destacado. El computer scientist nació , ante todo, como un programador de sistemas artificiales. Pero en cierta forma en un momento ulterior del esfuerzo bélico, en el que la ciencia tendería a separarse de la tutela de la ingeniería. Y convertir lo que habían nacido como teorías ingenieras ("information", "control") en teorías científicas. Es el paso del "computer" (Mauchly, Eckert) al "computer science" (Von Neumann).

La "computer culture" en CMU participa de ambas tendencias. Por un lado, al estar dependiendo de la Defensa, ha de mantener una orientación ingeniera o "mission oriented". Pero por otro lado, se desarrolla en un periodo (años 50s-60s) donde una vez finalizada la II Guerra Mundial, los científicos tomarán protagonismo sobre los ingenieros y se establecerá el sistema Science and Technology, tendencia que finalmente dará a los matemáticos y lógicos predominio en la nueva disciplina de la Computer Science, que nació de una mera tecnología con fines militares.

En CMU confluye un tipo de investigación por objetivos, tecnológicamente innovadora, interdisciplinar y estratégica es el que está en la base de la "computer culture" de CMU. Ello vienen reforzado por la cultura ingeniera de esta institución. Pero, al mismo tiempo, CMU va a convertirse tras la II Guerra Mundial en una universidad, siendo presionada para adquirir la cultura de la misma a la que antes se refería Herbert Simon, uno de los principales promotores

de la misma en esta institución.

En Computer Science se condensa esta doble cultura contradictoria que se ha generado en Estados Unidos en estos últimos 30 años. Es una ciencia de una tecnología, construida después de inventada ésta, pero que pretende liderar su desarrollo, y competir en el universo de las otras ciencias.

Dado que la "computer science" es quizá una disciplina directamente afectada por intereses estratégicos, el empleo de científicos a gran escala para la investigación aplicada que se inició en la II Guerra Mundial ha podido pervivir durante los 45 años de la Guerra Fría en áreas como la Computer Science, dando lugar a un fenómeno inusual: lo que fue en otras disciplinas, como las ciencias sociales un fenómeno temporal, la transformación del científico en un "problem solver" o en un "ingeniero social", en el ordenador se ha consagrado como una especialidad nueva: el "científico del diseño" o "computer scientist", un científico que a su vez es ingeniero, de forma permanente.

Finalizada la Guerra Fría este matrimonio anómalo puede disolverse en sus componentes iniciales: matemáticas e ingeniería.

Durante los pasados 40 años la libertad del científico que se dedicaba a computer science era ilimitada con tal que produjera resultados, sin importar si su trabajo se ajustaba o no a los patrones de la Academia. No era ante ésta ante la que tenía que responder sino ante agencias como DARPA, que a su vez gozaban de una gran libertad de asignación de recursos.

Carnegie Tech participó de ambas tendencias: la Computer Science la iniciaron en 1955-56 científicos interesados en el ordenador como Herb Simon, Allen Newell y Alan Perlis, reunidos en torno a un ordenador IBM 650 en el Computation Center situado en la Graduate School of Industrial Administration.

La transformación de Carnegie Tech de una escuela de educación de ingenieros a una universidad fundamentalmente de investigación se inicia en torno a la investigación en esa área, denominada por unos "information processing" y por otros "computer science".

En 25 años, Carnegie Tech se ha convertido en una "research university", gracias principalmente a esa investigación en informática.

Este proceso de focalización de la investigación en esta disciplina se explica por dos factores: 1. La enorme competencia entre las universidades norteamericanas de investigación a la hora de querer destacar en algún área. Si

se quiere una investigación de excelencia, se ha de ser excelente y como no se puede ser el mejor en todas las áreas hay que seleccionar alguna, en este caso, se ha elegido durante esta época a computer science. 2. El ser excelente es lo que permite atraer fondos para la investigación. Y el computer science es un área que durante las últimas décadas permitía obtener esos fondos. Máxime cuando la investigación en las universidades norteamericanas está financiada en su gran mayoría por los presupuestos del Estado. Y los fondos para investigación de esos presupuestos del Estado los controlan principalmente tres agencias el Department of Defense (DoD), la National Science Foundation (NSF) y los National Institutes of Health (NIH).

La investigación informática ha generado, y genera, una sustancial parte de los fondos de los que vive CMU. El mecanismo específico que existe en la "research university" norteamericana para transferir fondos de la investigación a los gastos generales de la institución se llama : el "overhead" o gastos indirectos. Este es un porcentaje que, en esta universidad es del 58%, se queda la institución para pagar sus gastos generales y que carga a los ingresos por investigación generados por sus profesores o personal investigador. Así si un equipo consigue un proyecto cuyos gastos directos (salarios, viajes,...) valen 1 millón de dólares, el costo final sera de 1 millón que se queda el equipo investigador más 580.000 dólares que se queda la universidad y que van a parar a parte de su presupuesto o E&GO ("Educational and General Operating Budget"). Para entender la importancia de la investigación informática en CMU es conveniente analizar brevemente el presupuesto de una institución que, como la universidad privada ,vive principalmente de la investigación y de las matrículas de sus estudiantes.

El 25 de mayo de 1991 el nuevo President Mehrabian explicaba la situación financiera de la institución : CMU ingresará en 1991 una cifra global de 291 millones de dólares. De este ingreso, el presupuesto que la universidad realmente controla (el E&GO) será algo menos de la mitad: 140,7 millones. Del resto hasta los 291 millones la mayor parte son ingresos por investigación, 99 millones al año, ingresos que van a pagar gastos directos de la misma y que la universidad no controla. Lo que si hace la universidad es cargar a esta cifra un plus del 58% (27 millones) que van a parar a su E&GO, con lo que éste queda equilibrado. En resumen, los ingresos por investigación que llegarán a CMU en 1991 serán de 126 millones (99+27), mientras que los ingresos por matrículas de

estudiantes sólo alcanzaran 91,5 millones.

La investigación informática cubre una gran parte de esos 126 millones de ingresos por "research", el 40% aproximadamente.

La subida de la investigación en la década de los 80s fue del 300%, pasando de 23,5 millones de dólares en 1978 a 96,3 en 1987. Por el contrario, los estudiantes de grado aumentaron sólo el 10% y los graduados el 77%.

En 1989, el presupuesto total de investigación de CMU fue de 120 millones de dólares. De éstos, la mitad, 60 millones, fueron generados tan sólo por dos centros de CMU : la School of Computer Science (39,249,000) y el Software Engineering Institute (20,900,000). (CMU Facts 1990:64). El tercer gran centro generador de fondos es la escuela de ingeniería, el Carnegie Institute of Technology que obtuvo 22,800,000 dólares ese mismo año. Su departamento más grande es el Electrical and Computing Engineering.

Las previsiones hasta el año 2000 indican que , mientras los ingresos por estudiantes permanecieran al mismo nivel, sera la investigación esponsorizada el área que seguirá subiendo entre un 20% y un 190%, y con ella el personal de investigación y las categorías técnico-administrativas, mientras que la plantilla de profesorado permanecerá estancada.

Hasta la II Guerra Mundial, Carnegie Tech. era principalmente una institucion para la educación de ingenieros. Pero en los años 50s, esta dirección cambió, transformando a esta institución de un instituto de enseñanza en una universidad de investigación.

El Presidente John C. Warner (1950-1965), fue un presidente que decididamente apostó por convertir a Carnegie Tech en una institución de investigación. A éste le siguieron otros dos presidentes que profundizaron esta orientación: H. Guyford Stever (1965-1972), que fusionó Carnegie Tech con el Mellon Institute para formar Carnegie Mellon University, y Richard Cyert (1972-1990), bajo cuya presidencia, y con la colaboración del Provost Angel Jordan, esta universidad se convirtió en uno de los principales centros de investigación informáticos de Estados Unidos y del mundo.

No obstante, esta orientación hacia la investigación hay que entenderla según el patron cultural de una institución norteamericana marcadamente tecnológica.

La investigación, y dentro de ella, la informática como área principal, ha tenido un carácter inicialmente aplicado, como veremos en el siguiente capítulo al analizar los objetivos del Computer Science Department. Es una investigación

financiada en mayor medida por fondos de la Defensa, lo cual no significa que tenga aplicaciones militares directas, pero sí que va orientada a objetivos ("goal oriented"), y ésto le imprime un carácter distinto a la investigación científica clásica como la financiada por la National Science Foundation, o los National Institutes of Health, más en la línea de la investigación básica al modo tradicional.

Los profesores que trabajan en Computer Science en CMU provengan de la matemática, física, sicología, lingüística, sociología o ética, todos ellos acaban haciendo programas de software, hasta el punto que el College of Humanities and Social Sciences, en particular su Psychology Department y su Philosophy Department, es un centro importante de producción de software en el campus.

La "problem solving mentality" durante los años 60s y 70s fue en gran medida revitalizada por el Computer Science y personas como Allen Newell y Herb Simon desde el College of Humanities and Social Sciences. Finalmente, la tarea de convertir el diseño en un "core curriculum" recayó en el Center for Design of Educational Computing, dirigido por Preston Covey, profesor de ética aplicada. Por otra parte, Carnegie Mellon no ha sido ajena a la tendencia antes descrita de "absorción en la cultura general universitaria", lo que significa absorción en la "liberal-arts culture". El periodo de la "research university" ha significado la entrada de CMU en el ranking de las universidades de élite norteamericanas, grupo liderado por el selecto grupo de las "Ivy League": Harvard, Yale, Princeton, Cornell, Dartmouth, Pennsylvania, Brown, donde se forma tradicionalmente la clase dirigente de Estados Unidos. En 1988 las tres primeras competidoras de CMU para captar estudiantes eran Cornell, el MIT y la University of Pennsylvania. (D. Scott. Search for Presid. 1989:19). CMU forma parte irreversiblemente del mundo universitario norteamericano.

Este mundo universitario a finales de los 60s y principios de los 70s se fue democratizando a consecuencia de las denominadas "students troubles". (Simon. Models, 1991:279).

Este fenómeno conllevó una creciente oleada de críticas sobre la tecnología y el papel de los tecnólogos. Como confiesa un ingeniero de CIT, en los años 70s ser ingeniero estaba "mal visto en la universidad norteamericana". Esta oleada coincidió con la publicación del informe de Domela Meadows del MIT "Los límites del crecimiento", documento básico del ecologismo y alimento de una corriente

internacional sensibilizada sobre temas como el deterioro del medio ambiente, la seguridad y la paz internacional, y el respeto por todas las culturas, lo que en Estados Unidos se engloba en el respeto a los derechos de las minorías o "minorities" (african-american, women, gays,...).

Lo paradójico es que en Carnegie Mellon estas tendencias, dada la influencia del proceso liberalizador del Carnegie Plan, encuentran apoyo en el campus tanto en departamentos de la escuela de ingeniería y como en los de Humanities and Social Sciences, su tradicional hogar.

Expresión del movimiento hacia la toma de conciencia de la responsabilidad social del ingeniero y los temas relacionados con la ecología, el pacifismo y el control políticos de los efectos de la tecnología es la formación, en 1970, del departamento de Engineering and Public Policy, dirigido en los últimos años por Granger Morgan, uno de los más influyentes en la escuela. Sus principales áreas de investigación en 1990 eran: "Energía y Sistemas Medioambientales", "Política de Telecomunicaciones", "Análisis de Riesgo", o estudio de los efectos nocivos de la tecnología en determinados trabajos, "Control de Armamentos y Política de Defensa", y "Manufacturas y Política Industrial".

Por su parte, el movimiento hacia una mayor formalización científica de la clásica actividad ingeniera de diseño tiene lugar en el Engineering Design Research Center, centro financiado por la National Science Foundation, uno de los principales centros para la investigación del diseño en U.S.A, se ha marcado una clara orientación destinada a convertir el diseño en una ciencia. Sus objetivos principales son: 1. "Desarrollar nuevas metodologías de diseño que constituyan el terreno de la ciencia del diseño". 2. " Incorporar nuevas metodologías de diseño en herramientas informáticas que faciliten el uso, el acceso, la transferencia y la modificación de la información de diseño a lo largo de todo el ciclo del desarrollo del producto". 3. "Desarrollar un medio ambiente que permita el trabajo conjunto de equipos de diseñadores dispersos por todo el mundo." (EDRC, 1991-92: 2).

En resumen, formalizar los principios y la metodología de diseño, sacándolo de una actividad artesana, y transformándolo en ciencia, al objeto de aumentar la productividad industrial.

A su vez, la GSIA, que empezó como una escuela de "industrial management", se ha ido transformando con el tiempo en una escuela con profesores de prestigio mundial en "accounting" (contabilidad) y economía como Yuji Ijiri. Entre 1962 y

1972 fue dirigida por Richard Cyert, y con una clara orientación hacia la ciencia económica neoclásica y los métodos matemáticos. Aunque ahora está en un proceso de profundo cambio.

4.2. EL DEPARTAMENTO DE COMPUTER SCIENCE DE CARNEGIE TECH: "THE ARPA ARTIFICIAL INTELLIGENCE LABORATORY".

Esta tendencia de Carnegie Tech a integrar los patrones de la cultura tradicional universitaria, la "liberal-arts culture", fue a mediados de los 50s, contrarrestada por un nuevo fenómeno que reforzó valores propios de la educación tecnológica, la "professional education" : la irrupción del ordenador y de la "computer science", una nueva disciplina en la que se mezclaban matemática aplicada, ingeniería electrónica, y teorías de la psicología del "problem solving". Todo ello unido a una casi exclusiva conexión de esta investigación con el Departamento de Defensa, y su agencia de proyectos avanzados. (ARPA)

La irrupción del computer en Carnegie Tech fue visto como una innovación cultural. Art Evans, uno de los primeros participantes de este nuevo campo en Carnegie Tech, recuerda así estos años fundacionales: "This was all new. It was all new. We were inventing the things, discovering all the things which are now part of the culture. We were inventing the culture".(CMU-SCS.25 Aniv.:49).

El departamento de Computer Science en Carnegie Tech se formó con unas características particulares respecto a lo que era entonces la institución:

1. A diferencia de otros departamentos del instituto más centrados en la enseñanza, este nació con una tendencia claramente hacia la investigación. Sus fundadores provenían de dos líneas de investigación distintas: la línea del "complex information processing" de Newell y Simon, y la línea de "programming languages" de Perlis.

Una tercera línea de investigación provino de la escuela de ingeniería: el diseño de "computer systems" y la desarrollaron Gordon Bell y Newell. Otra línea de investigación sobre semiconductores, iniciada entre otros por Angel Jordan, y que tenía posibilidades directas de entroncarse con la naciente industria de los transistores, no fue considerada en el Departamento. Esta concentración en áreas de computer science desconectadas de la industria influirá en la futura ausencia de una industria informática en Pittsburgh, a diferencia de Stanford o MIT, y constituye uno de sus puntos débiles de Carnegie Mellon una vez finalizada la Guerra Fría.

En 1971, con J. Traub, se iniciará una cuarta área de investigación, la "Theory", que no tomará cuerpo hasta 1981 con Merrick Furst.

2. La investigación que se acordó poner en marcha como prioritaria fue en torno a una disciplina denominada "the science of information processing", con lo cual este departamento se tendería a adscribir al College of Science, y no al de Engineering.

3. La investigación se orientaba, inicialmente de forma exclusiva hacia el "software", desde un enfoque particular que denominaba "pragmatic approach", dándole preferencia sobre el enfoque denominado "formal".

4. La investigación en "information processing" se planteaba en Carnegie Tech unida por fuertes lazos interdisciplinarios tanto con las "behavioral and management sciences" como con las "natural sciences". El departamento se veía como un "open center" abierto al conjunto del campus, inventando la figura del "joint faculty", o profesor en dos departamentos simultáneamente.

5. La investigación se financiaba de forma exclusiva gracias a un contrato iniciado con la Advanced Research Projects Agency del Departamento de Defensa, que en 1965-66 ascendía a 992.000 dólares. Esta dependencia casi total de la investigación del Department of Computer Science de CMU respecto a intereses estratégicos de la defensa nacional, y la débil relación con la industria informática, será una característica propia de esta comunidad, y que la diferencia de otros departamentos como los del MIT o Stanford donde las relaciones son más diversificadas. Las razones pueden encontrarse en que en el momento de formación de ese departamento la industria local de Pittsburgh, el acero, estaba declinando, y hasta los años 80s no se ha empezado a generar una industria local informática, basada sobre todo en el software, aún incipiente.

En las proposals planteadas por los investigadores del Departamento a DARPA, se habla repetidamente de la "ARPA Artificial Intelligence Laboratory". Ello indica los tres rasgos fundamentales de la investigación de Computer Science de CMU:

1. El estar centrada en la Artificial Intelligence, como área que dirige al resto de la computer science.

2. El realizarse con los métodos experimentales propios de un laboratorio.

3. El estar financiada exclusivamente por DARPA, la única institución que ha sostenido a gran escala durante décadas la investigación básica en Inteligencia Artificial en Estados Unidos.

4.3. UN "PARTICULAR APPROACH" EN LA INVESTIGACION EN COMPUTER SCIENCE.

Desarrollemos estas características.

El documento histórico que sirvió de base para la constitución de esta línea de investigación en el Department of Computer Science es la "proposal" o propuesta presentada a ARPA en abril de 1964 denominada: "Proposal for a Center for the Study of Information Processing", y firmada por Allen Newell, Alan J. Perlis y Edward R. Schatz. (1964).

En una cultura prefigurativa que funciona hacia el futuro, como en el caso de la "computer culture" de CMU, tiene una gran importancia analizar los planes, proyectos o "proposals" donde se codifican los objetivos a cubrir, el estado de la cuestión y las estrategias a seguir para conseguirlos. Las culturas prefigurativas son "goal-oriented", y la antropología que las estudie conviene que parta de sus "proposals", de sus objetivos. Lo que en una cultura postfigurativa, por utilizar una terminología de Margeret Mead, son los mitos o historias que fundamentan su práctica social, o en una logo-cultura son sus principios básicos y universales extraídos del libro de la naturaleza, en una cultura tecnológica son sus fines, objetivos o proyectos estratégicos, que estructuran su práctica social en pos de su consecución. Ello explica que no sea la verdad preexistente su criterio de validación sino lo que llaman "results". En la citada "proposal", por un lado, se especifica en ella que "the fundamental aim of the center is understanding the nature of information processing... The attempt to produce a science in an area that not long ago was purely a technology... implies a belief that there is much to discover and deep understanding to be had". (1964: 1).

El hecho que esta área haya sido antes una tecnología que una ciencia, lo acaba de confirmar de nuevo un perplejo Allen Newell al afirmar en el 25 aniversario del departamento: " Computer Science has been such an event. Its emergence has been specially confusing because of the computer's origin as an engineering and manufactured device" (CMU-SCS, 25th Anniversary. :41).

Este paradójico fenómeno marca lo que denominó "inversión del conocimiento" en esta disciplina, según la cual su avance sólo sera posible si antes se diseña un nuevo artefacto, programa o sistema que se toma como objeto empírico a posteriori para su estudio científico. El diseño, bien sea ingenieril o lógico formal, va antes que su descubrimiento empírico.

Este fenómeno lo vuelve a advertir el citado documento: "In mathematics, for

instance, which also pervades all fields, we do talk of pure mathematics and of applied mathematics. But information processing is much more closely tied to its applications. The new ideas and insights that move it towards being a science arise mostly from applications rather than from any self-generative character of the pure science of information processing." (ib.:3-4).

Este fenómeno lleva a los autores a fundir la teoría científica del "information processing" con los programas informáticos que la validan, con lo que la diferenciación ciencia básica-aplicada queda borrada. La teoría científica está unificada con la tarea práctica a realizar. En resumen, la teoría es el programa: " The scientific knowledge is contained, not in a formal theory, but in the knowledge, however expressed, of the structure of a class of information processing tasks and the programs that can do them. The validation of this scientific knowledge lies in the fact that operating programs run and communicated to new programmers, who can then produce new programs of high effectiveness in the area without themselves recapitulating all the mistakes and grotesqueries of the past" (Ib.: 6-7).

El conocimiento que desarrolla esta "science of information processing" en Carnegie Tech se basa en la construcción de programas que funcionan ("run"). Es la misma mentalidad basada en el "it works!" que tenía Andrew Carnegie.

El enfoque o filosofía del Centro, que 25 años más tarde se convertirá en la School of Computer Science, se define así : "A center is better characterized by its selections of specific problems, methods and attitudes, than by its most general goals. Almost a decade of research in information processing at Carnegie Tech has produced an approach which still seems to its major participants to be extremely productive. The proposed center will be, by and large, a continuation of this approach. We are oriented towards software--towards programs written on digital computers. We are empirical, in that we believe in constructing programs that do things, and in learning about information processing from the difficulties of construction and from the behavior of the resulting programs. We are theoretical, although not so much by a dependency on formal models...as by trying to formulate the essential nature of information processing." (1964:4).

El fin es "to formulate the essential nature of information processing", el medio " constructing programs that do things". Esta secuencia es la reconstrucción lógica de este pasaje, pero su formulación textual es la inversa:

1. "Constructing programs that do things" y
2. Formulate the essential nature of

information processing". Ya hemos explicado antes la razón de este enfoque. La mentalidad pragmática no sólo es propia de la "computer science" en esta institución sino de su misma configuración cultural, que fue primero un instituto tecnológico y sólo después una universidad.

Los fines científicos (el conocimiento sobre la naturaleza del "information processing") son inseparables de los medios técnicos. Dado que el programa informático en este esquema no es un simple medio sino un fin en sí mismo, la única forma de materializar la teoría, siendo necesariamente previa a ésta. El "information processing" es el modo de comportarse de un programa de software y sólo se le puede analizar construyéndolo.

Este enfoque , "a much more pragmatic approach", lo proponen expresamente Perlis y Newell como alternativa al desarrollo de una "formal theory of information processing systems" , consistente en un esquema matemático básico para la representación de todo tipo de información.

Desde Logic Theorist hasta SOAR, la actividad fundamental de la computer science en CMU se medirá por los programas o sistemas informáticos que construya.

Incluso departamentos como Psicología, Filosofía, o Música han adoptado esta filosofía de construir sus conocimientos en forma de software. Este fue el intento tras Andrew: convertir al profesor y al estudiante de CMU en un constructor de software informático.

Este es un rasgo principal de la "computer culture" de CMU: la investigación tiende a ser principalmente construcción de "software" o sistema informático, y no sólo para el "computer scientist" sino para el conjunto de disciplinas del campus.

4.4. UN "OPEN CENTER".

La propuesta en cuestión termina proponiendo la creación de un centro " 'open' towards the entire campus" (ibid:25). Ya desde 1961 operó en Carnegie Tech un programa de doctorado interdisciplinar denominado Systems and Communication Sciences, incluyendo a varios departamentos o escuelas: el de Mathematics, Psychology, Electrical Engineering y la Graduate School of Industrial Administration . Este programa fue el precursor del programa de doctorado en Computer Science de 1965.

Según la citada proposal, "information processing pervades other fields...and research on information processing ultimately becomes an integral part of many

fields". (ibid.:25). Por eso el centro debía estar abierto a todo el campus pues su objeto de estudio "information, the systems that process and transform it, and the way it is used to control, integrate and coordinate other systems" (ibid:2) les podía interesar a todos.

Carnegie Tech no fue la única universidad en organizar, por ese tiempo, un departamento de Computer Science. En 1965 más de 15 universidades norteamericanas impartían doctorados en Computer Science. Pero no todas ellas iban a entender el Computer Science como "information processing" y el "information processing" como una disciplina que debía crecer y desarrollarse en todo el campus de forma interdisciplinaria.

La visión específica de Perlis y Newell afirma que : "Research on information processing must grow up and flourish throughout Carnegie Tech because the Center is here, not in spite of it. Nothing would be more fatal than to erect an organization whose natural tendency was to absorb all such research within its confines, ...in which those 'outside' the center have little access to the advances achieved 'inside'. Thus, in a sense, a very substantial segment of Carnegie Tech should become the laboratory for the study of information processing" (1964:26).

20 años después Carnegie Mellon se transformaría en una "computer-intensive university". Según su exalumna y actual profesora de la escuela , Mary Shaw, fue Alan Perlis el que inició esta política de abrir el computing a todo estudiante o profesor de Carnegie Tech: " Carnegie Tech was one the very first schools that made a commitment to making computation broadly available to everyone on campus...Alan saw far enough ahead to open access, and he helped create the campus policy that made computing as freely available as the technology of the time would allow".(CMU-SCS, 25th Anniversary,..:33).

En el momento de la propuesta (1964) había ya en Carnegie Tech 7 departamentos implicados en el computing: Mathematics, Psychology, Electrical Engineering, Graduate School of Industrial Administration, Physics,Civil Engineering y Chemistry. El mecanismo desarrollado para mantener la interdisciplinarietà fue el "joint faculty", o doble adscripción del profesor, según el cual éste pertenecía a su departamento de ciencias, humanidades o ingeniería y al mismo tiempo trabajaba conjuntamente en la investigación del Computer Science Department.

Esta solución se concibió expresamente como "an official way of blurring the

lines of separation between those inside and outside the center". (ibid:26)

Esta forma de organización, al debilitar las tradicionales barreras de departamento y disciplina, ha hecho menos dificultoso la formación de una red como Andrew, y constituye una ventaja comparativa para el trabajo interdisciplinar entre culturas académicas distintas como la "professional" y "la liberal- culture". En 1990, la práctica totalidad de los 25 departamentos de Carnegie Mellon participan en la utilización del computing, la School of Computer Science tiene 27 "joint faculty" de un total de 86 profesores, y se da el caso de profesores como Dana Scott que pertenecen a tres departamentos en tres escuelas distintas: Mathematics en el College of Science, Philosophy en el College of Humanities and Social Sciences y la SCS.

Esta política de colaboración entre el computer science y el resto de disciplinas es lo que denominó "collaboration across boundaries" e influirá en introducir la mentalidad programadora a disciplinas como la psicología o la lingüística inicialmente ajenas a ella.

4.5. EL TRABAJO DEL CENTRO: "MIXING SERVICE AND RESEARCH".

Para llevar adelante el trabajo, el centro introducía una modificación fundamental en el tipo de actividad a realizar en una universidad: la realización simultánea de investigación ("research") y servicios ("service"). Desde 1956 existía en Carnegie Tech un Computation Center que proveía de servicios técnicos a todo profesor y estudiante que necesitara utilizar el ordenador. La propuesta citada propondrá incorporar el citado centro en el nuevo.

En el futuro el Computer Science Department contará además de la clásica estructura universitaria de profesores y estudiantes, que se dedicaran a la investigación "research", una estructura operativa, el "full time programming staff", que será la encargada del "service", consistiendo éste en implementar los programas informáticos. En palabras de un ingeniero español, visitante en esta universidad, "estos programadores no los tenemos en la universidad española, y son muy útiles dado que son los que llevan el trabajo adelante" En resumen, la investigación a realizar en lo que será el Department of Computer Science será integra R&D, o investigación y desarrollo. Así lo justificaban Perlis y Newell: "The best part of the systems programming world today does not do much research by the definition implicitly given above; it does high class

engineering... We have labored this long to establish the legitimacy, in a proposal for a research center, of a large full time programming staff whose function is not research. The large size is dictated by the need to keep the total system stable in the face of continuous advancement in the state of the art. " (1964 :33).

En resumen, el trabajo en su conjunto seguirá esta secuencia: "The typical life history of much research done at the center will be as follows: A graduate student will get interested in some aspect of programming, say in monitor systems. Working with one of the professors as advisor, he develops a programming language in whose terms monitor routines may be expressed... Focussing on the essentials, he works with a "pseudo-machine" that he has defined for his own purposes and simulates it on the computer... At the termination of this research it has become abundantly clear that this monitor language should be used for the Carnegie Tech system. At that point there may be a full man-year of hard development work ahead, for it is unlikely that any such new idea could be applied for the first time without considerable effort. Yet this next year-man is just development work; it is not research... At that point the full time programming staff assumes control, and does the additional work necessary to incorporate it into the system... They make a substantial enough contribution to the programming art so that, in retrospect, research has been done: revealed as a sideline to the task of making the Carnegie Tech operating system as good as possible". (ibid:32-33).

Esta estructura combinada de investigación y desarrollo será la que produzca 25 años después programas como MACH o máquinas como iWARP o NECTAR, que constituyen motivo de orgullo para la escuela. No fue otra la estructura que produjo el primer programa de Inteligencia Artificial, LT, en 1955. Esta actividad significa una complejización de la estructura universitaria desbordando su carácter tradicional basada en la relación bilateral profesor-alumno, e introduciendo una tercera componente: el personal operativo, que es el que permite llevar la lógica del "problem solving" a su final: la construcción de un sistema informático que funciona.

En 1990, el personal operativo de programación igualará en la escuela al de profesores, y la mitad de ellos, no tendrán tareas directas de educación. Una categoría particular de profesor es el denominado "research professor", dedicado exclusivamente a la investigación, esto es, a los proyectos, sin obligación de

enseñar, aunque también puede hacerlo. Esta figura se excluía de la propuesta inicial pero la realidad ha sido más fuerte que las intenciones iniciales: " No permanent full time research personnel would exist at the center" (ibid:31). La razón es que el trabajo de investigación supera al de enseñanza en "computer science".

4.6. LA RELACION CON EL DEPARTAMENTO DE DEFENSA Y CON LA INDUSTRIA : LA DEFENSE ADVANCED RESEARCH PROJECT AGENCY.

La formación del Computer Science Department marca un giro en la relación de Carnegie Tech respecto a la relación de la universidad con su entorno social. Carnegie Tech, como hemos analizado, nació como un instituto de formación profesional para formar técnicos cualificados para la industria de Pittsburgh. Su actual Board of Trustees sigue estando formado en su mayoría por industriales de la ciudad o antiguos alumnos que ocupan puestos de dirección en industrias a lo largo del país. Carnegie Tech era el instituto tecnológico de Pittsburgh y su comunidad.

Sin embargo, su transformación en una "research university" ha significado durante estos últimos 25 años el pasar a depender de organismos públicos y agencias de la Defensa, como DARPA. Con ello Carnegie Mellon ha alcanzado un rápido crecimiento pero desligándose en gran parte de la base social que le dió nacimiento: la industria local.

Este cambio fue favorecido, entre otros factores, por la línea de investigación que se adoptó desde un principio esta propuesta precursora del Department of Computer Science. Esta línea optó exclusivamente por el "software", rechazando explícitamente el trabajar en hardware. "The center, by inclination, talent and past action, is a center of software research...The center as organized contains no real effort on hardware research...Our plans for the full time faculty do not include scientists working on computer hardware" (1964: 34-40-42).

Pero la industria del "computer" no se inició en los años 50s por el software sino por los semiconductores. Tuvo como centro regiones que rodean otros centros de investigación informática de Estados Unidos, lo como Stanford el M.I.T. La influencia de estas universidades no fue directa en la formación de sus áreas industriales, pero si que sirvieron de lugares de reclutamiento de los ingenieros para las industrias que entonces nacieron. De momento, Pittsburgh no ha generado un fenómeno de tal magnitud, aunque su industria de "high tech" es

florecente contando con 700 empresas y 70.000 empleados en este sector.

(Pittsburgh Facts, 1990-91:10).

La ingeniería ha sido la profesión que tradicionalmente ha estado ligada con la industria. Carnegie Tech tenía una escuela de ingeniería bien relacionada con la industria local. La elección de iniciar la investigación en "computer science" excluyendo de entrada al hardware y sus componentes, el área más conectada con la ingeniería, con el tiempo ha favorecido un divorcio en Carnegie Tech entre el ingeniero y el computer scientist que no ha beneficiado a ninguno de los dos. Y sobre todo ha creado un vacío social alrededor de Carnegie Mellon al hacer depender casi toda su investigación de un único patrón: el Estado, y dentro de él, de un exclusivo ministerio, el de Defensa. Esto crea en la nueva era post-Guerra Fría que ahora se inicia una situación muy problemática para la propia universidad.

En 1979, con la fundación del Robotics Institute, esponsorizado inicialmente por una industria local, la Westinghouse, la investigación en Computer Science iniciaba la senda de su diversificación al servicio de la industria. Pero la escasa extensión del uso de robots en la industria norteamericana, en comparación con la japonesa, hace que este instituto no acabe de ofrecer una solución al problema ya detectado: como conectar la investigación del Computer Science de Carnegie Mellon con el mundo industrial.

La relación estrecha de la investigación en Computer Science de CMU con intereses de la Defensa, por el contrario, ha tenido ciertas ventajas para un instituto tecnológico pequeño y alejado de los centros tradicionales de poder académico de Estados Unidos, la costa Este y California. Le ha permitido durante 25 años fijarse con mayor libertad objetivos en su investigación, no reducidos a los intereses de la industria local, sino de alcance estratégico al servicio de intereses nacionales.

Mientras que la industria norteamericana exige una investigación dirigida a objetivos inmediatos, medidos en pocos años, la investigación financiada por DARPA en programas como el de Inteligencia Artificial se mide por décadas. Programas como el Speech Recognition, iniciado en 1971, lleva 20 años funcionando. Ello permite abordar investigaciones imposibles de financiar por la empresa dado su resultado incierto.

Computer Science en CMU ha sido inseparable de DARPA durante este primer periodo de su existencia. Fue esta agencia, antes denominada ARPA, (Advanced Research

Project Agency) la que seleccionó a este centro junto con el M.I.T y Stanford como centro estratégico para hacer investigación en "Big Science" en Computer Science. Se estima, según J.M. Newcomer, veterano "computer scientist" de CMU, que esta es la que se financia con proyectos por encima de un millón de dólares.

DARPA es una agencia que tiene poder para financiar este tipo de proyectos. Es una agencia del Department of Defense, con sede en Arlington, Virginia. Tiene un presupuesto anual para 1992 de 1.200 millones de dólares (AAAS Report XVI, 1992: 69). Está formada por 160 empleados civiles.

Un resumen completo de su trabajo lo encontramos en el Government Research Directory, editado por Kay Gill y Susan E. Tufts (1992), sus objetivos son: "Pursue highly imaginative and innovative research ideas and concepts offering significant military utility; support and manage projects assigned by the Secretary of Defense and marshal advanced research through a demonstration of feasibility for military application. DARPA's role in basic research is to develop selected new ideas (usually high-risk, high-payoff) from conception to hardware prototype to transfer to Service development agencies. DARPA does not conduct in-house research...Programs are conducted through contracts with industrial, university, and nonprofit organizations and with selected Service research and laboratories. DARPA's programs focus on technology development and proof-of-concept demonstrations of revolutionary approaches for improved strategic, conventional, rapid-deployment, and sea-power forces; and on scientific investigation into advanced basic technologies of the future. It selects for support only those research and development initiatives that promise significantly U.S. national security interests, while lowering costs, through technological advances or to prepare for further technological progress. DARPA comprises technical program offices for Aerospace Technology, Defense Sciences, Directed Energy, Information Science and Technology, Naval Technology, Nuclear Monitoring Research, Prototype Projects, Strategic Technology and Tactical Technology". (1992: 184). Entre otras oficinas.

Este carácter de proyectos innovadores, estratégicos, demostrables en cuanto a su factibilidad ("feasibility"), que interesen a la Defensa y que tengan un alto riesgo a cambio de altos rendimientos (high payoffs), son los que han marcado el carácter de Computer Science en CMU.

La investigación en Computer Science no es la clásica línea de investigación de

un departamento universitario, que funciona tradicionalmente sin plazos. Es una investigación por contratos, cuyo cliente es el Departamento de Defensa. Estos contratos son bianuales y tienen plazos de finalización. Desde 1962, cuando se inició la colaboración con la entonces denominada ARPA (Advanced Research Project Agency) hasta 1991, cada dos años el departamento ha tenido que revalidar la utilidad y calidad de su investigación, actualizando su "proposal" sobre "information processing".

Por estratégica entiendo que esta investigación 1. va dirigida a desarrollar un tipo de conocimiento con su tecnología correspondiente que se considera de valor para los intereses nacionales de Estados Unidos. y 2. que va destinada a conseguir una ventaja comparativa respecto a sus competidores nacionales, antes la URSS y crecientemente Japón y Europa.

Esto ha formado un tipo de investigador en Computer Science que tiene una visión estratégica de la tecnología que está produciendo. Como afirma Raj Reddy, director del Instituto de Robótica y director del comité de planificación estratégica que asesora al Presidente y al Congreso sobre el futuro desarrollo y aplicación del Computer Science and Technology: " We must create a vision for the future, which is exciting and challenging. ...Whenever possible, we must identify and work on problems of relevance to the nation--bold national initiatives that capture the imagination of the public.

Let me illustrate what I mean by two examples from biology and physics: the Decoding of the Human Genome, and the Superconducting Super Collider Project. These grand challenges of science are expected to require several billion dollars of investment each. However, the expected benefits to these projects to the nation are also very high. AI is in a unique position to undertake and deliver on such nationally relevant initiatives." (1988: 17-18).

Uno de los resultados de este pensamiento por "estrategic planning" fue la creación en los años 80s de la red Andrew, colocando a esta pequeña universidad de una mediana ciudad de Pennsylvania, a la cabeza de la innovación entre las universidades en Norteamérica.

Pero esta visión, para algunos profesores, no se detiene en los marcos meramente nacionales. Estrategas como el profesor Raj Reddy, de origen indú, considera que las implicaciones sociales de la Inteligencia Artificial abarcaran al conjunto de la humanidad en las próximas décadas: "By the turn of the century, it appears possible that a low cost (e.g. costing less than \$1000) supercomputer could be

accesible to every man, woman and child in the world. Using such system, AI researchers should be able to create a personalized intelligent assistant which would use voice and vision for man-machine communication, tolerate error and ambiguity in human interaction with machines, provide education and entertainment on a personalized basis, provide expert advice on day-to-day problems, make vast amounts of knowledge available in active form, and make ordinary mortals perform superhuman tasks leading to new discoveries and inventions at an unheard of rate. Believe it or not, such a system would help the illiterate farmer in Ethiopia as much as the scientist in U.S.A. or Japan.... It is my belief that the most important wealth we can share with the disadvantaged is the wealth of knowledge. If we can provide a gift of knowledge to village communities that would make them expert at what they need to know to be self-sufficient, we would then witness a true revolution". (1988: 19).

4.7. EL "STRATEGIC PLANNING" DE CMU: HACIA LA "COMPUTER-INTENSIVE UNIVERSITY".

Esta mentalidad estratégica, que forma parte de la actual cultura académica de CMU, es una innovación introducida en los años 70s, y en la que jugó un papel destacado el entonces presidente de CMU, Richard Cyert.

Este concibió el strategic planning en CMU como una vía "to identify the areas in which the university should be doing research and offering educational curricula, to specify the goals of the institution, and to determine the kind of university that is desired by the community". (Cyert, 1988: 91).

A diferencia del antiguo concepto de planificación, mecánico y determinista utilizado para controlar el proceso productivo de un país, el strategic planning seguido en CMU se basaba en estas ideas esenciales:

1. "Seeking a distinctive niche". Cada departamento, colegio, y el conjunto de la universidad, trataría de distinguirse en alguna área de investigación diferente a las de otros departamentos y colegios de otras universidades.
2. "Analytical ability and comparative advantage". Se trataba de analizar las disciplinas y determinar, con la ayuda de expertos, cuál iba a ser su evolución futura. A ello había que unir un análisis de qué ventajas tenía cada departamento o colegio en comparación con otros departamentos o colegios con los que competía en la lucha por conseguir una mejor calidad de la investigación y enseñanza.

3. "Uniqueness is Essential". El objetivo del strategic planning era colocar a la universidad en una posición única, distinta al resto de universidades. Para ello cada departamento debía analizar sus fuerzas ("strengths") y sus debilidades ("weaknesses") y tratar de desarrollar sus departamentos para hacer el mejor uso de sus fuerzas. " The worst action that an organization can take is to imitate another". (Cyert 1988: 92-93).

Se trataba pues de diferenciar, de adquirir una superioridad mediante esta diferenciación, y la mejor manera era abrir un campo del saber nuevo, donde la institución fuera única.

La universidad nació en Europa como una institución con vocación de abarcar al conjunto de saberes. Era una institución enciclopédica. En América, la cultura de los institutos de tecnología como Carnegie Tech conllevan criterios distintos: especialización, profesionalismo, competitividad, innovación. El strategic planning ha sido una vía para llevar adelante esta nueva cultura académica.

Durante cerca de 20 años, desde 1972 a 1990, ésta ha sido la filosofía básica de la Administración Cyert. "I have followed that philosophy for the past sixteen years". (1988: 93).

Curiosamente los departamentos que Cyert pone como ejemplo de esta filosofía son el de Psychology y el de Mathematics. Su línea ha sido focalizar al primero en "cognitive psychology" y al segundo en "applied mathematics".

Es interesante ver la relación de estos departamentos con Computer Science, y como su planificación estratégica ha sido una consecuencia de la fuerza de la Computer Science en CMU, en esta pasada etapa.

Así ve Cyert la transformación del departamento de Psicología: " Perhaps an ideal example of the application of these strategic principles has occurred in Carnegie Mellon's Psychology Department. For many years it attempted to cover the complete range of specialities in psychology (for example, clinical, industrial, experimental, and physiological), and, while it was a good department, it was not distinguished in any respect (in 1969, it was ranked thirty-fourth in the country). Though the strategic planning process, it became clear that a major potential strength of the department lay in the quality of our Department of Computer Science and its work on problem solving. The Psychology Department thus began to develop a concentration in cognitive psychology. With a few judicious appointments of new faculty members, the department quickly attained the leading

position in the country in the cognitive area" (ib: 95-95)

Esto es, la razón de estos cambios no está en estos departamentos sino en el de Computer Science, que se nutre tanto de sicología cognitiva para alimentar la Inteligencia Artificial, como de Applied Mathematics para el area de Algoritmos y Optimization.

No obstante, esta filosofía del strategic planning no entronca con la cultura norteamericana tradicional mayoritaria. " In the United States planning has been viewed as unacceptable to our society. The majority of the population tends to view planning as a part of a socialist society. Planning is viewed as antithetical to the nature of American society...The concept of the invisible hand dominates thinking in the United States with respect to planning" (Cyert 1986: 1-2).

¿De donde viene pues esta cultura de "planning" en CMU ?. Tiene varios componentes.

El primero es la propia cultura tecnológica norteamericana. A diferencia de la filosofía de los economistas, la mentalidad ingeniera, fundacional de Carnegie Tech, funciona por problem solving, no confiando en la "mano invisible". Y como decía Dick Teare, antiguo decano de Engineering and Science de CMU, en una discusión con Akram Midani, antiguo decano de Fine Arts, los cinco pasos del problem solving son: "1. Define the problem. 2. Plan de Attack. 3. Execute the Plan. 4. Check. 5. Learn and Generalize". (Midani, Teare. Focus. January 9, 1976:). El plan es consustancial con el "problem solving" ingeniero.

Un segundo componente es la cultura del "management" o dirección de empresas, nacida también en este país. CMU formó, entre otros con la ayuda de R. Cyert, una de las mejores escuelas de management del país. Pese a la influencia de doctrinas neoclásicas, el mangement es una disciplina que proviene de la ingeniería, fundada por un ingeniero mecánico norteamericano, Frederick Taylor. Un tercer componente, es la cultura del "planning" introducida en Estados Unidos en los años 30s, con el New Deal de F.D. Roosevelt, y promulgada por científicos sociales reformistas y políticos progresistas, para paliar los efectos de la depresión. En CMU, Cyert y Simon pertenecen a esa generación formada en esta época. La defensa del "organizational man", del "model of man" de racionalidad limitada, que confía no tanto en mercado con sus propias leyes autónomas sino en su propia actividad de "decision-making" es una expresión de esa cultura. En 1981, en las Sciences of the Artificial, Simon acaba proponiendo, con reservas

no obstante, un "social planning" como nueva variante de las ciencias de lo artificial.

Esta cultura del planning fue reforzada por la experiencia de la II Guerra Mundial y los cuarenta años de Guerra Fría, y la participación en masa de la ciencia en Norteamérica en los planes estratégicos nacionales. La expresión de este Zeitgeist es la cibernética (kibernetes: el timonel, el que gobierna). A su vez dió lugar al desarrollo de una serie de técnicas como la "operational research" y los juegos de estrategia útiles para fines militares. Esta cultura pasó a la empresa en forma de "strategic planning" y finalmente a la universidad, en primer lugar a aquellas como CMU que tenían una tradición de "teoría de la organización" en sus escuelas de management.

Un cuarto componente, es la influencia del modelo de planificación estratégica japonesa. Expresamente Cyert afirma: " We are interested, in view of the Japanese experience, of determining whether some form of central planning can improve the functioning of our system without destroying the basic freedoms that are an integral part of our lives". (1986: 3). La experiencia del strategic planning de CMU era vista así por Cyert: " I propose that research universities...function as MITI does. A university with a particularly strong research area should form consortia with the firms in the industry. Each firm should agree to contribute research funds to the consortium for a given period of time." (:1986:27). Esta experiencia ya se había iniciado casi 10 años antes, en 1979, con el Robotics Institute de CMU, fundado por Raj Reddy y Angel Jordan entre otros.

Finalmente el strategic planning acabó en la red Andrew, base del computer-intensive campus, que proporcionó a CMU una relevancia nacional. Esta iniciativa la acabaremos estudiando como un proyecto de investigación de la propia "computer science community" de CMU, en la que por primera vez, líderes de la misma como Allen Newell, aparecerán como líderes del conjunto de la institución universitaria, a la cabeza del Task Force fo the Future of Computing, comité que formuló la política del computing en CMU.

PARTE TERCERA.

LA SCHOOL OF COMPUTER SCIENCE,

CENTRO DE CARNEGIE MELLON COMO "RESEARCH UNIVERSITY".

1. QUE SIGNIFICA "COMPUTER SCIENCE" EN LA SCHOOL OF COMPUTER SCIENCE.

1.1. "GOALS AND STRATEGIC PLANS"

Analizar Carnegie Mellon como "research university" conlleva estudiar lo que ha sido el foco principal de investigación en esta universidad durante los últimos 25 años, la "computer science community". Esta comunidad ha terminado construyendo en 1988 una School of Computer Science, una de las primeras escuelas del país que se ha independizado de las facultades de ciencia o ingeniería, teniendo una entidad como colegio autónomo.

El Department of Computer Science se formó en 1965 dentro del College of Engineering and Science como departamento autónomo. En 1971 se adscribió al College of Science cuando este se separó del de ingeniería y en 1985 el Department of Computer Science se separó del College of Science. Y tres años después se constituyó en colegio independiente. En un documento sobre los objetivos y planes estratégicos de Carnegie Mellon de 1988 podemos leer esta valoración que hace la propia "computer science community" de su status: " The independent status of the Department has stimulated new and increased interdisciplinary activities in computer science and a variety of disciplines such as business administration, fine arts, psychology, philosophy, statistics and others. The independence has liberated computer science at Carnegie Mellon from the tendency to focus on mathematics or engineering. It has helped to establish information processing as a dimension of science that plays a role in many disciplines". (CMU, Goals and Strategic Plans, Feb 1988: 29).

Tradicionalmente, en Estados Unidos se ha tendido a considerar "computer science" como una rama de las matemáticas. En 1988, en el National Research Council, la principal agencia operativa conjunta de la National Academy of Sciences y de la National Academy of Engineering, la Computer Science and

Technology Board estaba incluida en la Commission on Physical Sciences, Mathematics, and Resources. La separación de uno de los principales centros de investigación del mundo en computer science tanto del College of Sciences como del Carnegie Institute of Technology tiene implicaciones sobre la manera de organizar la universidad y las diferentes disciplinas en ella.

El objetivo que claramente se marcaba el Departamento en su conjunto, pocos meses antes de transformarse en escuela, está expresado claramente en este documento sobre los planes estratégicos de CMU: " The overall goal of the Computer Science Department is to maintain its worldwide leadership role in computer science research. The Department has been consistently ranked among the top three departments in the nation for more than a decade". (ibid.:28).

El 16 de septiembre de 1988, Angel Jordan, entonces Provost de CMU y Nico Habermann, el jefe del departamento, escribían un "position paper" denominado "The formation of the School of Computer Science" en el que afirmaban: "Although faculty members, department heads, and deans were divided on the issue of the separation of CS from MCS, deans and many department heads took the position that if CS were to be a unit separate from MCS, it should become a "school" rather than a "department". The authors of this position paper agreed with that point of view and strongly argued in favor of the formation of a school".

(Jordan, Haberman, 1988: 1)

Meses después se llegaba a un acuerdo para organizar una escuela donde el antiguo departamento, con su "federation of interrelated disciplines: programming systems, computer systems, artificial intelligence, and theoretical foundations" (Jordan , Habermann, 1988:5) , se unía a otros tres centros de investigación de CMU también de excelencia: el Robotics Institute, el Machine Translation y el Information Technology Center.

Su investigación se define como "experimental computer science research", lo que, se evidencia, según afirmó recientemente el nuevo Presidente Mehrabian con ocasión de la despedida del hasta ahora decano de la escuela, Nico Habermann, por "the quantity and quality of the experimental systems built in CMU".

(Mehrabian, 27 Sept. 1991).

Una de las características de la institución, y considerada por ella misma como vía estratégica para conseguir el objetivo antes señalado ha sido el énfasis en la participación de los estudiantes de doctorado en proyectos de investigación, así como la contratación de profesores de investigación (" research faculty")

dedicados plenamente a esta tarea. " The Department has a worldwide reputation in experimental computer science research. It has received this recognition largely because of its emphasis on active participation of Ph.D. students in research projects rather than on a heavy course load. This strategy has paid off not only for the graduate students, but also for the quality and quantity of experimental systems that have been produced at Carnegie Mellon. We plan to continue this successful strategy and give leading-edge research the highest priority" (CMU, 1988, Goals and Strategic Plans:30).

En resumen, esta escuela, con fecha 1991, está constituida fundamentalmente por un "Doctoral Program in Computer Science", un "Master Program in Software Engineering", realizado en colaboración con el Software Engineering Institute, y tres institutos de investigación citados. Los responsables de cada uno de ellos forman un comité, presidido por el decano de la escuela. No obstante, la institución está en periodo de cambio.

A su vez la escuela tiene otros programas de doctorado hechos en colaboración con otras escuelas o departamentos de Carnegie Mellon: "Algorithms, Combinatorics and Optimization", hecho con la Graduate School of Industrial Administration, y el programa "Pure and Applied Logic" hecho con el departamento de Philosophy.

Sus actividades con otros departamentos son mucho más numerosas y variadas, incluyendo doble pertenencia ("joint appointment") de 28 profesores de la misma, a otros tantos departamentos de la universidad, programas de investigación conjuntos y actividades diversas como seminarios, charlas, y otras de carácter informal.

En total, la escuela tiene, en el curso 1990-91, alrededor de 700 personas, el núcleo más numeroso de investigación de toda la universidad, núcleo que incluye profesores, estudiantes graduados (no tiene estudiantes de grado), staff y personal técnico(programadores, técnicos de servicios...).(Rashid, 1991:vii).

Sólo de profesores y staff cuenta con 564 personas en el curso 1990-91, distribuidas así: 284 Computer Science, 222 Robotics Institute, 29 Machine Translation, y 29 Information Technology Center.(Faculty and Staff Guide 1991: 175-177).

Su presupuesto de investigación en 1989 fue a su vez el mayor de la universidad :39 millones de dolares, casi doblando el de la segunda institución, el colegio de ingeniería (22 millones). (Carnegie Mellon Facts 1990:64).

1.2. ¿QUE SIGNIFICA "COMPUTER SCIENCE" EN CARNEGIE MELLON?.

Pero este estudio pretende centrarse no en la estructura organizativa de la institución sino en el significado que tiene el sistema de conocimiento denominado "computer science" para los investigadores de esta organización. Dado los escasos recursos disponibles y la complejidad de la organización he preferido empezar por estudiar de entre las clásicas áreas de toda etnografía, lo que M. Harris denomina el "patrón universal" (1983:130), "infraestructura", "estructura", y "superestructura", aquella que es más propia de la antropología: los significados compartidos de la comunidad en cuestión. Al ser ésta una comunidad tecnológica, la "superestructura" tiene una relación particular con la "infraestructura", dado que la tecnología se plantea como un conjunto de significados compartidos, que al mismo tiempo sirven para construir los objetos tecnológicos.

¿Qué significa "computer science" en esta comunidad?.

En el papel de N. Habermann y Angel Jordan antes citado se afirma: "Computer Science at CMU has always been considered an experimental science" (1988:5).

Esta es una escuela de investigación, que como ya vimos en capítulos anteriores, esta tarea significa investigación y desarrollo ("research and development").

Esta escuela produce conocimiento en forma de sistemas experimentales, como MACH o Nectar, que basan su prestigio mundial. Estos sistemas están agrupados en disciplinas (AI, programming, etc.) y el conjunto de las mismas forma lo que se denomina "computer science" y se considera que ésta es una "experimental science".

Esta definición viene a reafirmar la establecida por los "founding fathers" de Computer Science en CMU: Alan Perlis, Allen Newell y Herbert Simon. En septiembre de 1990, la comunidad de computer scientists de CMU celebraba el 25 aniversario de la formación del Department. En esta reunión se volvió a recordar dicha definición de Computer Science, que fue publicada en una carta enviada a la prestigiosa revista Science el 10 de julio de 1967.

Esta empezaba así: "Whenever there are phenomena, there can be a science to describe and explain those phenomena. Thus, the simplest (and correct) answer to 'What is botany?' is, 'Botany is the study of plants'. And zoology is the study of animals, astronomy the study of stars, and so on. Phenomena breed sciences. There are computers. Ergo, computer science is the study of computers". (

CMU-SCS 25th Anniversary.:43).

Más adelante se explicitaba esta definición frente a la objeción que pretendía reducir la computer science al estudio de los algoritmos o programas: " In the definition , 'computers' means 'living computers' --i.e. the hardware, their programs or algorithms, and all that goes with them. Computer Science is the study of phenomena surrounding computers. 'Computer plus algorithms', 'living computers' or simply 'computers' all come to the same thing-- the same phenomena". (25th Anniv: 43).

La definición de Computer Science como "the study of phenomena surrounding computers", para Mary Shaw, veterana profesora de la escuela, "it is still valid"(Entrevista personal. 24 Mayo1991). La podríamos considerar como el conocimiento compartido básico de la comunidad de computer scientists de CMU. Sin embargo caben distintas interpretaciones de esta definición como veremos a continuación.

1.3. EL "CYBERNETIC DIALOGUE" DE ALAN PERLIS.

Alan Perlis ocupa un lugar protagonista en la computer science en CMU, tanto como investigador como educador. Mientras Simon y Newell vinieron de culturas distintas a la antigua Carnegie Tech, Alan Perlis , nativo de Pittsburgh, recibió su Bachelor in Science en Chemistry en CIT en 1942, y el doctorado en Matemáticas en el MIT , siendo el primer director del Department of Computer Science de Carnegie Tech. Para su antigua alumna Mary Shaw, Perlis "helped create the campus policy that made computing as freely available as the technology of the time would allow" (25 Aniv. :33). Y para Habermann, "Alan Perlis truly represents the best tradition in teaching". (ibid: 36).

En su libro Introduction to Computer Science(1972) Perlis explica los elementos que según él componen el nuevo campo introduciendo lo que denomina el " 'cybernetic dialogue' (between us and machines)". Este diálogo consistía en cuatro componentes: "Us, algorithms, languages, machines". (Introd. CS. 1972: 2).

"Introduction to Computer Science" fue un libro para educar a estudiantes en computer science. Está escrito por un educador, un "teacher". Y desde esta óptica Alan Perlis afirma: "Education reveals that the computer's role is shaped by people and not by the machine itself...Thus, it is my conviction that an introduction to computer science must be significantly more than mere drill in

the use of a popular programming language. It must provide students with a pousto (A term used by Archimedes. It is a platform on which to stand in order to pry up the world-if only one had a lever that were long enough". (1972: xii).

A diferencia de la visión más propia un matemático como Donald Knuth para el que computer science es el "study of algorithms" (1974:323), para Perlis lo que era nuevo en esta tecnología no eran los algoritmos sino su unión con la máquina:

"What will be new is the attachment of algorithms and languages to a machine-the electronical digital computer-which is the agent commanded to perform (to do , to accomplish) algorithms expressed in languages". (1972:2).

Para Perlis no sólo están atados los algoritmos a la máquina. Todo el diálogo está encabezado por el propio computer scientist, "us"(nosotros). Como hemos visto para Perlis, la función del ordenador es construida por las personas, no por la propia máquina. De aquí, la figura del "us". Lo curioso es que no dice en general "the man" sino "us", situándose a sí mismo también en el inicio de este diálogo.

Esto resulta nuevo. La ciencia tradicional en Occidente se constituyó bajo la premisa griega del orden natural. Lo primero para un científico es el descubrimiento de un orden objetivo, exterior al propio científico. Robert Merton afirma por ello que el "universalismo" es el primer valor del "scientific ethos". La ciencia moderna empieza por la astronomía copernicana y la física de Galileo. No con la sicología introspectiva. La computer science a su vez empezó gracias a la labor de físicos (Mauchly), ingenieros electrónicos (Eckert), y matemáticos (Turing, Von Neumann). Sin embargo, en la cultura de Carnegie Mellon, como antes hemos descrito, ha habido una combinación específica entre "professional" y "liberal education" que permite entender esta anomalía: que el círculo cibernético lo empiece el "us", sin el cual no se entiende de dónde procede el ordenador y todo el sistema pierde sentido.

En el diálogo cibernético de Perlis, los computer scientists juegan el papel iniciador del círculo, papel activador del que depende la construcción de algoritmos, los lenguajes y las máquinas. Y este papel consiste en un diálogo.

Comparemos esta visión de un hombre de Carnegie Tech con la de, por ejemplo, Marvin Minsky del M.I.T., uno de los iniciadores de la Inteligencia Artificial.

En su libro "Computation: Finite and Infinite Machines" (1967), uno de los primeros libros sobre teoría de computación, escribe en su introducción:"Man has within a single generation found himself sharing the world with a strange new

species: the computers and computer-like machines...all of us are falling under the shadow of their ever-growing sphere of influence, and thus we all need to understand their capabilities and their limitations...We still lack the tools necessary to fully analyze, synthesize or even think about them". (1967:vii).

Esta visión es la de un científico que descubre una realidad que necesita comprender. Esta es la visión que concuerda con la de la ciencia clásica europea, el descubrimiento de un orden objetivo a comprender ("understanding") mediante la razón y los sentidos.

El ser humano ("man") se ha encontrado ("found") compartiendo el mundo con una especie extraña ("strange new species"): los ordenadores. En el texto no se pretende explicar como han llegado, tan sólo se constata que "todos nosotros estamos cayendo bajo la sombra de su creciente esfera de influencia". La imagen aquí descrita es la de un poder ajeno al ser humano ante cuya influencia "caemos" ("falling"), hasta dejarnos ensombrecidos ("shadow").

El interés del científico es, según Minsky, primero, analizarlo, "analyze", a continuación, sintetizarlo, "synthesize", y por último, pensar, "think", acerca de esta extraña especie.

1.4. HERBERT SIMON : "COMPUTER SCIENCE AS EMPIRICAL SCIENCE". Herbert Simon ha sido el principal defensor de la definición de Computer Science realizada en 1967. Y es la que aún se evidencia en la expresión de "experimental science".

Para este profesor de CMU y premio Nobel, y sus colaboradores: "the story of scientific progress reaches its periodic climaxes at the moments of discovery" (1989. Langley, P, Simon, H. Bradshaw, G, Zytkow, J.: 3). Este programa de investigación no se diferencia mucho del iniciado por Francis Bacon en el siglo XVI. Así, "Bacon", se denomina un sistema experto recientemente construido por este equipo. "The scientific enterprise is dedicated to the extension of knowledge about the external world. It is usually conceived as being made up of four main kinds of interrelated activities: gathering data, finding parsimonious descriptions of the data, formulating explanatory theories, and testing the theories" (ib: 18).

En el libro *The Science of the Artificial* (1981), Herb Simon, figura clave en la cultura de CMU, da su propia versión de la "computer science": "Since there are now many such devices in the world, and since the properties that describe them

also appear to be shared by the human central nervous system, nothing prevents us from developing a natural history of them. We can study them as we would rabbits or chipmunks and discover how they behave under different patterns of environmental stimulation. Insofar as their behavior reflects largely the broad functional characteristics we have described, and is independent of details of their hardware, we can build a general-but empirical- theory of them". (ib.:24).

En este sentido, Computer Science sería para Herb Simon una ciencia más en el sentido clásico occidental del término, como puede ser la física o la zoología.

Esta sería su visión de esta disciplina.

Y culmina afirmando: "To understand them, the systems had to be constructed, and their behavior observed". (: 1981:25).

Si comparamos esta visión de la computer science con las de Perlis y Minsky apreciaremos sustanciales diferencias. Para Simon, al igual que para Minsky, los ordenadores están ya ahí, son una realidad. Esta constatación de un hecho es la propia de la ciencia empírica. En efecto, la ciencia moderna, desacralizó el mundo, lo convirtió en un objeto de estudio, en un "fact". Y por consiguiente, el conocimiento que aporta parte de un "análisis" de este hecho. En esto, se diferencia de la visión de Perlis que parte del "us" y según la cual las máquinas no son hechos objetivos, sino productos del "computer scientist", del "us".

Simon admite que para comprender los ordenadores desde un punto de vista empírico, primero "the systems had to be constructed". Si no hay ordenadores primero, no puede haber ciencia empírica de los mismos. Siguiendo su lógica, si no hay "phenomena", no puede haber ciencia empírica de los mismos. Por tanto la primera actividad del computer scientist ha de ser diseñar esos ordenadores :

"The research that was done to design computer time-sharing systems is a good example of the study of computer behavior as an empirical phenomenon....The main route open to the development and improvement of time-sharing systems was to build them and see how they behaved". (1981:25).

Pero, para Herb Simon, el diseño es una actividad instrumental en la computer science, que es una ciencia empírica. Preguntado, para qué hacía programas de ordenador, contestó comparándolos con el cultivo que Mendel hizo de los guisantes para entender las leyes de la herencia genética. (Entrevista, 30 Nov.).

Tradicionalmente así ha sido visto el diseño por la ciencia, como la actividad

de los ingenieros, subordinada a un fin más elevado :el "understanding of the nature". Esto ha podido ser válido para la física, la química e incluso la biología. Pero es un esquema que al trasplantarlo a la computer science produce un resultado inesperado.

En computer science, entendida como ciencia empírica, se da la paradoja que la labor de diseño ha de preceder necesariamente a la de la ciencia empírica para que ésta se pueda desarrollar. Es su condición sine qua non. Ello produce lo que he denominado una inversión del conocimiento. La tecnología precede, se anticipa sistemáticamente a la ciencia y dirige su evolución. El científico si quiere seguir dirigiendo ha de hacerse un científico diseñador, y acaba reconociendo al diseño como centro del "artificial world" del que forma parte el ordenador.

La definición de Computer Science en 1967 estaba inspirada en las ciencias naturales, como la biología o la zoología, pero, dos años más tarde, en 1969, Herb Simon admitía en unas charlas dadas en el M.I.T., un instituto de ingenieros, que existía una diferencia fundamental entre "natural worlds" y "artificial worlds", y que habían unas "sciences of the design" que se diferenciaban de la "natural science", pues trataban ,no con cómo las cosas son, sino con "how they might be" y concluía: " As soon as we introduce 'synthesis' as well as 'artifice' we enter the realm of engineering" (1981:7).

No obstante, esta ambigüedad entre "experimental science" y el papel del diseño en su construcción no se ha puesto en evidencia, al menos en forma de un debate público. Computer Science de CMU se le sigue definiendo como una "experimental science" al modo de las ciencias naturales y sin embargo, como veremos, se le hace avanzar mediante una labor incesante de diseño, actividad propia de la ingeniería. El propio Newell en el 25 Aniversario de la institución simposium reconocía que la computer science había tenido un inicio "specially confusing because of the computer's origin as an engineered and manufactured device". ("Revisiting 'What is Computer Science'"1990:41).

1.5. "COMPUTER SCIENCE AND THE UNIVERSE OF DISCIPLINES". 1985.

Durante 25 años, la definición de Perlis, Simon y Newell ha servido para poder unir dos denominaciones con las que se suele nombrar el mismo campo: "computer science" e "information processing" .

Con motivo de los debates para la formación de la SCS, Allen Newell, junto con Nico Habermann, John McDermott y Raj Reddy trataron de avanzar en la

clarificación del lugar de la computer science respecto al resto de disciplinas. Su papel, no publicado, " Computer Science and the Universe of Disciplines". (1985 Habermann, N. McDermott, J. Newell, A. Reddy, R.), es un intento de esclarecer el lugar de la computer science en relación con todas las otras materias.

La idea central del mismo es que la computer science ha de desarrollarse y mantener contacto con todas, absolutamente todas , las otras disciplinas. Se le compara con las matemáticas, en tanto que éstas sirven a su vez a una gran parte de las ciencias. Pero la computer science va más allá. Se pone el ejemplo de la Inteligencia Artificial y su intento de explicar la poesía. No se trata, finaliza el papel, de establecer una nueva disciplina-reina sino hacer la información y la computación propiedad común de todas las disciplinas.

Para estos autores, en los años 80s, el universo de disciplinas relacionadas con el procesamiento de información era ya increíble: inteligencia artificial, ciencia cognitiva, comunicaciones, ingeniería informática y de las telecomunicaciones, cibernética, educación, epistemología, lógica , lingüística, matemáticas, sicología, semiótica, estadística, ...Y entre ellas, la computer science parecía jugar el papel central.

Desde el punto de vista de estos autores, las razones son complejas pero destaca una: la computer science juega el papel central en tanto que es una disciplina que se centra en la programación ("programming").

Recordemos que la computer science de Carnegie Mellon ha priorizado desde los años 60s el trabajo en software y programación. Fue la cuna de la Inteligencia Artificial y acoge al primer centro federal para el estudio del software, el Software Engineering Institute.

Comúnmente, se considera la información como la materia sustantiva de la "sociedad postindustrial", y su procesamiento, mediante el computador, como el adjetivo. Pero el centro de la "computer science" no es la información, materia de estudio que comparte con otras muchas disciplinas, como las ciencias sociales o el periodismo. Su epicentro parece ser, en CMU al menos, la programación, la información procesada según un programa de ordenador. Algoritmos, lenguajes lógicos, hardware, e incluso inteligencia humana, son instrumentos de una función que da sentido a este conjunto de herramientas: el computer program.

Como afirmó Simon: "En el campo de las computadoras la hora de la verdad es un programa que funciona; todo lo demás es profecía." (1977:8).

En este sentido, ni siquiera el conocimiento ("knowledge") sería lo sustancial. Ni tampoco la inteligencia, ("intelligence"). La clave sería el conocimiento como programación. Y la programación como computer program. El "mythos" y el "logos" son función, en esta cultura, del "programming", actividad intelectual que da sentido a la utilización del resto de conocimientos como simples medios para construir programas.

Ello implica que la jerarquía intelectual se establece en torno al programa informático. El computer scientist con más influencia será el que produzca el más potente y complejo programa informático. Para ello se puede utilizar la psicología o la lógica, o los algoritmos ingenieros. Pero la "cognition", o el "learning", o la "logic" o los "algorithms" son, en esta cultura funciones del programa, de la máquina. No objetos de estudio en sí mismos. Así fue con el Logic Theorist Machine de Newell, Shaw y Simon en 1956, así fue con el General Problem Solver de Newell y Simon en 1958, con Dendral de Feigenbaum desde 1965 a 1983, y ha sido con SOAR de Newell, PRODIGY de Jaime Carbonell y THEO de Tom Mitchell en los 80s.

"Programming" en computer science significa computer programming :programas ejecutables por un ordenador. No todo programa humano es un computer programming. El antropólogo Roy d'Andrade estableció en 1981 una diferenciación interesante entre "computer programs" y "cultural programs". (1981), basada en entre otros elementos en la "especificidad" de los mismos. Mientras los primeros han de estar rigurosamente especificados siendo exactos y no-ambiguos, los segundos tienen una falta de especificación.

El problema de esta división es que parece no considerar los programas informáticos como programas culturales. Y lo son. Son obras humanas, y para su construcción el computer scientist ha debido dejar a un lado las ambigüedades y centrarse en determinar un "effective procedure", un "well-structure problem". La diferencia es pues entre programas culturales, no entre programas culturales y programas informáticos.

Con todo sí hay una distinción a rescatar en la matización de D'Andrade. Una cosa es el programa informático, el programa de la máquina, y otra y programa de investigación del computer scientist que da lugar al programa informático. Su distinción permite abrir una antropología de la "computer science culture", basada en el análisis de los programas de los "computer scientists", aquellos que dan lugar, que construyen las máquinas, en tanto que artefactos humanos. La

tarea de analizar las máquinas y sus programas es propia del experto en computer science, pero la de analizar a los constructores de dichas máquinas es tarea de los antropólogos que estudian tecnoculturas.

Así, como veremos, la propia Artificial Intelligence es una ciencia que tiene un programa cultural que va más allá de los programas informáticos que construye: el programa de la unidad de la inteligencia del ser humano y la del computador. Un programa de origen religioso y científico al mismo tiempo, que da sentido y significado a los concretos programas de ordenador que construye.

Han habido intentos en CMU de generalizar el "programming" al conjunto del procesamiento de información humano. Así Nico Habermann, decano de la escuela hasta 1991, puede afirmar: " The symbolic processing of information is something that has interest in it self, but it affect other disciplines in a similar way that mathematics affects physics and engineering. Information processing is not restricted to just physics and engineering. It applies to everything.

Computer science shouldn't be buried in an engineering school, for example, because it is not just engineering , mathematics, data processing or artificial intelligence. Computer science really has something to say about the way that information is processed in all disciplines."(Carnegie Mellon Magazine 1990:56)..

Esta equiparación ("similar way") de la computer science con la matemática y la física no deja de ser sorprendente. La cultura científica moderna, en efecto, nace de la unión de la matemática y la física en el Renacimiento, renovando la tradición de la cultura científica griega. La física, la ciencia del "orden natural", es concebida como la disciplina que fundamenta la visión del mundo como Cosmos. (El surgimiento de una física del Caos, a partir del principio de incertidumbre de Heisenberg, es una expresión de la crisis de la antigua cosmología, pero conserva la misma problemática).

El que se empiece a plantear que existe una nueva disciplina, el "information processing", dentro de una universidad, que tiene una función similar a la que la matemática juega en la moderna ciencia, es un hecho con implicaciones culturales significativas.

En su discurso de despedida, el 27 de septiembre de 1991, Nico Habermann volvió a repetir estas ideas: " Creo firmemente que computer science es un nuevo tipo de disciplina, distinta a las matemáticas y a la ingeniería, un tipo de disciplina que posibilitará a las restantes materias de la universidad el

dedicarse al diseño de nuevas realidades simuladas con ayuda del ordenador".
(1991).

Lo primero que un análisis cultural ha de hacer es estudiar el programa cultural de la computer science, qué fines últimos persigue, qué métodos utiliza, que personas lo llevan a cabo, y que realizaciones concretas (esto es, que programas informáticos) produce. La computer science no es un programa de un ordenador sino un sistema de conocimientos que un grupo de seres humanos han puesto en marcha para hacer programas de ordenadores. Y como tal no puede prescindir de los deseos ("desires") y creencias ("I believe that"), de las ilusiones, de los conflictos de poder, los pactos y convenciones que acompañan la cultura humana, sin los cuales no habrían nacido los ordenadores.

En Carnegie Mellon, se llegó a un pacto entre dos versiones de la disciplina, la de Newell y Simon defensores del "information processing" y la de Perlis, profesor de "computer science". Este pacto es el que ha permitido que durante 25 años personas provenientes de la matemática, las ciencias sociales, el management y la ingeniería, se consideraran trabajando en el mismo campo. La red Andrew, y la propia SCS, se pensó con la visión de poder abarcar al conjunto del campus como un laboratorio del "information processing".

Si esta versión del computer science se mantiene, si la disciplina se sigue definiendo como "el estudio de los fenómenos que rodean al ordenador", dado que éste es un fenómeno en expansión (que abarca a 55 millones de ordenadores personales sólo en Estados Unidos), esta propia definición puede acabar conduciendo al estudio de las "computer cultures" en su complejidad, esto es, el conjunto formado por los seres humanos, sus programas informáticos y sus máquinas y redes electrónicas que las conectan, en un país y a escala cada vez más internacional.

1.6. CARNEGIE MELLON Y LA "COMPUTER SCIENCE COMMUNITY" DE NORTEAMERICA.

La definición de los "founding fathers" del Computer Science de CMU fue un acuerdo entre dos visiones de la disciplina que han estado en conflicto en Estados Unidos desde los años 60s.

Esta disciplina se denomina en este país "computer science" por características culturales propias. En 1965, la Association for Computing Machinery celebró una histórica votación a cerca de su cambio de nombre. La alternativa era pasar a

denominarse "Association for Computing and Information Sciences". Ganó la opción de conservar el nombre por 3794 votos contra 2203. Destacados directores de departamento como George Forsythe eran partidarios de la segunda opción. Herb Simon y Allen Newell han estado denominando su investigación "complex information processing" y no computer science, que consideran un término "parochial" (1972:888). Pero la nueva institución se denominó School of Computer Science.

En efecto, "computer science" es un término propio de la cultura informática norteamericana. Y tiene un carácter ambiguo.

En Europa, la disciplina se denomina "informatique" (Francia), "informatik" (Alemania), "datalogi" (Dinamarca). Para Donal Knuth, prestigioso computer scientist de Stanford, "These other names for computer science all de-emphasize the role of computing machines themselves, apparently in order to make the field more 'legitimate' and respectable" (1974: 324). Sin embargo, también habría que decir que algunos de ellos como "informatique" o "informática" no necesitan del calificativo de científicos como en Estados Unidos. En Estados Unidos los doctorados son en Computer Science, término que funde ciencia y máquina en una sola expresión. El propio equipo interdisciplinar inventor del ENIAC, el físico Mauchly y el ingeniero Eckert son un símbolo de la disciplina.

La definición de computer science como "the study of the phenomena surrounding computes" es propia de CMU. Pero no es la única definición ni la mayoritaria entre los computer scientists en Estados Unidos. El debate sobre la naturaleza del computer science dura ya 42 años en Estados Unidos y no se ha terminado. Ward Goodenough definió en los 60s la cultura como "shared knowledge", un sistema de conocimiento compartido. Desde hace siglos, el sistema de conocimiento dominante de la cultura occidental es la ciencia y la tecnología, formando lo que podríamos denominar una logo-cultura. La computer science ha nacido en Estados Unidos hace apenas unas décadas y constituye la primera ciencia nacida en Norteamérica.

Tras la II Guerra Mundial, Estados Unidos recogió de Europa la tarea de liderar los valores de la moderna civilización occidental. Este sistema de valores tiene como conocimiento superior la ciencia. Hasta ahora el sistema de ciencias que rige en Estados Unidos, como en Europa, está presidido por las ciencias naturales y representado en su escalón superior por la National Academy of Sciences, bajo la dirección de los físicos, y otros científicos.

La computer science ha nacido en América en los años 60s, de la confluencia de la ciencia, tanto natural como social y de la ingeniería, y sus objetivos, métodos y estructura de actuación son una innovación en ese sistema de conocimiento cultural. Es quizá el principal cambio en el ámbito académico desde el establecimiento de el moderno sistema de las ciencias, establecido por Auguste Comte a principios del siglo XIX.

En el reciente informe de la Association for Computing Machinery, se parte de esta incertidumbre sobre la naturaleza de la disciplina: " It is ACM's 42nd year and an old debate continues. Is computer science a science?. An engineering discipline? Or merely a technology, an inventor and purveyor of computing commodities?. What is the intellectual substance of the discipline?." (P. Denning.1989: 9).

Esta situación de incertidumbre no llega comunmente al gran público, ni siquiera al resto de disciplinas. Se considera a la computer una tecnología en sí misma determinante de los cambios en el conjunto social. Un análisis más afinado de la propia comunidad de computer scientists nos permite apreciar las enormes contradicciones que implican las innovaciones que la computer science conlleva. En 1978, en una reunión entre directores de departamentos de Computer Science celebrada en Snowbird, Utah, bajo el título, "Quo Vadimus". , y presidida por Joseph Traub (1981), el entonces director del Department of Computer Science de CMU, se trató de abordar un análisis entre otros temas sobre la naturaleza de la computer science y sus paradigmas. Tres posiciones destacaron. 1. Las que afirmaban que computer science era una ciencia nueva pero de la misma naturaleza que la física o la química. 2. Las que sostenían que era una "nueva especie de ciencia" con paradigmas distintos a las anteriores y 3. por último los que colocaron a la computer science como una disciplina al servicio de la sociedad al mismo nivel que la ingeniería o la medicina.

En el primer caso, paradójicamente no estaba un académico universitario sino J.S. Birbaum, miembro del Computer Science Department de IBM T.J. Watson Research Center, sostenía que: " For me, computer science is an emerging science; I view it as an experimental science whose nature is almost interdisciplinary...Computer Science does not differ in these regards from other more established sciences at comparable stages of gestation, such as physics, chemistry, and applied mathematics, and in my opinion is that its paradigm should be based on the history of those sciences". (1981:352).

En la segunda corriente, Juris Hartmanis, matemático y director del Computer Science Department de Cornell University, argüía que : "I believe very sincerely that computer science is a brand new species among all the known sciences and that is fundamentally differs from the older sciences...I am deeply convinced that computer science does not obey these and other paradigms from physical sciences. To repeat my main point, ...in computer science we are much more interested in exploring what can be done, what kind of computing structures can be built, what kind of languages are easily specified and understood, what data structures are economical, etc...At the risk of oversimplifying, we can say that in physical sciences we are primarily interested in the existing and in computer science (or the new species of the sciences of which there surely will be more) we are primarily concerned with that which is possible, with what can exist." (1981:354).

La última posición estuvo representada por Raj Reddy, profesor de Inteligencia Artificial de Carnegie Mellon. En lugar de preguntarse según la clásica forma socrática sobre qué son las cosas, o qué es computer science, se preguntó por su papel en la sociedad: " The question I would like to adress myself to is, "What is the role of computer science and how can it say relevant to the needs of the society in the 80s and the 90s?". Note that I am not asking "What is computer science? but rather, "what is the role of computer science?". Once we agree on the role of computer science we can derive as a consequence the central issues that will drive the field in the future. I would like to propose an expansive view of computer science and say that "computer science is a science devoted to the enhancement of the mental abilities of the human being". I suggest that computer science has the same type of role to play as engineering or medicine". (ibid.:354-355).

Estructura básica de este enfoque: 1.¿ Para qué computer science?.¿ Qué objetivos sociales puede cubrir?. 2. En función de ello, ¿qué es computer science?.

Esta pregunta por la utilidad, por el "para qué" es característica de una mente tecnológica, más que científica. Y es la propia de CMU como universidad nacida de un instituto tecnológico. Y en cuya cultura hay una inversión de las relaciones profesional-liberal education, en la que la segunda está en función de la primera.

Pero ésto es sólo un aspecto. El otro es, que esta definición se pregunta por la

utilidad social de esta tecnología, por su lugar al servicio del desarrollo del ser humano. Esta pregunta no es ya propia de un tecnólogo, sino de un educador. Un clásico ingeniero pone la ciencia al servicio del desarrollo tecnológico. Un científico pone la ingeniería al servicio de su labor descubridora. Un político pone la tecnología y la ciencia al servicio de la sociedad. Pero sólo un educador pone la tecnología al servicio del reforzamiento de las habilidades mentales del ser humano.

" How can one enhance the mental abilities of a mathematician or a biologist? Will computer science take over all other fields? What does it mean to enhance the mental abilities? And so on. I believe our job is not to replace a biologist of a civil engineer but rather to develop tools and techniques that will make it possible for them to do their job better, faster, and with greater productivity...The primary mission of computer science is to develop techniques and systems which make "effective use of knowledge in day-to-day problem solving". (ibid.:355).

1.7. CARNEGIE MELLON Y EL INFORME DENNING: LA VUELTA " TO TRADITIONAL SCIENTIFIC STANDARDS".

Este debate en los últimos años ha entrado en una etapa en la que parece que existe un consenso en torno a la mayor parte de los directores de departamentos de Computer Science del país en torno a la necesidad de conformar la enseñanza de la computer science según los patrones tradicionales de la ciencia y la ingeniería.

Estos patrones rigen un sistema de conocimiento, codificado según la Nomenclatura Internacional Normalizada para los campos de la Ciencia y la Tecnología, propuesta por la UNESCO en 1974 (Comisión Interministerial .1990" :27-68).

Esta clasificación está encabezada por los "campos" de la Lógica (código11), y la Matemática (código12), y seguida por las ciencias naturales (Astronomía y Astrofísica, Física, Química, Biología...), las Ciencias Tecnológicas . A continuación vienen las Ciencias Sociales. Y finalmente los últimos escalones los ocupa la Etica (código71) y la Filosofía (72). Su orden no es arbitrario o casual sino expresión de un consenso internacional en la comunidad científica y tecnológica a cerca de la jerarquía de saberes en la presente cultura.

En esta clasificación la "ciencia de los ordenadores" (1203) es una disciplina

dentro del campo de las matemáticas, y la "tecnología de los ordenadores" (3304) una disciplina dentro del campo de las "Ciencias Tecnológicas" (33).

En 1989, un informe de la "Task Force on the Core of Computer Science", dirigida por Peter J. Denning expresaba el claro propósito de conformar la computer science "to traditional scientific standards". La definición que hacía de la disciplina era : "The discipline of computing is the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation and application. The fundamental question underlying all of computing is . "What can be (efficiently) automated." (P. Denning, Communication ACM January 1989 : 12).

Esta pregunta ya fue formulada por George E. Forsythe, matemático, y fundador en 1965 del departamento de Computer Science de Stanford. Pero este consideraba en 1963 la "computer science" como " the birth of a coherent body of technique" y concebía su departamento incluyendo "experts in Programming, Numerical Analysis, Automata Theory, Data Processing, Business Games, Adaptive Systems, Information Theory, Information Retrieval, Recursive Function Theory, Computer Linguistics, etc..." (1963: 166-178).

La nueva definición del Informe Denning sobre Computer Science estrecha la disciplina y está en conflicto con la de CMU.

Como señaló A. Newell en el 25 aniversario citado al referirse a otras definiciones acerca de qué es computer science: " The preferred candidate usually has a strong flavor of computer science being the study of algorithms. However important algorithms are--and they are indeed important--much of the richness of computer science would disappear if we really took the central task to be discovering algorithms and understanding their nature". ("Revisiting..."1990: 41).

El informe Denning es significativo dentro de la situación de la "computer science culture" en Estados Unidos. Es el resultado de un trabajo de varios años de la Association for Computing Machinery , organización que se define como "scientific and technical society" y que abarca la mayoría de la comunidad de computer scientists del país, así como a estudiantes, ingenieros, y técnicos de las empresas "computer professionals" que trabajan en esta área. Su análisis permite apreciar mejor el significado diferencial de "computer science" en CMU. El modelo tradicional que propone el informe Denning para encuadrar el computer science de ahora en adelante esta organizado en tres patrones culturales o

"cultural styles" que corresponden a la matemática, la ciencia natural y la ingeniería: " The three major paradigms, or cultural styles, by which we approach our work provide a context for our definition of the discipline of computing. The first paradigm, theory, is rooted in mathematics...The second paradigm , abstraction, (modeling) is rooted in the experimental scientific method...The third paradigm, design, is rooted in engineering... Despite their inseparability, the three paradigms are distinct from one another because they represent separate areas of competence. Theory is concerned with the ability to describe and prove relationships among objects. Abstraction is concerned with the ability to use those relationships to make predictions that can be compared with the world. Design is concerned with the ability to implement specific instances of those relationships and use them to perform useful actions. Applied mathematicians, computational scientists and design engineers generally do not have interchangeable skills" (Denning, 1989:10-11).

Lo sorprendente de esta definición es que intenta meter en una sólo disciplina "computer science" los paradigmas que hoy están repartidos entre tres tipos de campos distintos: matemática, ciencias naturales e ingeniería.

Al comentar esta nueva orientación que apunta en la comunidad de computer scientists norteamericana con la veterana profesora Mary Shaw de la SCS afirmó textualmente que no estaba de acuerdo con esta triple división, ni con las conclusiones del informe.

Si esta definición de "computer science" de la ACM se impusiera en Carnegie Mellon, se daría la paradoja de que quedarían fuera de la escuela dos de sus fundadores, Allen Newell y Herb Simon, que han dedicado su vida a demostrar que la computer science no se reducía a la matemática sino que abarcaba todo tipo de manipulación de símbolos. Científicos a su vez que confiesan que su principal objetivo no es hacer más inteligentes los ordenadores sino "understanding the nature of human mind". (Newell. "Desires and Diversions". Dec,4.1991).

Pero guste o no en CMU, esta tendencia a la normalización de Computer Science es la tendencia que se está imponiendo en todo el país. La matemática sirve de teoría a la ciencia natural que proporciona los modelos para los diseños ingenieros. ¿Ha sido este el tipo de investigación seguida en CMU?. ¿Qué ha distinguido la investigación de computer science en CMU para que no acabe de cuadrar en los "traditional scientific standards" del informe Denning?.

Carnegie Mellon aparece como una comunidad de computer scientists particular en

Norteamérica con características propias:

1. Como señala Merrick Furst, Vicedecano de la escuela, un hecho excepcional en CMU es que no ha tenido área de Theory, hasta los años 80s, a diferencia del resto de departamentos de similar categoría. (Entrevista. 16 Abril).

Todavía hoy, profesores como Allen Newell sostienen que un programa informático (SOAR) es a su vez la teoría en "information processing" ..

2. La Artificial Intelligence (AI) ha tenido, en proporción a otras áreas como hardware o programming systems, un mayor peso en CMU que en otros departamentos del país.

Y a su vez, la AI de CMU ha estado muy influenciada por un científico social, Herb Simon, educado en Ciencia Política, más que por la matemática o la ingeniería.

El punto de partida de los padres fundadores de la disciplina en CMU ha sido más un "model of man", el hombre de racionalidad limitada, que un modelo de máquina.

Y es a imitación del "model of man", y su forma de pensar (heurística) como se han construido los programas informáticos.

La pregunta, en el programa de investigación de Newell y Simon, no es tanto qué proceso humano puede ser automatizado sino qué software puede ser construido según el modelo de pensamiento de un human problem solver.

En consecuencia, se ha tratado sistemáticamente en CMU de abrir el ordenador a los científicos sociales, incluso a los escasamente formalizados como los psicólogos, y a todo tipo de disciplinas, en un esfuerzo de integración, de "putting it all together", de hacer del "problem solving" un lenguaje común a la ciencias y la ingeniería. Andrew fue una red pensada principalmente no para los computer scientists sino para el resto del campus, con un especial énfasis en el College of Humanities and Social Sciences.

Sin embargo, para el Denning Repport solo hay cabida para tres categorías de investigadores en este campo: matemáticos aplicados, "computational scientists" al modo de los científicos naturales, e ingenieros diseñadores.

3. Por último, la computer science de CMU ha tenido unos lazos con la Defensa proporcionalmente mayores que los otros departamentos en Estados Unidos, en parte determinada por su menor relación con la industria. Ello ha permitido a este centro un desarrollo de una visión estratégica basada en planes a largo plazo, muy relacionados con el interés nacional. .

Esta trayectoria ha terminado favoreciendo un ejercicio de strategic planning de la propia universidad, siendo una de las primeras universidades en programarse a

sí misma con esta técnica.

Esta dependencia de los fondos de DARPA, y no de la Academia (National Science Foundation) le ha permitido una gran libertad de determinar su estructura educativa y su organización interna, más similar a un "think tank" tipo RAND Co, que a una institución universitaria.

En estos momentos, los doctorados que otorga CMU en computer science los avala el prestigio investigador de la institución, no una orden oficial.

La tendencia que apunta el informe Denning parece perseguir un sentido contrario: fijar los límites de la profesión, delimitando quién está dentro y quién fuera. Para ello funciona una comisión de acreditaciones (the Computing Science Accreditation Board. CSAB), como ya se hizo con la carrera de ingeniería, encargada de normalizar los curriculums en todos los departamentos.

Describamos más en profundidad el programa de investigación en Computer Science de CMU y observemos sus innovaciones, entroncadas en una tradición cultural propia de instituto de tecnología con una tradición humanística y en ciencias sociales.

1.8. "THE DOCTORAL PROGRAM IN COMPUTER SCIENCE" DE LA SCS Y LOS PROGRAMAS DE INVESTIGACION DE LA ESCUELA.

La investigación básica de la escuela se realiza principalmente por profesores del área denominada "THE DOCTORAL PROGRAM IN COMPUTER SCIENCE" en la School of Computer Science. El grupo de profesores que forma este programa constituye la columna vertebral de la School. Este grupo y sus proyectos pertenecen a lo que antes era el Computer Science Department de CMU, fundado en 1965.

La investigación y la estructura educativa de este programa se divide en cuatro "áreas" : "ARTIFICIAL INTELLIGENCE", "PROGRAMMING SYSTEMS", "COMPUTER SYSTEMS" Y "THEORY".

Dentro de estas áreas, el profesor en cuestión trabaja en varios "interests", (p.e. redes de ordenador, problem solving, o supercomputers). Y estructura con ellos sus "projects".(p.e. Prodigy, Mach,...). Estos proyectos forman la espina dorsal de los grandes programas de investigación que toman la forma de contratos, esponsorizados en su mayor parte por DARPA (Defense Advanced Research Projects Agency). En 1990 se agrupan en cuatro grandes proyectos: "INFORMATION PROCESSING", "STRATEGIC", "SPEECH" y "ERGO".

Estos programas forman parte de un inicial contrato único que desde los años 60s el Computer Science Department de CMU ha mantenido con DARPA, denominado "INFORMATION PROCESSING RESEARCH", y que ha sido la columna vertebral de la investigación en computer science en CMU.

Este contrato ha marcado un hecho diferencial de esta escuela, y de los otros tres enclaves estratégicos seleccionados por DARPA para realizar la investigación básica en Artificial Intelligence en Estados Unidos durante los pasados 30 años: Stanford, Massachusetts Institute of Technology , el Stanford Research Institute y CMU. Hasta el punto que en sus proposals se afirma: " It is often referred to ...as one of the four ARPA AI Laboratories (though the work under this contract is much broader than AI,) (CMU-CS-Proposal,1976-78:8).

Ello ha colocado a los seleccionados centros en el centro de la investigación en computer science en Estados Unidos. La diferencia con otros departamentos medianos que también tienen áreas de investigación similares, es que ellas no tienen estos grandes programas de investigación estratégicos, que son los que permiten la producción de los sistemas como Mach o Nectar de los que se enorgullece con motivo, la escuela, y hacen aparecer a ésta como un centro de primer orden en computer science en Estados Unidos y a nivel mundial.

La escuela cimienta su prestigio y se basa en la investigación en torno a los programas de investigación y desarrollo con interés nacional, y de larga duración que tiene. La SCS no es, principalmente, un centro de enseñanza para la producción de doctores en computer science, sino un centro de investigación básica en computer science, y en especial en Artificial Intelligence. Y entre ambas funciones existe un conflicto real, dado que obedecen a dinámicas culturales distintas.

Es en el sector de investigación donde se han producido históricamente las principales innovaciones de la School. Lo cual es una razón para escogerlos como base de análisis. Estos proyectos están organizados por los profesores más cualificados de la School.

El objeto de estudio quedará pues delimitado por estas cuatro áreas de investigación y dentro de ellas por aquellos proyectos dirigidos por los profesores que combinen dos factores: mayor cualificación investigativa reconocida en la School por sus colegas, y también se tiene en cuenta la dedicación a las tareas educativas, medida por el número de alumnos que supervisan. A esta figura se le reconoce por "advisor".

He seleccionado distintos proyectos que reúnen algunas de las características que considero valores culturales propios de la School, y que forman lo que podríamos considerar su patrón cultural:

- 1.) La SCS está estructurada principalmente en torno a una investigación básica en computer science, y en especial en Artificial Intelligence, investigación financiada principalmente por DARPA. Durante la mayor parte de su corta existencia su carácter fue el ser un "ARPA Artificial Intelligence Laboratory".
- 2) La SCS rige mayoritariamente su investigación por un "pragmatic approach" orientado a "construcción de sistemas" ("building systems"), rasgo propio de esta School. Este enfoque fue adoptado en 1964 en la propuesta de fundación del Center for Information Processing, firmada por Allen Newell y Alan Perlis, origen de la actual School. (1963: 13). En particular, la actividad característica de la School es la construcción de "programs" o software informático, bien de Inteligencia Artificial o de Programming Systems, las dos áreas fundacionales del Department en 1965.
- 3.) El enfoque para acometer la investigación es "sistémico" u holista: "total system effort", abierto al trabajo interdisciplinar y de innovación entre distintas áreas del computer science y entre esta disciplina y otros departamentos de CMU considerados como laboratorios para el estudio del "information processing" ("open center"). Esta orientación está ya en el origen de proyectos interdisciplinarios como el Logic Theorist, y a experimentos como la red Andrew en los años 80s.

Dentro de este enfoque, y el papel clave de la AI en esta escuela, es característico de la computer science de CMU una relación preferente con determinados departamentos del College of Humanities and Social Sciences: en particular con Psychology (Herbert Simon) y Philosophy (Dana Scott).

- 4.) Este "total system effort" conlleva un estilo de trabajo que promueve la formación de equipos ("teamwork"), y una estructura informal en la escuela con menor compartimentación y controles burocráticos que en instituciones de similar tamaño, estructura que favorece el trabajo en forma de redes ("network"). Internet, y dentro de ellas, subredes por especialidad, es la red electrónica de mayor uso en la SCS,

El resultado de todo ello son sistemas como MACH o NECTAR. Nico Habermann, con motivo de la celebración del 25 aniversario de la School celebrado en Septiembre de 1990 los valoraba así: " Another example of Carnegie Mellon's contributions

to computer science is the Mach system, which is very successful and useful in the design of architectures and the development of large programming systems....That is indeed a success story. You might say a similar thing is true of iWarp and the Nectar project..." (Carnegie Mellon Magazine, Fall 1990). Ambos sistemas son resultado de la Strategic Computing Initiative adoptada por DARPA en Octubre de 1982.

Otros proyectos como SOAR, PRODIGY Y THEO son programas de Inteligencia Artificial basados en capacidad de aprendizaje y están en proceso de maduración. Uno de ellos, Theo de Tom Mitchell, ha encontrado ya una aplicación en forma de robot autónomo. Ambler es el primer robot autónomo construido por el Robotics Institute de esta escuela en 1990 para la exploración del planeta Marte, con apoyo de la NASA, dentro del programa "Space Exploration Initiative" de la Administración Bush.

Por último, hay proyectos de lenguaje de programación como ERGO y TYPES IN PROGRAMMING, dirigidos por Dana Scott, que pretenden utilizar la lógica en beneficio de una mejor y más fiable construcción de programas informáticos. Los profesores o "faculty" constituyen el núcleo principal de investigación. En total suman 86 profesores, 35 en Inteligencia Artificial (40%), 22 en Programming Systems (22%), 16 en Computer Systems (18%) y 12 en Theory (14%), aunque gran parte de estos últimos trabajan en el área de Programming también. Su distribución por área de interés fue , en el curso 1990-91 la siguiente según el Faculty Research Guide de la escuela:

ARTIFICIAL INTELLIGENCE.

SCIENTIFIC DISCOVERY. H. Simon.

INTELLIGENT ARCHITECT.....SOAR...Newell, Lehman.

PRODIGY....Carbonell.

THEO...Mitchell, Mason, Schlimmer, Simmons.

SPEECH....Reddy, Rudnicky, Stern, Ward, Young.

VISION....Kanade, Ikeuchi, McKeown, Shafer, Webb, Witkin.

MACHINE TRANSLATION...Nirenburg, Tomita, Waibel.

REPRESENT.KNOWLEDGE....Berliner, Touretzky.

MAN-MACHINE INTERFACES.....Bates, Giuse, Perlin.

AUTOMATED THEOREM PROVING...Andrews

COMPUTATIONAL LINGUISTICS...Evans.

NEURAL NETWORKS...Fahlman.

PROGRAMMING SYSTEMS.

SOFTWARE (SEI)...Barbacci, Shaw, Garlan, Gibbs.

PROGRAM. ENVIRON...Habermann, Wing, Clarke, Rollins, Cooper.

PROGRAM. LANGUAGES... Lee, Pfenning, Reynolds, Scott, Brookes.

HUMAN-COMPUTER INTERACTION...Myers.

DISTRIBUTED SYSTEMS... Howard, Tokuda, Satya, Morris.

OPERATING SYSTEMS...Rashid, Forin.

COMPUTER MUSIC...Dannenber.

PROGRAMMING EDUCATION...P.L. Miller.

NETWORKS.....Bruegge, Sansom, Steenkiste, Gross.

SECURITY...Tygar.

COMPUTER SYSTEMS.

PARALLEL COMP...Bianchini, Bisiani, Blelloch, Fisher, Subhlok, Rutenberg.

VLSI SYSTEM VALIDATION....Bryant,

VLSI CAD...Director.

NETWORKS MULTICOMPUTERS...Kung, Gross, Melziolgu, O'Hallaron,

DATA STORAGE SYSTEMS...Gibson.

MULTIPROCESSORS...Siewiorek, Segall.

THEORY.

ALGORITHMS....Bianchini, Furst, Kannan, Miller, Sleator.

COMPLEXITY....Rudich.

LOGIC....Andrew, Scott, Reynolds, Leivant, Harper, Lee, Pfenning, Clarke,
Brookes, Statman.

Estos 86 profesores cuentan para la investigación con 227 estudiantes de grado seleccionados en función de su capacidad investigadora. La selección en la School es rigurosa. En 1987 se recibieron 900 solicitudes tanto del país como de todo el mundo, siendo admitidos tan sólo unos 30 estudiantes, de los cuales un tercio aproximadamente eran estudiantes extranjeros. Como dato de referencia, en el curso 1991-92 se han admitido dentro del Programa de Doctorado de Computer Science 67 estudiantes, y de ellos, 15 provienen de otros países en su mayoría asiáticos: 5 de universidades chinas, 5 indúes, 4 europeas (3 inglesas y 1 griega) y 1 israelí.

Como afirma el Graduates Studies in Computer Science, la admisión en la School se rige por el criterio de capacidad investigativa: "The potential for

outstanding ability in research is always the primary criterion".

A estos estudiantes hay que sumar alrededor de 200 colaboradores entre programadores (68 personas), profesores visitantes, secretarias, managers y técnicos de servicio.

Por último, la escuela se ha dotado de una infraestructura técnica de primera calidad. Según se afirma en su informe Goals and Strategic Plans 1988: " The Department is probably matched by no other academic computer science department in the richness of its computing environment. There are more than 300 scientific workstations in use, 45 DEC-VAX machines, several multiprocessor machines of the Encore and Sequent type, and a variety of special-purpose machines such as Symbolics, Warp, Hitech, etc." (:28-29). Por último, la Escuela está conectada por la red Internet a toda la comunidad nacional e internacional de computer scientists, y mediante la red Andrew al resto del campus.

1.9. AREAS DE CONOCIMIENTO EN CONTINUO CAMBIO.

La relación profesor-estudiante se establece sobre un terreno común: el de la investigación, está en función de la investigación. El profesor tiene unos intereses de investigación que ha conseguido concretar en un programa de investigación, financiados prioritariamente por Defensa. El estudiante es libre de seleccionar entre los profesores de la escuela aquél cuya investigación más se ajuste a sus preferencias.

Un elemento clave a la hora de estudiar una cultura como ésta en proceso de construcción es el tiempo. Los conocimientos son caducos y cambiantes como la propia tecnología. Los datos que aparecen en este estudio son válidos en la fecha 1990-91. Muchos de ellos quedarán obsoletos probablemente en los próximos dos tres años, sino lo son ya. El Departamento de Computer Science empezó hace 25 años con dos docenas de profesores y estudiantes. Hoy ya no existe. Lo que existe es una School of Computer Science, con 700 personas, dentro de la cual, lo que antes era el Department hoy es el Doctoral Program of Computer Science. Esta escuela tiene hoy tres institutos de investigación, pero otros se están formando ya en su interior. Se acaba de iniciar un programa sobre High Definition Television. Por otra parte, casi todo el personal del Information Technology Center ha sido trasladado al Instituto de Robotics, a fin de profundizar la investigación en multimedia.

¿Qué significa ello en materia de áreas de investigación y de disciplinas?

Uno de los mayores problemas para entender el sistema de conocimiento de la SCS es que este sistema está en construcción y estos propios profesores están edificando en estos momentos una clasificación o taxonomía clara de los diferentes componentes de dicho sistema.

Dentro de ella los 86 profesores e investigadores que la forman tienen distintos intereses o "interests".

Bajo el título de "interests", el Faculty Research Guide del Programa de Doctorado en Computer Science publicado en 1989-90, enumera para el conjunto de los 86 profesores de las cuatro áreas, hasta 246 intereses distintos. Los profesores con mayor número de "interests" son Dana Scott (27), A. Newell (17), J. Carbonell (16), H.T. Kung (12), M. Shaw (11).

Por citar un ejemplo dentro del área de AI, los "interests" de Allen Newell en esa fecha eran 17 según esta guía: "Chunking", "Cognitive Architecture", "Cognitive Modelling", "Cognitive Science", "Computer Architecture", "Human Computer Interaction", "Human Factors", "Intelligent Architectures", "Intelligent Tutoring Systems", "Interfaces Human-Computer", "Machine Learning", "Parallelism", "Practice", "Problem-Solving", "Production Systems", "Program Synthesis", "Psychology".

En este conjunto hay componentes heterogéneos: hay técnicas como "chunking"; subáreas de Inteligencia Artificial como "Machine Learning"; hay también otras áreas de computer science como "Computer Architecture" o "Parallelism"; hay ciencias tradicionales como "Psychology", y por último, también hay nuevas ciencias como "Cognitive Science".

Dos años después los "interests" de este profesor han aumentado a 22, según el Faculty Guide de 1991-92. Se conservan los fundamentales pero se añaden unos y se dejan otros. Los nuevos intereses son: "Concept Formation", "GOMS", "Intelligent Control of Software", "Knowledge Acquisition", "Knowledge-Based Systems", "Parallel Processing", "Protocol Analysis". Se abandonan los siguientes: "Practice" y "Program Synthesis".

Estos cambios se relacionan con el programa SOAR, el software inteligente que está construyendo A. Newell desde principios de la década de los 80s, en colaboración con investigadores de ésta y otras universidades.

Esta diversidad de áreas de investigación los profesores de la escuela han tratado de agruparla, a efectos educativos, en conjuntos más reducidos. Pero, como veremos a continuación, el área educativa también cambia de un año para

otro.

Así en el Syllabus o programa de estudios de 1990-91 en el área de Inteligencia Artificial, cuyos responsables eran Alan Bates y Tom Mitchell , se delimitaban a efectos de los exámenes o "qualifiers" de los estudiantes sólo cuatro subáreas : "Search", "Logic and Knowledge Representation", "Machine Learning" y "Architectures".

En el curso actual, 91-92, esta división ya se ha cambiado. Está en discusión un cambio de curriculum. Al parecer el estudiante ahora puede elegir entre 2 o 3 áreas entre una docena de campos dentro de la Artificial Inteligence de esta Escuela:

- a. " Search and Problem Solving Techniques.
- b. "Representation of Knowledge".
- c. "AI Languages, Software Tools, and System Organizations".
- d. "Expert Systems and Knowledge Engineering".
- e. "Learning, Knowledge Acquisition, and Analogical Reasoning".
- f. " Parallel AI Arquitectures and Connectionism.
- g. "Natural Language Processing".
- h. "Speech Understanding".
- i. "Vision".
- j. "Robotics, Manipulation and Locomotion".
- k. "Theoretical Foundations of AI (logical and philosophical foundations).
- l. "External Areas of Study Directly Related to AI- for example:
 - "Linguistics".
 - "Human Cognition".
 - "Automated Programming"
 - " Automated Mathematics".

Estos datos son una muestra de cambios de las áreas de conocimiento en el sector de la AI, en relación con su clasificación y construcción taxonómica. Estos cambios han ido, hasta el momento, en la dirección de una mayor complejización de áreas y opciones a elegir.

Pasemos a analizar con más detenimiento el programa de investigación en AI de CMU.

2. LA ARTIFICIAL INTELLIGENCE DE CMU.

2.1.¿ QUE ES LA A.I. DE CMU?.

La "Artificial Intelligence" (A.I) en Estados Unidos es una corriente minoritaria dentro de la "computer science" , pero tiene unas características particulares que la destacan sobre el resto de la comunidad.

En primer lugar, es una corriente fundamentalmente académica, enraizada en el medio universitario, no empresarial. Se concentra principalmente en una élite de centros tecnológicos de excelencia de Estados Unidos: M.I.T., Stanford, y Carnegie Mellon. Se podría considerar la intelligentsia de la "computer science". De sus líderes, Herbert Simon, Allen Newell, John McCarthy, Marvin Minsky , uno de ellos es Premio Nobel en Economía, H. Simon, y todos ellos son miembros de la National Academy of Science y , a su vez, de la National Academy of Engineering, excepto Simon que sólo pertenece a la primera.

No obstante, ninguna de estas universidades donde la Artificial Intelligence es fuerte, pertenece a la Ivy League, el núcleo de las ocho universidades encabezado por Harvard, Yale y Princeton donde se forma tradicionalmente la clase dirigente norteamericana. A su vez, los líderes de esta corriente, a excepción de Herbert Simon, no ocupan un papel destacado de liderazgo de la comunidad científica norteamericana, encabezada por los físicos. Más bien ocupan un lugar un intermedio entre la comunidad de científicos y la de ingenieros. Considerados demasiado ingenieros por los primeros y excesivamente científicos por los segundos.

En segundo lugar, este carácter principalmente académico se ha debido en gran parte a que durante tres décadas la AI ha sido un campo de investigación estratégica, de alto riesgo, financiada exclusivamente por DARPA, en el que sólo empresas aisladas como Digital, con fuertes conexiones académicas, se han atrevido a invertir.

En el documento denominado "Proposal for Continuation of Research in Information Processing" de Marzo de 1975, sometido por CMU a la Advanced Research Projects Agency del Department of Defense podemos leer: "CMU has a long history as a center for research into the nature of intelligence, both in humans and computers. It is often referred to , along with the laboratories at Stanford, MIT and SRI, as one of the four ARPA AI Laboratories..." (CMU-CS. Proposal 1875-76:8).

Pero la AI de CMU tiene características propias distintas a las otras universidades con que compete, en especial Stanford y M.I.T.

Desde la formación del Computer Science Department, la Artificial Intelligence de CMU ha sido el grupo de computer scientists que ha ejercido el liderazgo en la investigación del citado departamento y luego de la School. Esto no es un rasgo común al resto de departamentos de informática del país, sino una excepción a la regla.

Esta fuerza de la AI dentro de la Computer Science en CMU, y en todo el país, proviene, entre otros factores, del hecho de que fue el denominado 'RAND-Carnegie group', formado por Newell, Shaw y Simon, los que iniciaron la disciplina en 1955, gracias a la invención del Logic Theorist (LT), el primer programa de ordenador inteligente. El LT representa para la Artificial Intelligence lo que ENIAC, el primer ordenador digital significa para la tecnología informática.

La propia estructuración de la AI en CMU es anómala. A diferencia de MIT o Stanford, donde sus respectivos líderes están dentro de laboratorios de computer science, en CMU se considera AI el trabajo realizado tanto en Computer Science, liderado por Allen Newell y Raj Reddy, como el que realiza Herbert Simon, en el Department of Psychology, al que también pertenece Allen Newell.

Así se explicita en la propuesta enviada a DARPA, y citada anteriormente:

"The totality of research at CMU that should be discussed under the rubric of Artificial Intelligence actually occurs in three separate places: the work on human cognition, going on the Psychology Department and supported by other than ARPA funds; the work in Speech Understanding Systems,...and the work described in this section, officially labeled "artificial intelligence". (ibid:8).

La excepción consiste en considerar que un científico, Herb Simon hace también A.I. al estudiar la cognición humana. El carácter más científico de la A.I. en CMU es la anomalía. El "scientist of the artificial" es lo extraño.

La Inteligencia Artificial (AI) es dentro de las áreas de la SCS el grupo más numeroso no sólo de profesores sino también de estudiantes. Es el que aporta el mayor número de proyectos y por lo tanto de financiación. Es una de las dos áreas fundacionales de lo que fue el antiguo Department of Computer Science. Como recordamos se formó por Allen Newell y Herbert Simon que provenían de esta área y por Alan Perlis, primer director del departamento, y cuyo trabajo estaba dedicado principalmente al área de los lenguajes de programación, hoy

considerada dentro de Programming Systems. El punto de unión de ambas corrientes era su interés en el "software".

Esta área de investigación está formada por 35 profesores e investigadores (Faculty Research Guide.1991) . De estos 24 se dedican también a supervisar la investigación de un conjunto de 90 alumnos que los han elegido como "advisors". Estos 90 estudiantes de postgrado son el 39% de un total de 227 en enero de 1991, del conjunto de este programa de doctorado.

Este núcleo de profesores es el que sostiene en el área de Artificial Intelligence en 1991, las dos actividades fundamentales propias de esta escuela: la investigación y la dirección de las tesis doctorales de los futuros "computer scientists". Son los responsables de los principales proyectos de investigación y de la dirección de los futuros "computer scientists" en esta área. Estos son sus nombres: Joseph Bates, Hans Berliner, Jaime Carbonell, Scott Fahlman, Mark Fox , Dario Giuse, Katsushi Ikeuchi, Bonnie John, Takeo Kanade, Kaifu Lee, Matthew Mason, David McKeown, Tom Mitchell, Allen Newell, Raj Reddy, Alexander Rudnicky, Jeffrey Schlimmer, Steven Shafer, Reid Simmons, Herb Simon, Masaru Tomita, David Touretzky, Alexander Waibel y Andrew Witkin.

De este grupo, si nos fijamos en la relación con los estudiantes como "advisors" vemos que hay grandes diferencias en cuanto al número de los estudiantes que cada uno dirige, lo cual constituye un dato significativo. En el curso académico 1990-91 los profesores Jaime Carbonell y Tom Mitchell supervisaron grupos de 13 y 12 estudiantes de doctorado respectivamente, los grupos más numerosos , no sólo de esta área sino de toda la escuela.

En ese año, un reducido nucleo de 7 profesores del grupo de Inteligencia Artificial (un 20% del total) , los dos citados más A.Newell (8 alumnos), A. Witkin (7), M. Tomita (6), T. Kanade (5) y A. Bates (5) supervisaron un total de 56 estudiantes, el 62% del total de graduate students del área de Inteligencia Artificial. Si analizamos a continuación los intereses de estos profesores podemos hacernos una idea de las áreas que suscitan más interés en los estudiantes de A.I. en CMU.

Allen Newell, Jaime Carbonell y Tom Mitchell participan junto con otros profesores (H. Berliner, D. McKeown, H.Simon, y K Van Lehn) en un programa de investigación básica denominado: "INTEGRATED ARCHITECTURE FOR INTELLIGENT SYSTEMS"(1990-1993). Como primera definición podríamos decir que se trata de construir programas de ordenador que puedan desarrollar una conducta inteligente

similar a la de un ser humano. En particular, Jaime Carbonell y Tom Mitchell, han sido protagonistas en la puesta en marcha durante los años 80s de una nueva subárea dentro de la A.I. denominada: "MACHINE LEARNING", cuyo objetivo es construir máquinas que puedan aprender, "one of the defining features of intelligence" (Carbonell, 1987: 464).

Por su parte, Andrew Witkin y Takeo Kanade trabajan en el campo del "COMPUTER VISION", que trata de conseguir que los ordenadores puedan adquirir una visión igual o mayor que la de los humanos. Ambos, junto a otros investigadores (S. Shafer, K. Ikeuchi), participan en el programa "IMAGE UNDERSTANDING"(1990-93), que desarrolla estos objetivos.

A su vez, Takeo Kanade, director del programa y co-director del Robotics Institute, ha colaborado con T. Mitchell, y Red Whittaker, ingeniero civil, en la construcción de AMBLER, un robot inteligente y autónomo, finalizado en 1990 en CMU y destinado a la exploración espacial dentro de los programas de la NASA. Masaru Tomita y Jaime Carbonell, trabajan en el área de MACHINE TRANSLATION, cuyo objetivo es construir máquinas que puedan traducir automáticamente lenguajes naturales. En una reciente reunión en Washington, the Machine Translation Summit III, Jaime Carbonell presentó el sistema KANT, construido en colaboración con el lingüista T. Mitamura y el programador E. Nyberg III, que puede asimilar información técnica en inglés y generar un texto en japonés, alemán o francés. Para esta tecnología, se basan en una subárea de la Inteligencia Artificial denominada NATURAL-LANGUAGE PROCESSING (NPL), definida por J. Carbonell y Philips Hayes como "the formulation and investigation of computationally effective mechanisms for communication through natural language". (1987: 661).

Alan Bates, coordinador, junto con Tom Mitchell, del área educativa de la A.I. en la School, para el curso 90-91, se dedica a la nueva área de los INTERACTIVE MULTIMEDIA. Estos últimos años han surgido los primeros ordenadores que pueden procesar imágenes, no sólo texto o gráficos. Este es el campo llamado "MULTIMEDIA", y que emparenta con la televisión. La diferencia estriba en que en el ordenador la persona puede manipular, cambiar, crear, las imágenes que aparecen en la pantalla. Alan Bates se dedica a crear estos medios visuales interactivos, como MetaPrl y Sythetic Realities, en palabras de Bates "Paper and film provide ways for people to express their thoughts and fantasies. Computers may provide equally significant means of expression by supporting interaction

with those thoughts and fantasies 'brought to life'. I work in the areas of interactive environments and artificial intelligence, with the goal of creating these new interactive media." (CMU-CS.FRG, 1990-91:3).

Por ultimo "SPOKEN LANGUAGE UNDERSTANDING", es un programa de investigación dirigido por el Dr. Raj Reddy y su objetivo es la construcción de un sistema que sea capaz de reconocer la voz humana, en concreto 5.000 palabras con un acierto del 95%. Forma parte del área denominada "Speech" en A.I.. Su equipo esta formado en 1990-91, por los profesores Kaifu Lee, Alexander Rudnicky, Richard Stern, Wayne Ward y Sheryl Young.

Estos programas, particularmente tres de ellos: "Integrated Architectures", "Image Understanding" y "Spoken Language Understanding" engloban a los profesores de A.I. con una mayor experiencia en el campo de la investigación en esta disciplina. Y sus directores Allen Newell, Takeo Kanade, y Raj Reddy se pueden considerar líderes representativos de la investigación en la A.I. de esta comunidad.

Estos programas de Inteligencia Artificial son el resultado de un programa de investigación que fundó la disciplina y que se inició en 1955-56 con la invención del Logic Theorist por Allen Newell, Cliff Shaw y Herb Simon. Programa que constituye el patrón cultural dominante sobre el que se ha construido la A.I. en CMU.

2.2. LA CONSTRUCCION DE LA "LOGIC THEORY MACHINE" : EL PROGRAMA CULTURAL INICIAL DE LA "ARTIFICIAL INTELLIGENCE" DE CMU.

2.2.1. "THE PARENTS OF THE ARTIFICIAL INTELLIGENCE".

La Inteligencia Artificial es una nueva disciplina, un nuevo sistema de conocimiento, en torno al cual se ha organizado inicialmente en Estados Unidos y luego en Europa y Japón, un grupo de personas para su desarrollo. Constituye por lo tanto una cultura particular, una construcción humana histórica, que puede ser analizada por el antropólogo como una innovación cultural, dentro de lo que serían las "computer cultures".

A efectos de un estudio antropológico, la construcción del programa de Inteligencia Artificial, la "Logic Theory Machine", inventado por Allen Newell, Cliff Shaw y Herb Simon en 1955-56, lo podríamos considerar como una de las bases para la confección del modelo etnográfico de esta comunidad.

Al referirnos a la "Logic Theory Machine" entendemos el programa de investigación y desarrollo que dió lugar a dicho programa informático.

El análisis de este programa puede aportar esta base, máxime cuando el mismo fue creado por dos de los fundadores del Computer Science Department, Allen Newell y Herb Simon, aún hoy dos activos investigadores con 64 y 75 años, y que se consideran a sí mismos como "parents" ("padres"), de la disciplina (Simon, 1991:190). La relación "parents"- "padres"- "patrones culturales" nos parece, no sólo etimológicamente ajustada, sino además, en este caso, culturalmente apropiada. Aunque la influencia de estos científicos va decayendo con los años, aún siguen siendo considerados por la nueva generación de la escuela como un modelo de referencia respetado.

Es evidente que todo modelo etnográfico es una reducción del conjunto de conductas y conocimientos observados en una comunidad dada, pero dado que es imposible aspirar a una observación sincrónica de la totalidad de las actividades de todo un colectivo humano, esta técnica está considerada, según antropólogos como Claudio Esteva, como "la técnica de conocimiento más productiva" (1978: 233). Dentro del modelo, trataremos de obtener un patrón cultural, o modo de conducta referencial para el conjunto del grupo. La actividad de estos profesores ha seguido ella misma un patrón sustancialmente idéntico durante estas cuatro décadas. Este modelo cultural ha sido y es el fundacional de la Inteligencia Artificial en CMU, y sigue aportando el modelo de actuación principal. En antropología, un patrón cultural, como vemos en este caso, es un "model of man" (utilizando la expresión "man" común para ambos géneros).

Anteriormente ya hemos citado al antropólogo Roy D'Andrade (1981) y su noción de "programa cultural" para el estudio de los sistemas culturales en general.

Estos, definidos hasta ahora como sistemas de conocimientos organizados, se podrían ver como programas organizados, esto es, como programas culturales compartidos. D'Andrade diferencia programas culturales y programas informáticos, considerando los primeros como "programas humanos" y los segundos, los "computer programs". Pero, como ya hemos analizado antes, esta diferencia no es del todo ajustada.

Los computer programs, como el caso del Logic Theorist muestra, son también programas culturales, un tipo particular de ellos. Son productos humanos, obras humanas. La oposición programa humano-programa informático se podría superar si

consideráramos ambos como subconjuntos de un programa mayor: el programa cultural.

Todos los programas informáticos son programas culturales pero los programas culturales no han sido hasta ahora programas informáticos.

Carnegie Mellon es una "computer-intensive university". Su cultura, sus hábitos de trabajo, sus modos de comunicación, sus costumbres ("mores") y por lo tanto su moral, finalmente, está siendo regulada parcialmente ya por un programa informático. Su computer environment, Andrew, se rige en efecto por un programa informático.

No obstante éste ha sido hecho por unos computer scientists que trabajan en una institución con un programa cultural anterior a Andrew, el programa pre-informático de CMU como institución iniciada en 1900 y sus sucesivas actualizaciones, a su vez, concreción de todo un programa cultural característico de Norteamérica.

Se puede pensar que ahora puede iniciarse la construcción de programas culturales, los programas de las tecnoculturas, que sean realizados conjuntamente por seres humanos y ordenadores inteligentes. Hasta ahora podíamos pensar que los programas culturales simplemente se heredaban. Los construía la historia. Luego se pensó que los construía la naturaleza y nos adaptábamos a ella. Aún hoy se habla de Andrew como "environment". Un medio al que adaptarse. Pero lo cierto es que este "environment" ha sido una decisión humana, deliberada, una innovación, no una mera adaptación.

El Logic Theorist fue a su vez otra innovación. El resultado de la acción voluntaria y deliberada inteligente de un determinado colectivo humano organizado sobre su propio proceso de pensamiento, y sobre el medio externo. En estos conjuntos, la iniciativa para formar nuevos programas informáticos la tienen los seres humanos. Son los elementos generadores del proceso, los inventores, sus autores, lo que les permite ser responsables de sus actos.

Mientras la ciencia era sólo observación y análisis podía equivocarse o no en sus predicciones, pero no era responsable. Cuando la ciencia pasó a diseñar en los años 40s, bombas atómicas o ordenadores, cuando empezó a actuar, a transformar, se convirtió en agente del cambio cultural. El problema de la responsabilidad, antes circunscrito a la figura del profesional, con su código de ética profesional, se ha extendido a la ciencia. La ciencia se ha hecho programadora. Y el "computer science" es una prueba de ello.

El uso del concepto "programa cultural" es particularmente útil para el estudio de la cultura informática. El concepto de programa permite entender mejor que la cultura es una actividad humana normativa que se caracteriza por su artificialidad, contingencia y posibilidad de acción innovadora sobre el medio.

En las computer cultures, la actividad del científico es claramente política ("policy"). Como afirma el antropólogo Erve Chambers: " The value of forming an idea of policy is that it enables us to start thinking about human issues and problems in terms of their intentionality. Programs and activities of change do not just happen. They are thought out, planned out, and put into action by human beings for particular reasons" (1985:38).

O como afirma Harold Abelson, profesor del Department of Electrical Engineering and Computer Science, del M.I.T.: "Computer Science is the business of formalizing imperative knowledge..." (Abelson, H.1991:193)

No obstante no hay que olvidar que un programa es una función de un sistema en el caso de una tecnología o de una comunidad en el caso de una ciencia social. Logic Theorist era una Logic Theory Machine, el olvido, más tarde del último término, trajo consecuencias graves para la A.I.

La Inteligencia Artificial la empezó un grupo de científicos y programadores decididos a construir un programa informático que podía ejecutar en el ordenador una función inteligente que sólo el ser humano había realizado hasta la fecha: encontrar una prueba de un teorema de lógica matemática.

Esta prueba se realizó el 9 de agosto de 1956 en un ordenador JOHNNIAC en la RAND Co. Fue el resultado final de un programa cultural iniciado por tres personas en torno a una investigación denominada "complex information processing". La secuencia podría describirse así. Un equipo de científicos vislumbraron la posibilidad de inventar una máquina con esas características.

Como dice Simon: " The sudden and permanent change came about because Al Newell, Cliff Shaw and I caught a glimpse of a revolutionary use for the electronic computers...We seize the opportunity to use the computer as a general processor for symbols (hence for thoughts) rather than just a speedy engine for arithmetic."(Models of My Life.1991: 189).

No había realidad natural preexistente a su invento. Vieron una posibilidad plausible.

A continuación , tras distintos ensayos y fracasos, seleccionaron una tarea que vieron realizable. Una vez seleccionada, uno de ellos resolvió dicha tarea

simulándola, a la vez que analizaba su propio proceso de resolución de esta tarea . Mientras, los otros miembros del equipo construyeron un nuevo lenguaje de programación para introducir esta forma de pensamiento en un ordenador. Finalmente comprobaron si funcionaba o no el invento en la máquina. Y funcionó el 9 de agosto de 1956.

En palabras de Simon:" By the end of 1955 we had invented list-processing languages for programming computers and had used them to create the Logic Theorist, the first computer program that solved non-numerical problems by selective search. It is for these two achievements that we are commonly adjudged to be the parents of artificial intelligence." (ib:189-190).

Con ello no sólo construyeron un artefacto inteligente sino al mismo tiempo una disciplina para construir estos artilugios, la Artificial Intelligence.

Analicemos más en concreto el proceso cultural de invención.

2.2.2. EN LA RAND CORPORATION. : DEL DISEÑO DE ORGANIZACIONES AL DISEÑO DE UN PROGRAMA INFORMÁTICO INTELIGENTE.

Herb Simon cuenta en su autobiografía que todo empezó en el verano de 1954 en la RAND Corporation en Santa Mónica, California: " Al and I had a long discussion of the possibility of simulating human thinking by computer" (1991:201). Diez años atrás Simon concedió una entrevista a un periódico de Pittsburgh, the Post-Gazette, el 7 de julio de 1980, donde afirmaba: "The work really began in Santa Monica about 1955...Al Newell and I began to talk about the possibility of writing a computer program that would solve problems".

Este modelo empieza pues con la misma pregunta que se formulara Allan Turing en 1950: "Can machines think?", aunque, según los autores, la influencia en ese periodo de Turing sobre estos personajes fue mínima o nula.

Quizá Simon y Newell no lo supieran en ese momento, pero esta posibilidad ya la había planteado A. Turing cuatro años antes, en 1950. Lo interesante a resaltar aquí es cómo aparece esta posibilidad a los ojos de Herbert Simon.

Según ha escrito en su autobiografía, "The completely new turn that my life took in 1955 was the unanticipated result of my work in the Systems Research Laboratory at RAND and my contact there with computers" (1991:198). Para este científico, Premio Nobel en Economía en 1978, : " The most important years of my life as a scientist were 1955 and 1956". (ibid.:189). Los años de la invención del Logic Theorist. Herb Simon tenía en 1955, 39 años y Allen Newell, 28.

Pero la razón por la que Simon y Newell se conocieron en la RAND Co en febrero de 1952 no fue el ordenador sino por el hecho de estar ambos implicados en un proyecto de la Defensa: el diseño de una estación de seguimiento aéreo de la Air Force para estudiar en ella el comportamiento humano en relación con la máquina en condiciones de stress.

La RAND Co. a finales de los 50s era un nuevo tipo de institución, "a nonprofit corporation formed to further and promote scientific, educational, and charitable purposes, all for the public welfare and security of the United States of America" (Articles of Incorporation). (The Rand Co.1963: The First Fifteen Years). La cultura de la RAND Co. afectará directamente al tipo de investigación a realizar por el Department of Computer Science de CMU, y a la propia estructura de esta institución académica. Allen Newell, en particular, afirmará: "In relation with problem solving, my influence is from the Rand Co, not from the Doherty Plan".(Entrevista 17 dic. 1990).

La RAND Co era entonces un nuevo tipo de institución, el original "think tank", "a university without students", cuyo propósito era la investigación aplicada con fines nacionales estratégicos. RAND era un acrónimo de "RESEARCH AND NATIONAL DEVELOPMENT". El propósito de su constitución era preservar el cuerpo de científicos que había participado en la guerra en tareas como el Manhattan Project, y seguirlo interesando tras la finalización de esta en tareas de investigación al servicio de la defensa nacional: "In postwar years the United States has undertaken vastly expanded responsibilities involving the well-being of much of the world...It was clearly important to preserve part of the scientific corps that had been mobilized during the war to develop the atomic bomb...That work had been done though emergency organizations, such the Manhattan District and the Office of the Scientific Research and Development...With victory, most scientists were anxious to return to former pursuits, and it was imperative that some cohesion be maintained among them...Various settings were considered after the war to a permanent research organization of a new kind" (ib: 6). Y así nació RAND Co.

En este centro, se mantendría el carácter aplicado de la investigación científica, no ya al servicio de la industria, sino de los intereses nacionales estratégicos. En ella tenían cabida, en proyectos interdisciplinares las siguientes especialidades: " Aero-Astronautics, Computer Science, Cost Analysis, Economics, Electronics, Logistics, Mathematics, Physics, Planetary Sciences,

Social Sciences and System Operations". (ib:2-3).

En esta institución, se empezó a desarrollar a fines de los 40s un enfoque denominado "system analysis", que progresivamente iría ganando terreno al anterior, la "operation research", utilizado ante todo durante la II Guerra Mundial. Los problemas de la postguerra se fueron viendo de naturaleza más compleja que los de mera optimización de las fuerzas militares en campaña. Uno de los subproductos de este enfoque fue el Systems Research Laboratory, cuna del programa de Inteligencia Artificial de CMU.

Mientras que la "operation research" consistía en una técnica matemática que proveía de ayuda cuantitativa para los "decision makers" militares en condiciones de guerra, decisiones caracterizadas por tener objetivos claros, una vez finalizada la II Guerra Mundial, las necesidades de planificación militar se transformaron, convirtiéndose en decisiones cada vez más de tipo político. (La operation research ha seguido siendo utilizada en los problemas del management industrial, especialmente a los niveles medios y bajos, donde el objetivo está claro ("profit") y se puede medir en unidades de análisis no ambiguas ("money"). Como explica Bruce L.R. Smith, historiador de este periodo de la RAND Co, "this type of approach has limited utility on the broad policy questions where objectives cannot be precisely specified and no "optimum" solutions are possible." (1966:162). Esto llevó al desarrollo de un nuevo enfoque denominado "system analysis". Este era un método menos cuantitativo y más orientado hacia "the analysis of broad strategic and policy questions, and particularly in seeking to clarify choice under conditions of great uncertainty" (ibid:8). "The useful systems analysis in this area typically attempts to identify relevant policy alternatives under conditions of great uncertainty, and helps policy maker to assess the costs and consequences of different courses of action" (ibid:162). Este es el nacimiento de lo que hoy se denomina "policy analysis".

"The task of sytem analysis work, indeed, is principally that of integrating a variety of specialized research skills into a common analytical framework bearing on an important problem. The aim of such work is to be precise where possible and where precission is ruled out by a context of conflicting values and unclear objectives, analysis strives to identify the many complex dimensions of a critical policy choice and to help the policy maker assess the costs and consequences of alternative courses of action". (Smith: 10).

Esta cultura de integración de elementos de distintas disciplinas para la solución de un problema importante, es similar a la del "total system effort" que se sigue en los proyectos de investigación de la SCS.

La RAND Co. en los años 50s fue, a su vez, la cuna de una mentalidad estratégica, derivada de su interés por la elaboración de alternativas políticas de alcance nacional: ""This strategic consciousness serves to distinguish RAND from some of the other advisory corporations whose major concern is more or less exclusively with nonstrategic matters". (Smith: 105).

Esta orientación estaba claramente orientada al futuro, siendo uno de los centros donde nacieron las técnicas de " technological forecasting" (Jantsch, E., 1967), así como la "social technology" (Olaf Helmer, 1966), el método Delphi, en suma, las bases de la cultura futuroológica que se desarrollaría en los 60s. Nació el científico del diseño.

La investigación que más tarde daría pie a la Inteligencia Artificial de CMU nació en 1950 en el marco de esta cultura propia de la RAND. "In 1950 a team of psychologists began to study how men and machines works together; in particular, how men respond to conditions of stress. They simulated an Air Defense Direction Center, and used college students as experimental subjects" (The Rand CO. 1963: 17). Allen Newell, graduado en Físicas por Stanford y con un año de matemáticas en Princeton, fue contratado en 1950 por la RAND incorporándose a ese proyecto de científicos sociales. Dos años después solicitaron la colaboración de Herbert Simon como experto en organizaciones.

Se trataba de un tipo de investigación para "immediate application", como era característico de la cultura de la RAND Co.. Como reconoce Herbert Simon, la RAND Co. gozaba en ese tiempo de "tremendous success and visibility as a new way of enlisting research talent to help advance applied goals". (Models.1991:164).

Durante los veinte años anteriores, la investigación principal de este profesor Simon había tratado con organizaciones y cómo las personas que las dirigen toman decisiones. De esa época arranca su concepto del hombre como animal de "bounded rationality" expuesto en su libro "Administrative Behavior" (1948). Partiendo de una crítica de la visión neoclásica del Economic Man, Simon ofrecía un nuevo model of man basado en un nuevo tipo de conducta, la "administrative behavior", la propia del animal político , más dedicado a la toma de decisiones ("decision making") en un limitado campo de opciones, que a una búsqueda incesante de

optimización de sus beneficios económicos en el mercado.

Este interés por la ciencia política y secundariamente por la economía es lo que le llevó a colaborar con la Comisión Cowles y con la RAND Co como experto en organizaciones. Ambas eran instituciones que trataban de asesorar a organismos políticos de la Administración y de la Defensa sobre "decision making" en temas económicos y estratégicos.

Una vez en RAND, y después de tres años intentando formalizar las diferentes comunicaciones entre los miembros del laboratorio sin conseguirlo, cambiaron de terreno de investigación: " Yet AI and I suffered from continuing frustration in trying to write formal descriptions of the process. Somehow, we lacked the necessary language and technology to describe thinking people as information processors...The frustration that AI and I experienced with the SRL experimental data had major consequences". (ibid: 168).

En un borrador de su biografía escrito en 1977, y hoy conservado en sus archivos personales en CMU, H. Simon explica su nuevo centro de atención : "At Systems Research Laboratory, I became fascinated by the method that AI Newell had devised for using a Card Programmed Calculator to produce the simulated radar maps needed for the air defense simulation. What was remarkable about this application was that the computer was being used not to generate numbers, but locations--points--on a two-dimensional map. Computers, then, were not merely number crunchers; they were general symbol manipulators, capable of processing symbols of any kind --numerical or not! This insight, which dawned only gradually, led AI and me even more gradually to the idea that the computer could provide the formalism we were seeking --that we could use the computer to simulate all sorts of information processes, and use computer languages as formal descriptions of those processes" ("Choices and Paths,1977:31).

En su más reciente biografía escribe: "All this changed radically in the last months of 1955...This sudden and permanent change came about because AI Newell, Cliff Shaw and I caught a glimpse of a revolutionary use for the electronic computers that were just then making their first public appearance. We seized the opportunity we saw to use the computer as a general processor for symbols (hence for thoughts) rather than just a speedy engine for arithmetic.". (1991:189).

La idea de "opportunity" nos parece clave en este texto. Puede traducirse al castellano por "oportunidad", y también por "posibilidad, ocasión". Esta palabra

tiene un significado cultural en la sociedad norteamericana que es distinto al que tiene en España. La cultura norteamericana, tradicionalmente, se ve a sí misma como "The Land of Opportunities", el país donde "everything is possible". Ser "oportunist" tiene una connotación en la cultura europea, al menos la española, de carácter peyorativo. Es sinónimo de abandonar los principios, la moral. En Estados Unidos, es lo contrario: socialmente está bien visto el saber ver una posibilidad única y sacar provecho de ella ("take advantage"), tanto en el aspecto económico como intelectual. Si ésta es la tierra de las oportunidades, ¿por qué no aprovecharlas?. La pregunta en la cultura norteamericana no es tanto la búsqueda de un "¿por qué?" sino un oportunista "Why not?". No es un interés tanto por los orígenes históricos o genealogías de las cosas sino a preguntarse por los "purposes", los "goals", los "desires", los objetivos de la acción humana, en suma, por su futuro.

Pero la idea de "opportunity" indica aquí en concreto no una posibilidad cualquiera, sino una posibilidad plausible, realizable. Esta consiste en una alternativa ya seleccionada, no realizada aún, pero no un simple sueño o una fantasía. Las posibilidades son múltiples, innumerables. La oportunidad es una, singular, particular, en un momento dado. De entre todas las posibilidades se selecciona una que se ve con visos de poderse realizar, a esta alternativa la denomino: posibilidad plausible, y es clave para entender una mentalidad tecnológica. Es sinónimo del término, "feasible" (M. Furst), realizable.

Ed Feigenbaum, alumno de Simon en 1955, ha sido el que ha recordado la expresión utilizada por Herb Simon, la primera clase después de las vacaciones de Navidad del curso 1955-56: "Over Christmas, Al Newell and I invented a thinking machine". (1989:166).

2.2.3. DISEÑO Y ADAPTACION: LA DOBLE PAUTA CULTURAL.

Para la mentalidad científica clásica más determinista, la tecnología se muestra, a modo de "naturaleza", como un proceso objetivo que discurre independientemente de los hombres, y ante la que éstos han de adaptarse. Ello explica que, hasta la fecha, la función principal de las ciencias sociales que pretenden acercarse al estudio de la tecnología ha sido el estudio de los "efectos sociales" de la misma en los seres humanos.

Se considera que el ser humano va detrás de la tecnología. La antropología ha colaborado en el campo industrial sobre todo a establecer el área de "human

relations", que se preocupa de adaptar el personal de la empresa a los cambios tecnológicos, mediante enseñanza y aprendizaje continuo. Es el campo que estudian habitualmente los psicólogos y científicos sociales aplicados que trabajan en la industria.

Ello se deriva de la visión que ellas tienen, siguiendo a Darwin, del ser humano como un animal fundamentalmente adaptativo a los cambios del medio ambiente. Esta fue a su vez la clásica la visión de Marx sobre el movimiento de las fuerzas productivas que empezando por las máquinas y su incesante cambio arrastraban y cambiaban toda la estructura social, visión compartida por el determinismo cultural de Leslie White y su escuela.

Esta visión del mundo se opuso al creacionismo, a la visión de un mundo fijo creado por Dios, doctrina que proporcionaba a la Iglesia su poder espiritual básico en el Ancien Regimen. Esta teoría ha servido para un estudio del cambio de las especies naturales sin necesidad de intervención del primer "diseñador". El problema es que ahora el ser humano empieza, mediante la biotecnología, a diseñar especies, y esta teoría se ha quedado insuficiente para explicar el nuevo mundo artificial, surgiendo de las nuevas tecnologías.

La tecnología es un invento humano . No es una adaptación pasiva al medio, sólo. Es un cambio innovador del mismo. O, si se prefiere la construcción de un nuevo medio. Newell, Simon y Shaw inventaron el LT, como CMU ha inventado Andrew. No sólo se adaptó a los cambios del medio preexistente, adaptó ese medio a sus propósitos. Han inventado un nuevo medio ambiente: "a computer-intensive campus".

Antes de 1955 no había Inteligencia Artificial, después la hubo. Si como antropólogos estudiamos como se generó esta tecnología podemos entender como unos seres humanos inventan tecnología.

La visión a posteriori de la tecnología afecta hasta a los propios tecnólogos. Estos, en gran parte, construyen su imagen personal sobre la que le proporcionan los científicos sociales que los estudian. Se ven a sí mismos como científicos, como si fueran físicos o químicos, lo cual está favorecido, porque la ciencia es un tipo de saber con más valoración académica que la ingeniería. El propio Herbert Simon que se define como un científico (Entrevista personal ,17 dic. 1990) , sin embargo, reconoce que los años marcados por la invención junto con Newell Y Shaw de la Inteligencia Artificial fueron los más importantes de su vida.

Por otra parte, la mente artística o humanística suele adoptar una doble actitud ante la tecnología: o bien la mira con recelo, al considerarla "inhumana", o bien la consideran aún como arte, como una expresión de un poder humano de transmutar las cosas casi "mágico", "divino". En ello se acercan a la visión del hombre corriente de esta cultura que ante la tecnología oscila entre el terror y el visceral rechazo o su asombro y abandono total ante lo que considera "fantástico", "incomprensible", o "misterioso". Edison era conocido popularmente en América como "The Wizard of Menlo Park", el mago de Menlo Park. Todavía en 1970, ingenieros como Alvin S. Weinstein y Stanley W. Angrist, profesores de CMU definían así al ingeniero: "The engineer is an artist who begins with an idea or a need, and using his special tools and raw material, he creates". (1970:2). En efecto magia y tecnología han estado ligadas estrechamente en las culturas primitivas, pero no como un rasgo propio de los dioses sino algo específicamente humano. Malinowski ya indicó que la magia, en toda tradición, en toda mitología, es siempre posesión del hombre: "La magia no resulta derivada de una observación de la naturaleza o del conocimiento de sus leyes, sino que es posesión primigenia de la raza humana que sólo puede conocerse mediante la tradición, y que afirma el poder autónomo del hombre para crear los fines deseados" (Magia, Ciencia, Religion, 1982: 89).

Hoy sabemos que magia y tecnología forman mundos separados. La creencia en la omnipotencia del pensamiento es una característica del mago pero no del tecnólogo, o al menos no de la mayoría. La ciencia nos ha enseñado a diferenciar nuestros deseos y un mundo objetivo que no se transforma por el mero hecho de que así lo queramos. El invento sabemos que es, como dijo Edison, un uno por cien inspiración y un 99 por ciento transpiración.

La Inteligencia Artificial es un invento técnico humano, no un acto mágico. De la misma forma, no podemos verla ya como un mero descubrimiento científico. Sus protagonistas son contemporáneos nuestros, aún viven. Han relatado sus inventos, han escrito miles de páginas que constituyen un material inapreciable para un estudio etnográfico detallado, y a través del cual empezamos a entender mejor la mentalidad tecnológica, y su relación con la cultura norteamericana. Este estudio es sólo una introducción a un programa de investigación a largo plazo sobre las tecnoculturas.

El proceso general que hemos descrito en palabras de H. Simon se resume así: 1. Se empieza a trabajar en un medio nuevo (los ordenadores). 2. Se vislumbra una

posibilidad de cambiar ese medio, introduciendo una innovación. 3. Se utilizan todos los conocimientos y habilidades que se tienen para llevarla a cabo. 4. Se termina diseñando dicha innovación, un objeto, programa o lenguaje que antes no existía. 5. Se prueba y se comprueba que funciona.

Podrían haberse adaptado sin más al nuevo entorno, tratando tan sólo de comprender mejor empírica o lógicamente sus propiedades. Pero entonces no habrían inventado LT. También podrían haber deseado con todas sus fuerzas construir una máquina pensante sin tener en cuenta la realidad del medio informático, o sin poner a prueba sus primeros experimentos mentales. En ese caso, se habrían quedado en meros esbozos sobre el papel como los dibujos de helicópteros de Leonardo. Pero no habrían inventado LT.

Imaginaron, analizaron empíricamente, y además consiguieron hacer una síntesis entre el medio que tenían ante sí y sus deseos y fantasías. El resultado : han engendrado una nueva tecnología, esto es, un nuevo medio , un nuevo "environment".

Veamos más en concreto cual fue el proceso de la invención.

2.2.4 "USING MYSELF TO SIMULATE MACHINE": HERBERT SIMON.

La invención consistía en la solución de dos problemas principales: uno, el proporcionar al programa las vías para resolver problemas intelectuales sencillos ,y dos, inventar un lenguaje nuevo para introducir al ordenador las instrucciones a cerca de su ejecución.

Cliff Shaw y Allen Newell tomaron la principal responsabilidad en este último tema, Herbert Simon y Al Newell en el primero.

La selección de la tarea fue una primera limitación del campo de posibilidades. Se eliminaron las más difíciles como el juego del ajedrez, quedándose con aquellas que se veían más aptas para conseguir el objetivo previsto. La tarea seleccionada fue la de descubrir pruebas de teoremas de lógica simbólica.

Una vez seleccionada la tarea, el procedimiento fue el siguiente: " We spent some time introspecting about our own mental processes while solving problems in different domains. By the summer of 1955, discovering proofs for theorems, either in geometry. or in symbolic logic, appeared to be a likely candidate. (The accident that I had the volumes of Principia Mathematica on my bookshelves led us to the latter task)." (Choices and Paths, 1977:32). .

A través de esta introspección, Herb Simon derivó la principal forma de trabajo

del LT, el trabajo "hacia atrás" o "backwards". Su característica era el trabajar desde los fines perseguidos, el teorema a probar, hacia atrás: los pasos necesarios para ello. No se trataba pues de un método secuencial, algorítmico, o progresivo: desde el estado inicial al final, sino que realiza una inversión del proceso: partía del estado final y considerando el estado inicial seguía una secuencia de pasos para conseguir el objetivo deseado. Este proceso se lo introdujo a la máquina.

En 1956 cuando se trató de generalizarlo como una teoría general del proceso de pensamiento humano, Newell y Simon se dieron cuenta que los individuos que resolvían problemas de lógica en el laboratorio no usaban el mismo proceso que la máquina, esto es, que el propio Simon al programar la máquina. La opción tomada no fue intentar cambiar el funcionamiento mental de esos individuos sino el de la máquina. Se introdujo el método "means-ends analysis", y se abandonó el anterior, que era la forma básica de trabajo del LT.

Paul R. Cohen y Ed. Feigenbaum, en su Handbook of Artificial Intelligence (1982) constatan el hecho: " When they tested their theory, they found that it failed in one respect: Humans did not use the same control process (working backward from theorem to axioms) as the program. Thus, they revised the theory, and wrote a new program, to incorporate what they had learned about human control processes during problem solving. They called the new program the General Problem Solver (GPS) , and the new control process means-end analysis, and found that this process was much more efficient (in terms of computer time) than its predecessor ". (Barr, A, & Feigenbaum, E.The Handbook of A.I.1982, vol. 3:3).

Prosigamos con el método de cómo , según Simon, inventó el L.T.: " By October, we were beginning to have a feel for the problem-solving heuristics used in tackling these problems, and we settled definitely on symbolic logic. AI was then working on the programming language and taking his preliminary examinations. He and I discussed almost daily our progress toward specifying the theorem-proving program. My method of working was to take theorems in the Principia and work out proofs while trying to dissect as minutely as possible, not only the proofs steps, but the cues that led me to each one. Then we tried to incorporate what I had learned in a flow diagram. We repeated this day after day, with the flow diagram steadily approaching a description that could be programmed on the machine. On December 15, 1955, I simulated by hand a proof of Theorem 2.15 of Principia in such detail that we agreed the scheme was programmable. I have

always celebrated that date as the birthday of heuristic problem solving by computer, although it was not until the following August (1956) that the LOGIC THEORIST, programmed on JOHNNIAC, produced its first complete proof of a theorem." (1977:33).

Para un antropólogo, este proceso es el primero quizá en la historia, al menos que quede constancia, en el que un ser humano conscientemente simula en su mente el funcionamiento intelectual de una hipotética máquina, a fin de poder construirla. Se mete "dentro" del posible ordenador y simula, no su funcionamiento, dado que aún no existía, sino su posible funcionamiento.

Ello exige un desdoblamiento mental : 1. Ha de resolver el teorema hallando la prueba. y 2. Ha de analizar y describir el proceso mental de como lo ha hecho.

El lógico matemático sólo realiza el primer esfuerzo. Simon realizó dos tareas : la del lógico matemático y además una introspección, que podría considerarse psicoanalítica, de su propio proceso de solución y la describe de forma que pueda ser entendida por el ordenador. En suma una tarea de psicólogo. El resultado final se obtiene sumando las dos partes en una síntesis nueva apta para ser programada.

El resultado es doble: el descubrimiento de un problema matemático y el descubrimiento del saber como lo ha descubierto. A partir de aquí no es extraño que Simon diga que la AI se basa en la psicología y prefiera residir en el College of Humanities and Social Sciences. Y visto desde su exclusivo punto de vista no le falta razón. Si se concibe que el LT como veremos más adelante que él lo ve así, como nacido en este momento, es claro que el L.T. nació de un proceso de descubrimiento científico, basado en un autoanálisis y cuyo resultado fue el descubrimiento de unas heurísticas determinadas. Pero estos descubrimientos eran sólo parte, un subconjunto, de un proyecto tecnológico mas amplio: conseguir que una máquina razonara, inventar la A.I.

2.2.5. "THE HEURISTIC SEARCH".

La aportación clave de Herbert Simon a la construcción del L.T. fue el haber descubierto un procedimiento mediante el cual el ordenador podía resolver esos problemas de lógica. Con ello inició el paradigma de "heuristic search", una técnica particular de resolución de problemas. Para Feigenbaum, este fue el paradigma fundacional de la Artificial Intelligence en su primera época.

Las heurísticas son reglas para resolver problemas, operaciones mentales que

ayudan a conseguir resolver una tarea. Por ejemplo, "think on the end before you begin", (piensa en el final antes de empezar), el antiguo "respice finem" de los romanos. Usualmente se denominan en Estados Unidos, "rules of thumb", reglas del pulgar. Son técnicas, normas, utilizadas por los expertos para emitir un juicio intuitivo sobre la vía a seguir en la resolución de un problema, adquiridas después de una larga experiencia.

Lo más curioso es que no forman parte de los métodos científicos clásicos, sean el de empírico-inductivo o el hipotético-deductivo. La heurística, rescatada por el matemático europeo George Polya en su libro "How to solve it" (1945), presentaba una nueva forma de entender la ciencia: tratata no sólo de las matemáticas sino del proceso de invención de las propias matemáticas: "Trying to understand not only the solution of this or that problem but also the motives and procedures of the solution".(1945:vi). Normalmente, todo curso de matemáticas o física está acompañado de un apéndice con problemas que el estudiante ha de resolver. Se considera suficiente que acierte en su solución. Pero no importa el procedimiento seguido ni los motivos por los que se ha seguido ese y no otro procedimiento. Pero eso es precisamente la heurística del problema, las vías que se han seguido para su solución, tanto o más importantes, como la última, pues son las que permiten resolver otros problemas similares. Procedimientos como el trabajo hacia atrás, antes descrito, ya se conocían hacía siglos. Simon no inventó la heurística como modo de resolver problemas. Lo que sí inventó es la heurística como método utilizable por un ordenador para resolver problemas, y con ello convirtió a la heurística en una forma de pensar sistemática y de uso general en la cultura informática, al menos, norteamericana. Para ello tuvo que ingeniarse una forma concreta de representar el problema en el ordenador de forma gráfica: "el problem space". En el "problem space" se representan los objetivos que se quieren o "desired state", el estado inicial o "initial state" y los pasos para alcanzar el estado final desde donde se está. Modelo que se denomina "maze", el modelo del laberinto. Con la heurística ideó una forma de buscar soluciones a un problema distinta a la seguida tradicionalmente hasta entonces: la solución algorítmica. En lugar de buscar todas las soluciones posibles y seleccionar entre ellas la mejor, consideró que la tarea inteligente no era generar el máximo de alternativas sino alternativas sólo "plausibles", para evitar una explosión exponencial de búsqueda: "to generate only structures that show promise of being solutions or

of being along the path towards solutions". (Newell & Simon, 1976: 123).

El programa generaba un árbol de búsqueda o "exploration trees" y tenía una serie de heurísticas o reglas de conducta que determinaban de entre los subproblemas generados cual debería resolverse a continuación, proveyendo garantías de que ese camino sería válido para llegar al objetivo previsto. Por ejemplo, la técnica "best-first search" inducía a una búsqueda del nodo que aparecía más cercano al objetivo.

En su autobiografía, Herb Simon relata el cuento de Hugo, un "ordinary man" , que vive en un castillo heredado de sus padres, un enorme edificio con innumerables habitaciones que va explorando a lo largo de su vida, ocupado en la búsqueda de soluciones , hasta que al final se da cuenta, releendo el pasaje de la Biblia en el que la mujer toma la manzana y la encuentra buena para comer, que el secreto no estaba en la manzana sino en él mismo.

Herbert Simon se ve a sí mismo como un investigador, como un científico dentro de un laberinto, interesado sobre todo en "to observe people as they encounter bifurcations and try to understand why they take the road to the right or to the left" (Models , 1991:179).

No obstante, un cosa es tratar de entender , "understand" y otra, inventar un procedimiento para la toma de una decisión por el ordenador. En un momento de su vida, en los años que el califica los "más importantes", no sólo observó o se observó, sino que construyó un procedimiento para representar la solución de problemas en un ordenador. Y esta construcción fue parte de un esfuerzo de un equipo que abrió paso a la A.I..

El descubrimiento de su funcionamiento de su procesos mentales fue, como hemos indicado, una función del proyecto de inventar la máquina inteligente. Ello explica que en el famoso episodio antes citado por Feigenbaum, afirmara: "Al Newell and I invented a thinking machine", en lugar de: " I discovered how does my mind work". Su introspección psicoanalítica no era simplemente un ejercicio de psicología, sino de psicología aplicada a la invención de la máquina pensante. El fin era inventar The Logic Theorist Machine, el medio, resolver un problema de psicología.

Esta invención se presentó en los papeles dirigidos a la comunidad científica e ingeniera de su tiempo de una forma ambigua. Por un lado se presentó en un papel titulado " The Logic Theory Machine, a Complex Information Processing System", (1956, July 12) y a su vez, se decía en él que éste era el resultado de un

proyecto más amplio de estudio de los "complex information processing systems". Y por ello este programa se escindiría luego en dos : la ciencia cognitiva y la Artificial Intelligence "in the narrow sense" realizada también por ambos, pero sobre todo por el segundo, en el Computer Science Department.

2.2.6. EL 15 DE DICIEMBRE DE 1955.

Herb Simon afirma en el capítulo de su reciente autobiografía titulado ' The Logic Theorist is Conceived' : "I have always celebrated December 15, 1955, as the birthday of heuristic problem solving by computer, the moment when we knew how to demonstrate that a computer could use heuristic search methods to find solutions to difficult problems" (Models,1991:206).

Si como hemos visto al principio por L.T. se entiende sólo un proceso de simulación de pensamiento humano formalizado, se puede en efecto afirmar que el L.T. nace cuando Herb Simon escribe en la hoja de papel el proceso simulado de su mente de una forma programable. Pero eso no es lo que dice aquí: aquí dice que es el nacimiento de " "heuristic problem solving by computer". Aquí habla de una acción de resolución de problemas por el ordenador no sólo simulada sino real, de una solución real de problemas por el ordenador.

Eso no se había conseguido aún el 15 de diciembre de 1955.

El programa de ordenador aún no existía. Simon nos dice que le preguntó a Newell si era programable su esquema, y al estar éste de acuerdo, a continuación empezaron a trabajar en el segundo problema: escribir el procedimiento en una forma programable en un ordenador.

El avance que realiza Simon, el saber qué heurísticas puede seguir el ordenador para resolver el problema es sólo una tercera parte de la invención del L.T.. Si el trabajo se hubiera detenido aquí, el L.T. no existiría. Existiría la simulación de un procedimiento simulado de un proceso mental humano. Faltaban tres pasos más: 1. escribirlo en un lenguaje intermedio o "pseudo-código" programable y 2. programarlo y 3. probar que funcionaba. Lo que se había generado era un avance en la posibilidad plausible de construir un programa inteligente.

Esta diferencia entre programa simulado y programa que funciona nos parece importante, máxime en una cultura que se define cada vez más como una "sociedad del conocimiento" y cuya fuerza estriba en el valor añadido que produce en forma de conocimiento que pone en las cosas, por sus programas. La tecnología no es

una "posibilidad plausible" o "realizable" (feasible), es una posibilidad realizada, que funciona, que ya se ha realizado, y funciona. Por eso lo que importa son los resultados, no tanto los fines, sino los finales.

La cultura de la tecnología está pasando a ser cada vez menos una cultura de cosas, a una cultura del conocimiento de las cosas, de cómo se hacen éstas. El problema es que si este conocimiento no se prueba, y la única forma de probarlo es construyendo lo que se dice saber construir, no se tiene conocimiento tecnológico, sino su simulación, que en el caso de la cultura informática usualmente se queda en un procedimiento matemático formalizado, y de "computer science" volvemos a las matemáticas de nuevo.

La simulación del pensamiento humano puede ser un objetivo científico interesante, pero es una simulación, una metáfora, algo que se asemeja al pensamiento humano. Si se considera la metáfora o la simulación como la realidad en lugar de pasar de las logoculturas a las tecnoculturas, volveremos a las mitoculturas, a las culturas basadas en la imaginación y la creación de imágenes como modo cognitivo básico. Un ejemplo aclarará la diferencia.

Con un programa de CAD-CAM podemos simular en el ordenador un avión. Este diseño no es el avión, sino la idea de avión que se hace el diseñador con ayuda del ordenador para hacer un avión que vuele de verdad. El puede creer que al hacer este diseño ya conoce como se hace un avión, pero lo que conoce es cómo diseñar una idea de avión en el ordenador. Pero aún no ha construido el avión. Para realmente probar que su conocimiento de como hacer un avión funciona, este diseño ha de servir para construir el avión, no sólo su prototipo, su idea. De lo contrario, se queda en la simulación. Por ello nos parece oportuno diferenciar dos conceptos: conocimiento tecnológico simulado y conocimiento tecnológico probado.

La idea de simulación ha servido para liberar al ingeniero de limitarse a una mera actividad estrecha de "learning by doing". Antes, en época de Edison se inventaba por el método de "cut-and-try". Mera práctica, y de la práctica salía el "expertise". La revolución cognitiva tras la II Guerra Mundial cambió todo esto. Puso la fuerza en el conocimiento: "Knowledge is Power" (Feigenbaum). Pero este planteamiento vemos que tiene también sus problemas. El peligro es quedarse en Platón, en el reino de las ideas, independiente de la materia. Esta visión del conocimiento en parte ha supuesto un avance respecto al viejo pragmatismo ingeniero, pero al mismo tiempo, se ha retrocedido a un mundo cultural,

pretecnológico, al mundo de la ciencia, separando lo concebible de lo realizable, la investigación de su producción en forma de productos industriales, una de las actuales debilidades de Estados Unidos frente a Japón. El conocimiento es poder si y sólo si transforma la realidad, no sólo si la simula. La ciencia cambió la visión de la realidad. La tecnología hace algo más, cambia la realidad misma.

2.2.7. "THE INTERPRETATIVE CODE": ALLEN NEWELL.

El 15 de diciembre termina una etapa clave en la construcción del L.T.: descubrir como una mente humana, en concreto la de Herbert Simon, resolvía problemas de lógica. Pero esto no es "el nacimiento de la solución heurística de problemas por el ordenador". H. Simon no era un ordenador sino que su mente había "simulado" un supuesto funcionamiento de un ordenador. Faltaba una serie de pasos decisivos para que realmente un ordenador digital procesara la información contenida en el L.T. y probara los teoremas. Había que hacer algo más: inventarse un lenguaje nuevo para que el ordenador realizara efectivamente ese procedimiento simulado, había que programarlo en ese lenguaje-máquina, y había que comprobar que el programa funcionara realmente en un ordenador digital. Y así lo hicieron.

" During the subsequent several days, Al and I worked hard to sharpen up the program, and put it in a form where one could consider coding it for the machine (that is in the interpretative code). ((Simon. "Some Notes..."1957:2).

Este fue el segundo paso. Newell y Simon tradujeron los teoremas y las formas de resolverlos hechas por Simon en un "intermediate language" o "logic language" ("L.L.").¿ En qué consistía éste?. En papel presentado en Mayo 1, 1956, a la conferencia sobre Computers and Automation, en Washington, así lo explicaban: " This language is similar to a computer code in that it consists of sequences of instructions that the computer can execute. However, to execute these instructions, the computer uses a special program that translates the intermediate language instructions into machine code." (Newell & Simon, Currents...1956:19). Este lenguaje se denominaba usualmente "pseudo-codes" o "interpretive languages" y su principal objetivo era "to free the user from the additional planning and envisioning of information flows that is required if a problem is put in machine code" (ibid. :20).

La "Logic Theory Machine" que Newell y Simon presentaron en esta conferencia era

como ellos mismos afirman " a code in an interpretive language". La primera simulación de los procesos de resolución de los teoremas realizados a mano por Herb Simon se había traducido a una segunda simulación por Newell y Simon, que se acercaba aún más a la solución del problema inicial: cómo construir una máquina inteligente. Pero el programa de ordenador aún no existía, o si se prefiere aún no funcionaba en el ordenador y por lo tanto no se sabía realmente si iba a funcionar o no. Se está hablando aún de un programa sobre papel, en proceso de construcción. Parecía que podía funcionar, pero no se había probado. Para que L.T. se convirtiera en un programa de ordenador faltaba un tercer paso: programarlo informáticamente, y para ello Allen Newell y Cliff Shaw, paralelamente al trabajo de Herb Simon, se estaban encargando de construir un nuevo lenguaje para la máquina. Según escribe Simon en su autobiografía: "Although all three of us participated in its design, Al and Cliff took primary responsibility for constructing such an information-processing language (IPL) or list-processing language...My role consisted primarily of comparing proposed language with the analogous human functions." (Models,1991:204).

El documento clave para analizar el proceso de programación del LT es elaborado por Newell y Shaw:"Programming the logic theory machine" (1957), en que se especifica que las ideas del mismo están compartidas por el conjunto del equipo. En este papel se describe un nuevo lenguaje de programación, el IPL o "Information Processing List". La novedad fundamental que aportaba es que no era un lenguaje para el procesamiento de números como los concebidos hasta la fecha, como el FORTRAN, sino un nuevo tipo de lenguaje para procesar símbolos lógicos.

2.2.8. EL 9 DE AGOSTO DE 1956: LA INVENCION DE LA LOGIC THEORY MACHINE.

En resumen, de todo lo anterior se puede deducir que la invención del L.T. es un proceso que parte de una posibilidad imaginada de construir una máquina pensante, una secuencia de acciones para conseguir este objetivo, y un resultado final, probado.

En este proyecto inventivo, el descubrimiento , mediante autoanálisis, por Herbert Simon de cómo una mente humana resuelve un problema de lógica es un medio de un proyecto más general basado no en este descubrimiento sino en la invención del L.T. La ciencia empírica se puso como instrumento de la A.I.. Este complejo proceso de medios y fines explicaría la evolución y las características del modelo de AI de Herb Simon y Allen Newell en CMU. Si se

considera que el 15 de diciembre fue el nacimiento de la resolución heurística de problemas por el ordenador, ello explicaría dos profundas convicciones que se tienen: 1. Creer que el L.T. nació de una copia del funcionamiento de su mente, esto es, de un descubrimiento. Por ello se piensa que la clave del avance de la A.I. es la simulación del pensamiento humano, ésto es, el mismo procedimiento que él siguió. 2. Pensar que la actividad de simulación es el punto común denominador entre la ciencia cognitiva y la AI.

Pero esta versión dificulta entender la unidad del proceso de invención del LT, está fuera de un contexto concreto: el ordenador como máquina de circuitos electrónicos, y de las personas de carne y hueso que materializaron la Logic Theorist Machine.

Se identifica invención con actividad mental exclusivamente. Lo cual induce a errores. Pocos meses antes en Octubre de 1955, según el propio Simon relata en su biografía, también tuvo la convicción de que podía programar la máquina para resolver problemas de geometría: " The example I had in mind had to do with angles inscribed in circles and semicircles...Suddenly I had a clear conviction that we could program a machine to solve such problems. I jotted some notes on a piece of paper and thought hard about it for a few minutes, the conviction remaining very strong. I think the conviction arose from the fact that I could see the heuristic I was using and how it cut down the search space".(Models, 1991:203-204). Pocas semanas después el equipo tuvo que abandonar esta tarea. No podían representar los diagramas en el ordenador. La lógica matemática se convirtió en la nueva tarea.

Esto demuestra que no bastaba tan sólo con simular la idea, había que programarla.

Podía haber ocurrido lo mismo con la prueba simulada a mano del 15 de diciembre. Pero¿ cuál fue la diferencia?. Se pudo programarlo. Y se pudo probar, además, que el programa funcionaba.

Sin estas dos tareas no hubiera habido invento. La invención tecnológica era la síntesis de las tres funciones. Ninguna de ellas por separado podría haber producido la Logic Theory Machine. A la Conferencia de Darmouth Simon y Newell llevaron un código, todavía no tenían el programa probado. No obstante, ello ya fue suficiente para tener ventaja sobre el resto de participantes. Pero tomar esa conferencia como punto de partida de la A.I. es separar a ésta de la prueba de sus programas. A esa conferencia no fue Cliff Shaw que en ese momento estaba

programando el L.T. en el ordenador de la RAND en California. Pero el L.T. no nació en la Darmouth Conference de junio de 1956.

Es tradición en la cultura tecnológica americana decir que lo que importa son los resultados. Y según confirma el propio Simon, sólo el 9 de agosto de 1956 el Logic Theorist, programado en el ordenador JOHNNIAC en lenguaje IPL-II, produjo su primera prueba completa del Theorema 2.01 de los Principia de Russell y Whitehead. Ese fecha quedó probado que un ordenador podía pensar utilizando un programa basado en una simulación de la mente humana. Pero esa prueba no fue un conocimiento simulado, sino un conocimiento realizado.

Ello sólo pudo lograrse por el trabajo conjunto del equipo triangular de Herb Simon, Allen Newell Y Cliff Shaw.

La visión de la invención del L.T. que manifiesta Herb Simon concluye el 15 de diciembre, al ser concebida la idea. Ello puede explicar la escisión del programa inicial de investigación de la A.I. en CMU en dos: una parte en psicología cognitiva y otra en la A.I. de la SCS.

El dejar fuera del invento al programador, a la persona que en palabras de A. Newell era "the genuine computer scientist of the three", ha tenido repercusiones en el curso de la computer science en CMU. La ciencia puede limitarse al mundo de las ideas pues su tarea es explicar con ideas una realidad dada. No esta obligada a construir otra nueva. La tecnología, no. Incluso la ingeniería del conocimiento implica su manipulación y transformación.

La ciencia puede pensar el mundo como idea, como mente. El mayor avance científico en las pasadas décadas ha sido el nacimiento de la ciencia cognitiva. Como reconoce su principal historiador Howard Gardner (1985), dicha ciencia tiene el programa de investigación establecido por Platon hace más de 2000 años en el Menón a cerca de cuál es la naturaleza del conocimiento. Y cómo Platon piensa que conocimiento son ideas.

Pero en la computer science, y en particular en la A.I., esta escisión ideas-materia está siendo sustituida por una entidad, el programa informático, que es un producto de conocimiento distinto al mundo de las ideas platónico. Newell, Shaw y Simon no hablan sólo de programa, sino de Logic Theory Machine, en la mayoría de los textos que escribieron explicando su invento.

2.2.9. ¿"QUIEN" ES Y QUE HACE "LOGIC THEORY MACHINE"?

L.T. es un programa de ordenador que prueba teoremas de lógica matemática.

Se considera el primer programa informático inventado por el ser humano que soluciona problemas no numéricos mediante la técnica de "heuristic search" o búsqueda selectiva. Sus autores: Allen Newell, Cliff Shaw y Herb Simon. Lugar: Carnegie Institute of Technology y RAND Corporation.

H. Simon y A. Newell en Human Problem Solving (1972) nos exponen un caso que ejemplifica el procedimiento seguido por los autores para que la máquina consiguiera realizar la prueba de un teorema : " We supply LT the list of five axioms, instructing it that these are theorems it can assume to be true...LT already has stored the rules of inference and the definitions -the processes that substitute, replace, detach, and chain. Next we give LT a single expression, say Theorem 2.01, and ask L.T. fo find a proof for it. LT works for about ten seconds, then prints out the following proof:

SOLUTION PATH SOLUTION ACTION SEQUENCE

$(P \supset \sim P) \supset \sim P$ (THEOREM 2.01, TO BE PROVED)

1. $(A \vee A) \supset A$ P1. (copy Axiom 1.2.)
2. $(\sim A \vee \sim A) \supset \sim A$ P2. (subs. $\sim A$ for A in 1).
3. $(A \supset \sim A) \supset \sim A$ P3. (replace \vee with \supset in 2).
4. $(P \supset \sim P) \supset \sim P$ P4. (subs. P for A in 3; QED).

" (1972: 110).

Esta fue la primera prueba construída por el programa LT en el ordenador JOHNNIAC el 9 de agosto de 1955.

Los cinco axiomas antes citados los tomó Herb Simon de los Principia Mathematica de Whitehead y Russell. Los símbolos que aparecen en el teorema (A, P, \dots) se denominan variables y están unidos entre sí mediante conectivas (\sim, \supset, \dots) cuyo significado es fácil de entender mediante simple traducción de la anterior prueba escrita en lenguaje de lógica simbólica a lenguaje "natural" :

CAMINO DE LA SOLUCION. SECUENCIA.

$(P \text{ implica no-}P) \text{ implica no-}P$ (TEOREMA 2.01)

1. $(A \text{ o } A) \text{ implica } A$ 1. Copiar el Axioma 1.2.
2. $(\text{no-}A \text{ o no-}A) \text{ implica no-}A$ 2.Sustituir no- A por A en 1.

3. (A implica no-A) implica no-A 3.Reemp. o con implica en 2.

4. (P implica no-P) implica no-P. 4.Sust. P por A en 3.

La parte de la izquierda indica el teorema probado y la de la derecha las heurísticas seguidas para probarlo.

¿Cómo consiguieron el equipo de Newell-Shaw-Simon que Logic Theorist probara teoremas matemáticos?. Una vez seleccionado el campo de la prueba de teoremas, al ser más fácil que otros que exploraron como los de geometría o el ajedrez, tuvieron que construir un problema: "a problem".

Según su teoría del HUMAN PROBLEM SOLVING: " A person is confronted with a problem when he wants something and does not know inmediately what series of actions he can perform to get it. The desired object may be very tangible (an apple to eat) or abstract (an elegant proof for a theorem). It may be specific (that particular apple over there) or a quite general (something to appease hunger). It may be a physical object (an apple) or a set of symbols (the proof of a theorem). The actions involved in obtaining desired objects include physical actions (walking, reaching, writing),perceptual activities (looking, listening,) and purely mental activities (judging the similarity of two symbols, remembering a scene,and so on)" (1972:72).

El "problema" según esta definición lo construye una persona que quiere algo, a "desired object". Empieza pues por un "want" ("querer"). Esta persona no sabe como conseguir lo que se propone. La solución consistirá en saber cuál es la secuencia de acciones a seguir para transformar su estado inicial en su "desired object". Este "goal, problem or task" se genera en un medio determinado "environment". La unión de ambos forma un "task environment" (Newell &Simon:1972:55). Y para Simon y Newell, "it is the task that defines a point of view about an environment, and that, in fact, allows an environment to be delimited" (Ibid: 55).

2.2.10. "OPERADORES BINARIOS" Y LAS TRES DIMENSIONES DEL PROBLEM SOLVING.

Levi Strauss, en el Hombre Desnudo,(1987, 1a ed. 1971) después de un exhaustivo estudio de los mitos de los indios norteamericanos, afirma que el pensamiento mítico nace, como todo pensamiento, inicialmente como una oposición entre el yo y el otro. "A fin de cuentas, para todo sistema mitológico no hay más que una secuencia absolutamente indecible. Llevada a sus contornos esenciales merced a

transformaciones sucesivas, se reduce al enunciado de una oposición o, más exactamente, al enunciado de la oposición como primero de todos los datos...está el cielo, está la tierra" (ibid.:543).

Los operadores binarios son mecanismos del pensamiento mítico que producen una continua división de todas las categorías que forman los mitos en una oposición binaria: muerte-vida, noche-día, hombre-mujer,... Según Levi-Strauss son "las piezas elementales de esa vasta máquina combinatoria que es todo sistema mítico" (ibid.:505). La estructura profunda del pensamiento mitológico.

Basándose en el modelo de la lingüística estructural de Saussure y Jakobson, las *Mythologiques* constituyen un enciclopédico esfuerzo por explicar los mitos desde este modelo de pensamiento binario. Para Levi-Strauss: " El problema de la génesis del mito se confunde pues, con la del pensamiento humano...Las condiciones de aparición del mito son así, las mismas que las de todo pensamiento, puesto que éste no podría ser sino pensamiento de un objeto, y un objeto no es tal...más que por el hecho de constituir al sujeto como sujeto, y a la consciencia misma como consciencia de una relación" (ibid:544).

Pero el modelo de pensamiento que vemos en esta cultura tecnológica, a partir del la obra *Human Problem Solving* (1972) introduce un cambio significativo. Lo que vemos en el "problem solving" es un intento de resolución intelectual del problema planteado de forma binaria, entre lo que el sujeto desea y la realidad externa a él. Puede considerarse un intento de superación de esta dicotomía que plantea al individuo: la oposición de sus deseos o "estado final" con la realidad o "estado inicial". Lo nuevo de este pensamiento es que traza un programa, una secuencia ordenada de acciones para transformar, cambiar su estado inicial en su estado preferido. No sólo analiza uno u otro estado sino que diseña una vía de cambio de la situación en dirección hacia su estado deseado. Por eso se constituye en pensamiento tridimensional, no binario.

Curiosamente éste es también un modelo nacido en una cultura americana, a partir de determinados grupos de investigadores de Inteligencia Artificial. Curiosamente no de toda la computer science. El funcionamiento binario se ha querido asociar, y en efecto lo está, con el de los ordenadores, y con su códigos binarios de 0 y 1. Pero el LT es un programa particular elaborado bajo un modelo de problem solving complejo, de comportamiento simbólico no sólo numérico-matemático.

Podría ser la base de una teoría de conocimiento no binaria sino, a mi parecer,

tridimensional. Un sistema de conocimiento nacido de la obra de construcción de un invento técnico: un programa de ordenador inteligente.

Lo nuevo que aporta el paradigma del Human Problem Solving es la posibilidad de que el ser humano, partiendo de esta escisión inicial, encuentre una vía de resolución de la escisión binaria, no para volver a la unidad original, sino para formar estructuras de realidad más complejas.

Levi-Strauss indica que en todas las mitologías, a pesar de funcionar por medio de oposiciones binarias, permanece un misterio inexplicable, sin embargo: ¿cómo, si existe tal separación cielo-tierra, vida-muerte, existe en la tierra esa cosa celeste que es el fuego?. " Está el cielo, está la tierra; entre ellos no podría concebirse paridad; por consiguiente la presencia en tierra de esa cosa celeste que es el fuego constituye un misterio; en resumidas cuentas, desde el punto y hora en que el fuego del cielo se encuentra ahora aquí a título de hogar doméstico, por fuerza hubo que ir por él de la tierra al cielo" (ibid. : 543-44). Levi Strauss, la conquista del fuego marca el tránsito del estado de naturaleza al de sociedad, iniciando la evolución cultural. La cultura humana nace como tecnocultura: " Este acontecimiento decisivo para la vida y el porvenir de la humanidad no es otro -ya lo sabemos- que la conquista del fuego en el cielo por un héroe terrestre que se aventuró por allá voluntariamente o a pesar suyo..." (ibid.:561). Como señala E. Benveniste (1969), citado por Levi-Strauss, la noción de dios ("deiwos") en la comunidad indoeuropea está asociada con lo "luminoso" y "celeste", lo opuesto a lo humano, lo "terrestre"(eso significa homo en latín).

Pero la conquista del fuego es un invento humano, entendible desde un pensamiento de "problem solving". Este misterio, y por tanto una clave para entender el sistema cultural humano, puede ser explicado desde una análisis de las tecnoculturas. El que hasta ahora la cultura humana se haya visto como una mito o una logo cultura quizá se haya debido al escado desarrollo e independencia del pensamiento propio del inventor. Pero, al empezar a generalizarse la cultura tecnológica, los citados misterios quedan aclarados. Pero ello es a costa de una crisis en la definición de lo que entendemos por ser humano, al menos en Occidente.

Herb Simon afirma que al inventar un programa de ordenador capaz de pensar no-numéricamente, "nosotros resolvimos el venerable problema mente/cuerpo, explicando como un sistema compuesto de materia puede tener las propiedades de

la mente" (1991: 190). Pero ello plantea un problema aún mayor: la crisis de la propia definición del hombre propia de Aristóteles, y del sistema de conocimiento que lo cimenta.

El ser humano ha sido definido desde hace 2000 años como "animal racional", teniendo la Lógica por obra cumbre en cuanto a sistema de conocimiento, y la Política, en cuanto sistema de organización del "zoon politikon". Detengámonos en el primer aspecto.

Aristóteles empieza afirmando en *De la Interpretación*: "No toda oración es un enunciado; sólo son enunciados las que tienen en ellas verdad o falsedad. Así, una jaculatoria es una oración, pero no es ni verdadera ni falsa. Desechemos todos los tipos de oración excepto el enunciado pues es este último el que importa para la presente investigación" (cit. Daniel Quesada. 1985:8).

Los enunciados se rigen por "el principio de contradicción": las proposiciones han de ser verdaderas o falsas. Esta doctrina fundamenta el principio del "tertium non datur" o tertium exclusum. Un pensamiento ha de ser verdadero o falso. No puede ser ambas cosas al mismo tiempo. O una tercer opción. Este tipo de pensamiento nace de la ruptura de la unidad inicial del pensamiento mítico, basado en la idea de participación. Levi-Bruhl explica que en todos los pueblos primitivos encontramos que todo se explica por metáforas y analogías, por símiles, dado que se considera que existe una realidad común a todos los seres, en la que todos formamos parte. "Mana", "wakan",... Se trata de un pensamiento unitario, holista. ("holly": sagrado, en inglés).

Levi-Strauss en su reciente libro "De pres et de loin" (1990) reconoce que: "Une des differences essentielles entre la maniere dont nous réfléchissons et la maniere dont ces peuples réfléchissent, c'est notre besoin de morceler. Nous avons appris cela de Descartes: diviser la difficulté en autant de parcelles qu'il est requis pour la mieux résoudre. La pensée des peuples dits primitifs refuse ce morcellement. Une explication ne vaut qu'à condition d'être totale...Aussi chaque expression de la vie collective constitue ce que Mauss appelait un fait social total." (1990:157-158).

Occidente con su pensamiento binario quebró en su día esta "com-uni3n" del hombre con la naturaleza. Permitió un análisis de la naturaleza independiente de los deseos o voluntades humanas. La L3gica de Arist3teles consagra la visi3n binaria propia de Occidente, basada en la oposici3n verdadero falso.

La Logic Theory Machine utiliza elementos de esta l3gica pero en un sentido

distinto, como instrumentos al servicio de un sistema tecnológico, que para su construcción precisa de un modo de pensamiento no binario sino tridimensional, más complejo que el de Aristóteles.

2.2.11. COMO RESUELVE PROBLEMAS LA "LOGIC THEORY MACHINE"?

El equipo en cuestión se planteó el problema en un contexto determinado.

Según la definición anterior, los problemas partieron de sus "wants". Los "wants" de Simon-Newell-Shaw, sus "desires", dentro de un medio determinado, Carnegie Tech y la RAND, y en un momento histórico dado, los Estados Unidos de 1955. Y adaptándolos al medio con el que tenían que trabajar, el ordenador, construyeron el siguiente problema a Logic Theorist: "If we take as the desired object the theorem to be proved, then the condition for a problem solution is that this object be obtained by a sequence of admissible operations on the axioms and previously proved theorems taken as the initial objects". (1972: 112).

Ellos eligieron el objetivo final a resolver : teoremas de lógica simbólica a probar, y cuáles de ellos. Seleccionaron el estado inicial del que partir: los axiomas de Russel y Whitehead. Y una vez construido el problema le proporcionaron a la máquina la vía de cómo resolverlo : el " heuristic search method".

Este método se basaba en cuatro procesos: 1. seleccionar un elemento. 2. Seleccionar un operador. 3. Evaluar el nuevo elemento y 4. Decidir el siguiente paso. (1972: 103).

Los operadores o "heurísticas" o "methods" eran operaciones razonables y limitadas que en el caso de funcionar contribuían al descubrimiento de la prueba.

Estas heurísticas eran los siguientes: 1. "Sustitution", un método que buscaba una prueba encontrando un axioma o un teorema previo probado que pudiera ser transformado mediante una serie de sustituciones de variables y cambios de conectivas en la expresión del problema. 2. "Detachment", un método que intentaba separar del problema un nuevo subproblema que , en caso de resolverse, proveería una prueba para el problema. 3. "Chaining", dos métodos que usan la relación de implicación para crear un nuevo subproblema que ayuda a encontrar la prueba. Al ser la relación transitiva, tanto puede operar hacia adelante ("forward chaining") como hacia atrás ("chaining backward").

Tanto los métodos de "substitution" y "detachment" son reglas clásicas de inferencia.

Una vez establecido el problema y el método de resolución del mismo decidieron que procedimiento seguir para aplicar estos métodos. La elección adoptada fue el que la máquina trabajara hacia atrás ("backwards"), ésto es, el elemento inicial en el espacio de búsqueda sería el teorema a probar, y la búsqueda procedería hacia atrás en dirección de los axiomas y teoremas supuestos.

En resumen, el "problem solving" tal como lo concebían Newell y Simon consistía, en un sentido amplio, en la generación de un conjunto de posibilidades y dentro de ellas mediante una labor de selección y test, delimitar aquellas que encajaban con los objetivos deseados.

2.2.12. EL PENSAMIENTO DE LA "LOGIC THEORY MACHINE" : 38 TEOREMAS PROBADOS.

Los autores como tarea inicial le proporcionaron a la máquina 52 teoremas y un conjunto de especificaciones a seguir. Y la máquina logró probar 38 (73%) de los 52 teoremas. En un caso, el de la proposición 2.85, la máquina encontró una prueba más sencilla incluso que la hallada por los propios B. Russell y A.

Whithead en su libro. Cuando Herb Simon escribió a principios de septiembre de 1957 a B. Russell dándole a conocer orgulloso este hecho, al final de la carta añadía : "I am not sure that these facts should be made known to schoolboys" (Models, 1991:209). La respuesta del lógico británico fue : "I am delighted by your example of the superiority of your machine to Whitehead and me. I quite appreciate your reasons for thinking that the facts should be concealed from schoolboys. How can one expect them to learn to do sums when they know that machines can do them better?. I am also delighted by your exact demonstration of the old saw that wisdom is not the same thing as erudition". (Carta personal 21 Sept, 1957. Models: 209).

El hecho nuevo y muy significativo culturalmente es que una máquina inventada por un científico norteamericano había sido capaz de "create a beautifully simple proof to replace a far more complex one", en palabras de Simon, que un lógico inglés, uno de los fundadores de la moderna lógica simbólica.

Los teoremas resueltos por L.T. los resolvió la máquina no sus inventores. Igual que los problemas escolares de los niños los resuelven éstos aunque sus padres les hayan engendrado y los maestros los esten instruyendo. No simulan la

inteligencia de los padres, tienen una real inteligencia propia. Esto plantea nuevos problemas a la antropología. Ya no somos la única especie que procesa símbolos. Ni nos podemos definir por esta función. Sin embargo las ciencias sociales aún creen en el animal simbólico, como rasgo único del sapiens. Las preguntas claves de una tecnocultura, las planteó más acertadamente Antonio Gramsci y su "filosofía de la praxis": " Pensando un poco se ve el hacernos la pregunta de que es el hombre queremos decir: ' que puede llegar a ser el hombre?, es decir, si el hombre puede llegar a dominar su propio destino, puede 'hacerse', puede crearse una propia vida. Decimos, pues, que el hombre es un proceso y , concretamente, el proceso de sus actos". (1978:48). Igualmente, la pregunta no es ya ¿qué es el conocimiento?, sino que nuevos tipos de conocimiento pueden generarse en las tecnoculturas?, ¿cuales apuntan ya germinalmente?, y ¿ cómo construirlos activamente?.

2.2.13. LOGICA E INTELIGENCIA ARTIFICIAL: DOS CULTURAS ACADEMICAS ASIMETRICAS.

Este diálogo entre B.Russell y Herber Simon que parece anecdótico puede significar algo más profundo: el nacimiento de una nueva cultura académica asimétrica en el terreno del conocimiento. Esta expresión la sugirió Simon en la primera entrevista que tuvimos el 1 de mayo de 1990 en CMU. Pero el sentido que le doy no es el mismo que Simon apuntó.

El se referí al clásico esquema de C.P. Snow y la oposición entre la cultura científica y la humanística. Decía que los primeros solían conocer la segundo pero no ocurría tanto al contrario, lo cual hacía difícil la comunicación.

Ello se ve claramente en el tipo de lenguaje utilizado. Mientras que los de la cultura científica pueden expresarse en dos tipos de lenguaje: el simbólico tradicional y el lenguaje matemático o lógico-formal, los de la cultura humanística sólo conocen el primero, normalmente.

Pero ahora, está surgiendo una nueva cultura, que guarda con la científica, una relación parecida. Los lenguajes de programación de la informática utilizan la lógica formal, como el científico utiliza en sus artículos el lenguaje natural, pero no forman parte sustancial de la misma. Simon afirma que: "El científico es un problem solver". Que no es lo mismo que decir: el problem solver se reduce al científico. La ciencia es un subconjunto, en un campo mucho más amplio de conocimiento denominado "information processing" y que tiene el "problem

solving" como metodología. En palabras de Herb Simon: "The processes of discovery are just applications of the processes of problem solving".(1989: 375).

Newell y Simon utilizaron los Principia Mathematica como instrumento para construir un programa informático inteligente. Utilizaron lo que para Russell y Whitehead eran los fundamentos de la matemática como una técnica útil.

Su utilización de la lógica fue instrumental no final. Y no instrumental para hacer sólo ciencia empírica como los físicos, sino para construir una disciplina A.I., según ellos no basada en la lógica sino en el "heuristic search".

La relación asimétrica se establecía por el hecho de que Herb Simon conocía la cultura científica y al mismo tiempo la tecnológica, lo cual no era el caso de B. Russell para quien las máquinas eran "horribles y odiosas porque imponen esclavitud" (Boorstein,: 90).

Como dato curioso de esta subvaloración del lugar de la tecnología y del ingeniero en la cultura académica inglesa, en 1973, el Science Research Council del gobierno británico, organismo encargado de financiar la ciencia en ese país, pidió un informe sobre el estado de la Inteligencia Artificial al famoso matemático aplicado Sir James Lighthill, que publicó un informe declarando que el trabajo era insuficiente y en su peor aspecto bordeaba la charlatanería. Este informe retrasó en varios años la financiación de la AI en el país cuna de la moderna ciencia y de Alan Turing.

Simon acaba afirmando: " In computer programming languages, I found tools that classical mathematical languages had not provided for exploring the processes of human thinking and for attaining accuracy and rigor in the behavioral and social sciences" (Models, :214).

2.2.14. "HUMAN PROBLEM SOLVING" : EL ENFOQUE DE CMU EN ARTIFICIAL INTELLIGENCE.

El aspecto nuevo de la teoría del "Human Problem Solving", codificada en 1972 es que es una descripción del equipo de human problem solvers que inventaron el LT. Hasta entonces, el "problem solving" eran técnicas, métodos de resolver problemas. Con este equipo, lo que era una técnica se convirtió en una teoría de "tipo humano" y de "tipo de máquina". Y en esta teorización jugó un papel clave que hubiera un científico de organizaciones en el equipo.

La Constitución de los Estados Unidos empieza con esta frase: "We the People of the United States", pretendiendo indicar que es el "people" el autor de su

propio programa cultural, de las reglas del juego social. El Logic Theorist fue un invento de un equipo de tres personas: Allen Newell, Cliff Shaw y Herbert Simon. Equipo que firma los textos conjuntos que entre 1957 y 1961 presentaron su trabajo a la comunidad científica, y lo firman en estricto orden alfabético, pretendiendo aparecer iguales en su contribución.

Este equipo son los autores del invento, y de su teorización. Para hacer un programa informático inteligente, y una teoría del mismo tres individuos con intereses y especialidades distintas tuvieron que formar un equipo humano con unas características particulares. Construyeron un artificio colectivo nuevo: un "task force", o equipo con una tarea, formado por un científico político (Simon), un matemático aplicado (Newell) y un programador (Shaw), una combinación particular de elementos teóricos y prácticos.

Como explica Herb Simon en una carta personal a Pamela McCorduck el 12 de agosto de 1975: "It was precisely because there were three of us, because each of us brought different knowledge and experience and different viewpoints to the project, and because we found a mode of operation that allowed us to take advantage of that variety of experience while working intimately within a common framework--it was precisely for these reasons that we were able to make such rapid progress and to solve the two basic problems (a) of representing heuristics for problem solving research, and (b) of designing programming formalisms that allowed those heuristics to be implemented. I do not believe that what he did was simply to pour into a common pot three separate, identifiable ingredients, each contributed by one of us, or that either of the two problems was solved by a subset of us". (Carta a P. McCorduck. 1975).

En el plano de las ideas, este equipo formó su propia teoría de cómo hacer una máquina inteligente, separándose en esto de la corriente mayoritaria en el campo, que desde Von Newman y N. Wiener, se basaba o en la teoría de los autómatas o en las redes neuronales.

En los medios de A.I. se considera la histórica Conferencia de Darmouth de junio de 1956 como el inicio oficial de la Artificial Intelligence en Estados Unidos.

En esto ha influido el libro de Pamela McCordurk, "Machines who think", (1979) la versión más completa, hasta la fecha, sobre la historia de la Artificial Intelligence en Estados Unidos. Esta reunión fue convocada por John McCarthy entonces en Darmouth, Marvin Minsky del MIT, Claude Shannon de la Bell Telephone y Nat Rochester de IBM. En total asistieron 10 personas.

Se presentaron muchas ideas de programas para resolver problemas, reconocer modelos, y realizar juegos, pero los únicos esquemas concretos de AI que se presentaron en ella fueron el Logic Theorist de Newell-Shaw-Simon y un programa para probar teoremas en cálculo proposicional por el método de deducción natural escrito por Trenchard Moore.

Dos meses después de la Conferencia, el 9 de Agosto de 1955, el L.T. programado en lenguaje IPL-II en un ordenador JOHNNIAC, produjo su primera prueba completa de un teorema de lógica simbólica. Con ello, el equipo Newell-Shaw-Simon resolvía el problema planteado por A. Turing en 1950, "can machines think?", en su famoso texto "Computing Machinery and Intelligence". Estos científicos probaron que podían y este hecho, este invento, dió nacimiento a la moderna Inteligencia Artificial como nueva disciplina probada.

¿Qué les dió la ventaja a este equipo sobre los restantes?. ¿ Por qué un equipo "marginal" en los ambientes informáticos se adelantaron a los científicos que trabajaban desde hacia años sobre el tema?.

Shannon era una reconocida autoridad ya en 1956, al haber creado la teoría estadística de la información, basada en una síntesis de matemáticas e ingeniería electrónica. En esa época trabajaba con John McCarthy, matemático, en un modelo de AI basado en la teoría de los autómatas, en la línea abierta por Von Neumann y Norbert Wiener, el fundador de la cibernética. M Minsky, también matemático, trabajaba en el MIT tratando de conseguir Inteligencia Artificial basándose en el modelo de redes neuronales de la fisiología, modelo expuesto por el neurofisiólogo Warren McCulloch.

Cada uno definía la Inteligencia Artificial desde sus conocimientos. Para los ingenieros-matemáticos, la A.I. era un autómata lógico, para los matemáticos-neurofisiólogos, una red neuronal.

Pamela McCorduck considera el "turning point" de la Inteligencia Artificial esa famosa Conferencia. Como antropólogos, nos interesa más el conocer cuál es la versión que de esta Conferencia tiene el propio Herb Simon, versión no publicada en la citada historia y aún a la espera de su publicación. Ello nos puede guiar en la pregunta antes formulada.

En una carta personal dirigida a Pamela McCorduck, el 27 de Septiembre de 1977, H. Simon explica así su versión: " There was the emergence of the cybernetic Zeitgeist...It flowered in the early 50's, then came to a kind of stalemate ...because of its preoccupation with physiology. That's where things stood at

the beginning of the Darmouth Conference, which John pulled together with the hope of becoming the organizer and leader of a team that would generate the new research direction...Except that it turned out that there were these other two guys, Newell and Simon, who, marching to a different drum, had already defined the new direction and had demonstrated its rightness with an actual running computer program, LT, that was an artificial intelligence. A great victory for the field but a shattering blow to the hopes for Darmouth. It is probably neither accidental nor dramatically uninteresting that the two gate-crashers came from outside the main cybernetic stream (though certainly not outside or unaware of the cybernetic Zeitgeist), and that they were responding partly to cues (heuristic, cognitive psychology) to which the others hadn't been exposed. A classic piece of science history, and a very different one from the present draft of the Darmouth chapter. More dramatic and less anti-climatic too...

Darmouth was not the successful launching of the new search; it was the moment of recognition that the old search had been superseded by an impulse coming from without... LT and GPS dominated the entire scene from 1956 to 1963. Gelernter and Slagle take these programs, not Darmouth or pre-Darmouth, as their starting point...

There was a major paradigm shift in 1956, it was produced by LT, a running program, and it made the Darmouth Conference rather irrelevant except as the channel through which it was announced to the then AI world." (Los subrayados son de H. Simon).

El equipo de Newell-Simon-Shaw ofrecieron un paradigma nuevo, distinto al anterior. Construyeron una inteligencia artificial inspirándose en cómo un humano resolvía un determinada tarea, acción que más tarde fue teorizada como "human problem solving". Se consideraba al ser humano como un "problem solver", y en concreto, un problem solver que usaba un particular método: la "heuristic search", o búsqueda heurística. Este paradigma, como se verá más tarde, no es compartido por toda la Inteligencia Artificial, ni por la psicología cognitiva. Este enfoque no partía del campo tradicional de la matemática o la ingeniería y su teoría de los autómatas, ni tampoco del campo tradicional de la psicología, dominado entonces por el conductismo. El nuevo enfoque provenía de autores minoritarios en psicología como Tolman y de la ciencia política y de los estudios sobre "decision making". Herbert Simon era ante todo un científico político, y sus estudios desde 1949 se habían centrado en el "Administrative Behavior".

Allen Newell conocía también al matemático aplicado Polya (: How to Solve It") y a autores provenientes de la psicología (Dunker, Wertheimer).

El "human problem solving" sintetizó estos distintos enfoques del problema.

2.2.16. " THE CARNEGIE-RAND GROUP".

Esta teoría fue codificada por primera vez en el texto de 1957, "Elements of a Theory of Human Problem Solving" firmado por los tres y realizado más de un año después de inventado el Logic Theorist.

La novedad que aporta la teoría del HUMAN PROBLEM SOLVING reside no en explicar el hombre desde la máquina, o al menos sólo desde la máquina, como es lo propio desde la Mettrie (1747), L'Homme Machine, (cuyo eco es la definición de hombre de Marvin Minsky , the Meat Machine), sino considerar la máquina como un "information processing sistem" capaz de simular tareas hasta entonces propia de los seres humanos. Esto es, la máquina imitaba al ser humano. Y no sus estructuras neuronales sino sus procesos considerados superiores: sus procesos mentales inteligentes. Esto fue una revolución: en el campo de la computer science y en el de las humanidades.

Esto puso por delante al equipo de Carnegie Tech-RAND de los trabajos más avanzados que en ese momento se hacían en las dos tradicionales culturas académicas en Estados Unidos: la ingeniera con el MIT y Stanford a la cabeza , y la de "liberal-arts universities" encabezada por Harvard, donde un equipo de psicólogos dirigido por George A. Miller sólo cinco años después de Dartmouth publicaría su obra Plans and the Structure of Behavior (1960), que marcó el cambio de paradigma del conductismo al cognitivismo en psicología en Estados Unidos. La llamada "revolución cognitiva" estaba en marcha.

Logic Theorist fue construido sobre la base de esta síntesis de conocimientos y de personas.

Esta síntesis es a mi entender un elemento clave que permite entender qué carácter tiene la A.I. en CMU, y el por qué del interés de sus líderes como Newell y Simon de basar su programa de investigación en dos instituciones simultáneamente, la School of Computer Science y el College of Humanities and Social Sciences de CMU. Esta combinación de psicología y computer science les dió la superioridad en 1955.

¿Qué se conseguía con todo ello ?. Tres objetivos: 1. fundar una nueva disciplina, la AI, 2. abrir un nuevo paradigma en una ciencia ya antigua, la psicología, y 3. conseguir validar ambas con una prueba. Realizaron un invento

técnico, una máquina inteligente, y además una teoría del mismo, una disciplina de invención de máquinas inteligentes. Y algo más : una nueva teoría de la inteligencia humana, entendida como problem solving, unificada con la inteligencia del ordenador en forma de "information processing systems". Una teoría que cambió el paradigma de la vieja ciencia de la psicología, pasando del conductismo a paradigma cognitivo, y en concreto, a una forma particular del mismo: el que se basa en la construcción de programas de ordenador para simular procesos cognitivos humanos.

Pero también señala los límites de este enfoque. Al pretender imitar al ser humano se ha cogido un "model of man" racional, inteligente, lógico. Pero el problema es que al inventar el Logic Theorist el científico (Simon) y el matemático (Newell) se transformaron a sí mismos en tipos humanos distintos. Para inventar una máquina lógica había que ser algo más que lógicos, había que ser inventores de máquinas lógicas. Y esta reflexión aún no se ha iniciado. Al considerar al animal racional el modelo del L.T. finalmente matemáticos y lógicos han desplazado a Simon y Newell en la dirección del proceso que ellos iniciaron. El modelo humano de la A.I. son sus inventores, y estos no son meros animales de "bounded rationality".

2.2.15. LA PRESENTACION DE LA "LOGIC THEORY MACHINE" A LA COMUNIDAD CIENTIFICA E INGENIERA.

Herb Simon considera 1955 y 1956 como los años más importantes de su vida como científico, y dice que son las invenciones que realizaron las que los consagraron como los "parents of artificial intelligence". Sin embargo, estos inventos según los textos donde se los presentó a la comunidad científica e ingeniera, se consideraban por sus autores como simples medios de un programa más amplio de investigación. Estos textos académicos se presentaron después de que Simon y Newell ya habían construido el programa informático.

En la presentación oficial del LT ante el Professional Group of Information Theory of the Institute of Radio Engineers, en Septiembre 10 de 1956 afirmaban: "The logic theory machine is part of a program of research to understand complex information processing systems by specifying and synthesizing a substantial variety of such systems for empirical study". ("The Logic Theory Machine: A Complex Information Processing System").

Para conseguir este fin propuesto: " One tactic for exploring the domain of

complex systems is to synthesize some and study their structure and behavior empirically". (:3).

Este objetivo aparece en textos firmados por todo el equipo de tres personas.

Así, ante la Joint Computer Conference realizada en 1957 se vuelve a explicitar en el texto "Empirical Explorations with the Logic Theory Machine: A Case Study in Heuristics" firmado por todo el equipo que : "The research reported here is aimed at understanding the complex processes (heuristics) that are effective in problem-solving...We wish to understand how a mathematician, for example, is able to prove a theorem even though he does not know when he starts how, or if, he is going to succeed...This focuses on the pure theory of problem-research solving...Our research strategy in studying complex systems is to specify them in detail, program them for digital computers, and study their behavior empirically by running them with a number of variations and under a variety of conditions". (1963:109-110).

¿Qué teoría nueva aportó este equipo al estudio de los "complex information systems": la teoría del "Human Problem Solving".

Para ello se basaron, según escribieron en 1972, (Human Problem Solving. Addendum) fundamentalmente en teorías de psicólogos gestaltistas (Duncker ,1945, Wertheimer,1945)que también habían trabajado este tema del "problem solving" en los seres humanos y también en los trabajos del matemático Polya que ya había escrito en 1945 un libro titulado "How to solve it", en el que basándose en los griegos, explicaba diferentes heurísticas para resolver problemas de matemáticas.

Así el 4 de marzo de 1957, un año después de construido el Logic Theorist, en "Elements of a Theory of Human Problem Solving" afirmaban: "Our position is that the appropriate way to describe a piece of problem-solving behavior is in term of a program--a specification of what the organism will (ilegible) under varying environmental circumstances in terms of certain elementary information processes it is capable of performing. This assertion has nothing to do--directly-- with computers. Such programas could be written...if computers had never existed. A program is no more and no less an analogy to the behavior of an organism than is a differential equation to the behavior of the electrical circuit it describes Digital computers come into the picture only because they can, by appropriate programming, be induced to execute the same sequence of information processes that humans execute when they are solving problems. " (1957:5-6).

El objetivo de simulación de la mente humana mediante el uso del ordenador fue el objetivo fundamental de Herb Simon en este proyecto. No obstante y de forma paradójica su contribución fundamental al mismo tuvo que empezar al revés : simulando el funcionamiento de un programa de ordenador en su propia mente, como el mismo reconoce. La razón es que ese programa de ordenador para simular procesos cognitivos humanos aún no existía. Había que inventarlo primero. No se podía hacer exploraciones empíricas sin máquina. Y el programa informático empezó al analizar Herbert Simon su propio comportamiento de programarlo. Los inventores de esta máquina se inspiraron no tanto en como piensa "la mente humana" universalmente sino, en como pensaban ellos cuando estaban resolviendo el problema de cómo inventar una inteligencia artificial: "By using a human (myself) to simulate the machine". (Carte de H. Simon a A. de Groot, January 3, 1956. Models 1991:206). A continuación escribieron estos procedimientos en forma de código, luego en forma de lenguaje informático y se los introdujeron al ordenador.

Este conjunto complejo de actividades organizadas en que se mezclaron: 1. un doble interés de varias personas, tanto en simular procesos cognitivos humanos en un ordenador, como conseguir una máquina que realizara procesos similares a la mente humana. 2. un trabajo a continuación se selección de entre los conocimientos de todas estas personas provenientes de las ciencias sociales, la lógica matemática y la ingeniería de aquellos que podían ser útiles para el el fin previsto, y al tiempo la construcción de una teoría de human problem solving y 3. por último un trabajo práctico de puesta en marcha del invento mediante una continuada y ardua experimentación fue lo que dio al equipo Simon-Newell-Shaw una clara ventaja sobre otros intentos de construir AI a mediados de los 50 en Estados Unidos, que se limitaron a síntesis de conocimientos más elementales, quedándose en la mera formulación teórica, esfuerzos basados en meros intentos individuales. Este es un capítulo histórico clave de la cultura de CMU.

2.3. EL "COMPLEX INFORMATION PROCESSING" Y LA "ARTIFICIAL INTELLIGENCE" DE CMU.

2.3.1. EL DOBLE PROGRAMA DE LA A.I. DE CMU.

Una cultura son los seres humanos y sus obras, como escribió el antropólogo norteamericano Melville Herskovits(1981). En esta cultura informática, las obras son los programas de ordenador, los sistemas construídos.

Estos forman parte de un programa cultural más amplio que resume la visión del mundo de sus autores, establece sus prioridades de investigación y determina sus estrategias a seguir.

En 1987, H. Simon definía así la disciplina en la Encyclopedia of Artificial Intelligence: " Artificial Intelligence (AI) is a domain of research, application and instruction concerned with programming computers to perform in ways that,if observed in human beings, would be regarded as intelligent" (1987:xi).

Y continuaba afirmando: "1.AI in the narrow sense is a part of computer science, aimed at exploring the range of tasks over which computers can be programmed to behave intelligently. ...2.The second branch of AI, part of the new field of cognitive science, is aimed at programs that simulate the actual processes that human beings use in their intelligent behavior. These simulation programs are intended as theories (systems of different equations) describing and explaining human performances" (Ib. ix).

La AI en CMU, bajo la dirección de Herbert Simon y Allen Newell, puede considerarse partícipe de dos ciencias, la computer science y la cognitive science, pero en cualquier caso se sitúa, según estos autores, en el terreno firme de la ciencia.

Este doble aspecto de la A.I. es propio de Carnegie Mellon, y responde a una estrategia de investigación que se ha mantenido en lo fundamental idéntica desde 1955. Los dos objetivos perseguidos han sido simultáneamente: " One was to use the computer to understand better how human thinking works by modeling it--by simulating it--and the second was to get computers to do smart things".

(Post-Gazette 1980).

En entrevista personal para estudio , Herbert Simon confirmó que : "El objetivo de mi vida ha sido entender la mente humana. Y en todos sitios funciona igual, por búsqueda heurística " (3/11/90). A la pregunta de cómo se entendía que construyera programas de ordenador contestó: "De la misma forma que Mendel cultivo guisantes para mejor entender la transmisión de caracteres". (ibid).

Por su parte Allen Newell, en una reciente conferencia en la SCS, confirmó que su "single ultimate scientific question" había sido, no los computadores ni la A.I., sino la "nature of human mind". ("Desires and Diversions. 6/12/91).

Newell y Simon, personajes que se reconocen al ser llamados "the parents of artificial intelligence" (Models. 1991:190), han representado durante 30 años lo

que se denomina la versión dura en Inteligencia Artificial, "strong AI". Según ésta, el ordenador tenía los medios suficientes para adquirir una inteligencia similar a la del ser humano. La versión "blanda" aceptaría construir programas de ordenador que manifestaran conducta inteligente, pero no necesariamente imitando a la humana. El M.I.T. está ahora, por ejemplo, construyendo una generación de robots inteligentes imitando la inteligencia de los insectos. El programa de la A.I. ha sido entendido de forma distinta en los otros centros de país. Así, Patrick H. Winston, Director del M.I.T. Artificial Intelligence Laboratory, afirma que : "The primary goal of the Artificial Intelligence Laboratory is to understand how computers can be made to exhibit intelligence"(Kernoff,Who's who in A.I.:132). Comprender cómo pueden ser hechos los ordenadores para que exhiban una conducta inteligente.

De esta finalidad principal se derivan dos objetivos secundarios, dos consecuencias: " "Two corollary goals are to make computers more useful and to understand certain aspects of human intelligence". Pero la premisa no es cómo en CMU "understanding the human mind".

Por su parte, para Avron Barr y Ed Feigenbaum, del Department of Computer Science de Stanford University: "Artificial Intelligence is the part of computer science concerned with designing intelligent computer systems, that is, systems that exhibit the characteristics we associate with intelligence in human behavior-understanding language, learning, reasoning, solving problems, and so on." (Handbook, 1981:3).

El modelo Newell-Simon constituye un modelo único, específico de la Inteligencia Artificial de Carnegie Mellon . Es un modelo nacido en 1955 y ha marcado a la AI de CMU con unos rasgos culturales propios, que la hacen distinta a otros programas seguidos en MIT o Stanford, los otros dos centros de excelencia reconocidos de Estados Unidos en este campo. Ello no significa que sea el único modelo en la AI de CMU.

¿Cuál es la diferencia?. Es un programa en que sus fundadores tienen como objetivo fundamental: "understanding the human mind".

Para H. Simon, "What is unique here is the closeness of the two sides of the coin -the human side plus the straight artificial intelligence. We have more of a balance and union of the two here than any other of the other places. I guess we at CMU, plus the people who are interested in this over at the University of Pittsburgh, have by a factor of two or three the largest collection of

activities of this sort in the country". (: Post-Gazette: 1980).

Mientras en el M.I.T. y Stanford, la Inteligencia Artificial se considera sólo una disciplina parte del Computer Science, en Carnegie Mellon se denomina Inteligencia Artificial a una misma actividad realizada tanto en la School of Computer Science como en el Department of Psychology del College of Humanities and Social Science, pero presidida por este departamento. La psicología cognitiva dirige, en gran medida, el programa de investigación de la A.I. de CMU.

2.3.2. EL PROGRAMA CULTURAL UNITARIO DEL "COMPLEX INFORMATION PROCESSING".

El programa antes citado se basa en una "teoría unificada de la cognición", expuesta en su forma más completa en la obra HUMAN PROBLEM SOLVING, escrita en 1972.

Para H. Simon y A. Newell existe una igualdad entre la mente humana y el ordenador a la hora del comportamiento inteligente. "Our theory of human thinking...postulates that the human operates as an information processing system". (1972: 19).

La idea que unifica mente humana y ordenador es la de "Physical Symbol Systems". Esta hipótesis es para Allen Newell, "the most fundamental contribution so far of artificial intelligence and computer science to this joint enterprise had been the notion of physical symbols system" (1980:1). Así, lo afirmó en la reunión constitutiva de la Ciencia Cognitiva celebrada en La Jolla, California en 1980.

Para Newell y Simon no se trata que el ordenador sea una "metáfora" de la mente humana, sino que tiene los medios necesarios y suficiente para la misma acción inteligente en el mundo que la mente humana.

A esta hipótesis se la denomina "PHYSICAL SYMBOL SYSTEM HYPOTHESIS". Así, para A. Newell, su sistema inteligente SOAR es, en sí mismo, una teoría de la mente humana. Igualmente, los programas de simulación que escribe H. Simon intentan ser, como el mismo afirma más arriba "teorías...describiendo y explicando actividades humanas". Los programas de ordenador pueden ser en sí mismos teorías psicológicas.

En esto se diferencian Newell y Simon de la mayoría de psicólogos cognitivos. Muy pocos de ellos han construido programas de ordenador para implementar sus teorías. La psicología cognitiva considera mayoritariamente al ordenador como una

herramienta para simular procesos cognitivos humanos, una "metáfora" de la mente humana, no un mismo tipo de sistema simbólico.

Por otra parte, para otros fundadores de la Inteligencia Artificial, como Marvin Minsky del M.I.T., la inteligencia exhibida por las máquinas puede basarse en principios simples de "feedback" como los de la cibernética, en principios biológicos como las redes neuronales o imitando a los humanos. Pero no necesariamente de esta última forma.

La posición radical de Simon y Newell, que ha marcado y sigue marcando el contenido de la investigación en Inteligencia Artificial en CMU, y no sólo de la AI sino de todo la Computer Science, implica que la AI ha de basarse en la construcción de programas informáticos que manifiesten una actividad inteligente semejante a la humana, una actividad inteligente de carácter general y no sólo basada en conocimientos especializados, como los sistemas expertos inventados por Edward A. Feigenbaum, uno de los primeros discípulos de H. Simon en CMU. En la A.I. de SCS hasta los años 90s no se consideró importante la investigación en Sistemas Expertos, a pesar de lo cual se realizó alguno importante como el R1 de Peter McDermott.

Esta unidad disciplinaria de la Inteligencia Artificial, dividida en dos patas una en Computer Science y otra en Psicología Cognitiva, y dirigida por esta última, se la denomina en Carnegie Mellon, "Complex Information Processing". Corolarios de esta visión es que las teorías psicológicas del Departamento de Psicología tendrán validez si se expresan en programas informáticos, y viceversa los programas informáticos de la AI en la SCS han de buscar representar una acción general inteligente. La consecuencia en los años 80s ha sido la nueva generación de arquitecturas inteligentes iniciada por Allen Newell con el programa SOAR, y la nueva disciplina del MACHINE LEARNING, puesta en marcha por J. Carbonell y T. Mitchell.

Las estrategias no obstante seguidas por Newell y Simon han sido distintas. Allen Newell se ha centrado en la construcción de arquitecturas cognitivas, como SOAR, un sistema, por un lado, que se pretende capaz de una acción general inteligente en el mundo, y al mismo tiempo, considerado una teoría unificada de la cognición.

Por su parte, Herbert Simon se ha dedicado en los últimos 10 años a simular, en forma de programas de ordenador, lo que desde una mentalidad científica puede considerarse las más avanzadas producciones cognitivas humanas: los

descubrimientos científicos. BACON es un programa para simular ciertos procesos de descubrimiento científico.

La extensión de este paradigma conlleva el que no sólo las teorías de la cognición humana realizadas en el Departamento de Psicología de CMU, sino cualquier teoría realizada en cualquier otro departamento para tener validez a los ojos de esta concepción del conocimiento tenga que ser presentada en forma de software informático. Pero a cambio, todo programa informático es una teoría científica.

Según este programa cultural, conocimiento significa "computation", y "computation" es la actividad de procesamiento de la información que realiza la mente humana. No existe por tanto ninguna razón para que sus resultados no puedan ser expresados en programas de ordenador. En principio esto es válido para todo "el universo de las disciplinas".

Las consecuencias de este programa de investigación científica, por usar una terminología de Lakatos, han afectado a todo el campus de CMU durante los últimos 25 años. Uno de los objetivos claves de la red ANDREW tanto en su comienzo en los años 80s como ahora es conseguir que los profesores del campus de cualquier disciplina pudieran producir software educativo. Como afirmaba James Morris, director del Information Technology Center: "The ultimate producers of educational software must be faculty members...We and the CDEC diffuse technology and tools to them"(IBM, 1986:16). En estos años, se han construido ya docenas de "courseware", o cursos educativos informáticos en disciplinas tan diversas como arquitectura, arte, filosofía, física o ingeniería.

Ha empezado a nacer un tipo de profesor-programador que traduce sus conocimientos tradicionales a programa de ordenador, como base educativa y de conocimientos. El Dr. Preston Covey, profesor de Ética y director del Center for Design of Educational Computing, es una muestra. Su video disco interactivo, "Right to Die?" es un programa de ética, construido bajo soporte informático, en el que estudiante se enfrenta ante un dilema de la vida real: ha de decidir si permite o no a un ser humano que sufrió un accidente el derecho a quitarse la vida.

2.3.3. "THE PHYSICAL SYMBOL SYSTEM HYPOTHESIS".

En su último libro, *Unified Theories of Cognition* (1990), Allen Newell afirma

que : "Science, even more profoundly than politics, is the art of the possible. It does only that can be done next" (1990:26). Allen Newell es miembro a la vez de la National Academy of Sciences y de la National Academy of Engineering. Y la definición de ciencia citada es significativa de lo que Herb Simon, experto en ciencia de la administración, llamó "las ciencias de lo artificial", expresión nacida en la cultura de CMU.

Esta figura del profesor-programador conlleva cambios importantes en la concepción del propio saber científico, aún conservando su naturaleza esencial. La ciencia se hace programadora, diseñadora, a cambio de intentar hacer el diseño una ciencia.

La hipótesis en la que se basa la AI tal como se concibe en CMU se denomina: "PHYSICAL SYMBOL SYSTEM HYPOTHESIS", o hipótesis del sistema de símbolo físico, formulada por Herb Simon y Allen Newell en 1975, coincidiendo con la concesión del Turing Award, el premio más prestigioso que la Association for Computing Machinery otorga a los computer scientists que han realizado una contribución más importante en el campo en los Estados Unidos.

Esta hipótesis dice así: "A physical symbol system has the necessary and sufficient means for general intelligent action" (Newell, Simon, 1976:116).

¿Qué significa para estos autores "a physical symbol system"? Este sistema es "a machine that produces through time an evolving collection of symbol structures..it bears a strong family resemblance to all general purpose computers".(ib. :116).

¿Qué significa a su vez "a general intelligent action"? Como ellos explican:

"By 'general intelligent action' we wish to indicate the same scope of intelligence as we see in human action: that in any real situation behavior appropriate to the ends of the system and adaptive to the demands of the environment can occur, within some limits of speed and complexity" (ibid:116)

Se podría resumir esta hipótesis de la siguiente manera: un sistema de símbolos físicos, como el ordenador digital, tiene los medios necesarios y suficientes para una acción general inteligente en el mundo, similar a la que vemos en una acción humana.

Para Allen Newell, "The notion of general intelligence can only be informally circumscribed, since it refers to an empirical phenomenon. However, the intent is clear --to cover whatever will come to be called intelligent action as our understanding of the phenomena increases...General intelligence implies that

within some broad limits anything can become a task..." (1980:42).

Herb Simon está de acuerdo con esta idea: " Creo que en nuestra época las computadoras podrán realizar cualquier tarea cognoscitiva que pueda realizar una persona" (1977: 6).

Allen Newell se ha concentrado en la construcción de arquitecturas integradas inteligentes, o sistemas informáticos que simulan o pretenden dicha acción general, como SOAR, consideradas a su vez como teorías unificadas de la cognición. Herbert Simon se ha centrado más en lo que llama "teorías de rango medio", o sistemas informáticos que sólo simulan determinadas tareas inteligentes como por ejemplo descubrimientos científicos. El objetivo ha sido el mismo, la estrategia distinta.

El problema es que esta hipótesis sólo se puede verificar construyendo ese sistema. No hay aquí sistema a menos que se construya, y si no se construye no puede validarse la hipótesis. El problema es que lo que está enunciando , no es una mera proposición empírica, el "un juicio de hecho" de Hume, sino una intención humana. No es una proposición sino un propósito.

Es un enunciado que no describe un acontecimiento dado sino que ordena una acción, una tarea: construir un sistema de símbolos físicos que tenga los medios de una acción inteligente general. Enuncia una "posibilidad sugerida" en palabras del filósofo pragmático norteamericano John Dewey. (1989. Cómo pensamos.:26). Simon indica:"las computadoras podrán realizar cualquier tarea cognoscitiva que pueda realizar una persona". Es una hipótesis que sólo se puede validar haciendo un programa informático. Dado su carácter normativo, implica un discurso ético que compromete la voluntad y el deseo ("desire state") del que lo emite. También un enunciado científico responde a una ética, como demostró R. Merton: lo que es, debe de ser. El problema es que la hipótesis formulada por Newell y Simon no enuncia un "hecho", un "facta" sino un "futura" en palabras de Bertrand de Jouvenel.

Esta hipótesis puede ser, en efecto, verificada. La Physical Symbol System Hypothesis no es un mero deseo, un "desired state", o "buenas intenciones". Implica tres acciones: 1. dicho deseo de construir ese sistema, 2. un conocimiento tanto de las capacidades del ordenador como de las de la mente humana. Y algo más importante, 3. una vía práctica de solución del problema o de validación de la hipótesis: el saber cómo se va a construir ese physical symbol system.

Newell y Simon comparan su método con la teoría de los gérmenes : " The basic paradigm for the initial testing of the germ theory of disease was: identify a disease; then look for a germ. An analogous paradigm has inspired much of the research in artificial intelligence: identify a task domain calling for intelligence; then construct a program for a digital computer that can handle tasks in that domain." (1976:118).

Pero una cuestión es buscar un germen causante de una enfermedad, y otra construirlo. Una actividad es un "look for" y otra distinta "construct". La primera es la propia de la ciencia, la segunda le corresponde a la ingeniería. Se ha formulado una hipótesis ingeniera, no científica.

En 1980, Allen Newell reconocerá que este concepto, de "physical symbol system" tiene un origen en la lógica y la ingeniería: " This concept has not followed the usual path of scientific creation, where development occurs entirely within the scientific attempt to understand a given phenomenon. It was not put forward at any point in time as a new striking hypothesis about the mind, to be confirmed or disconfirmed. Rather, it has evolved through a much more circuitous root. Its early history lies within the formalization of logic, where the emphasis was precisely on separating formal aspects from psychological aspects. Its mediate history lies within the development of general purpose digital computers, being thereby embedded in the instrumental, the industrial, the commercial and the artificial --hardly the breeding ground for a theory to cover what is most sublime in human thought". (1980:2).

Los programas informáticos no existen antes de su construcción. No se pueden buscar como si fuera un germen. Una vez construido , en efecto se puede analizar y estudiar empíricamente, pero dicha tarea de descubrimiento es posterior a la construcción del programa. Estos se construyen para probar una "posibilidad" (la tarea inteligente que se considera posible reproducir en el ordenador) no para comprobar una "realidad" dada.

John McCarthy , profesor de AI en Stanford, y uno de los fundadores de la AI, y P.J. Hayes de Edinburg, en 1969 señalaron el papel clave de la forma verbal modal "can" en la construcción de programas de ordenador y su estrecha relación con el "free will" o el libre albedrío humano: "The problem of 'free will' assumes an acute but concrete form. Namely, in common-sense reasoning, a person often decides what to do by evaluating the results of the different actions he can do. An intelligent program must use this same process, but using an exact

formal sense of can, must be able to show that it has these alternatives without denying that it is a deterministic machine". (1969: 467).

El contenido teórico de la A.I. esta por ello en los programas que construye.

Estos programas pueden tener objetivos más ambiciosos (SOAR) o más limitados (BACON) pero su construcción es el sentido final de la hipótesis. Por ello la prueba es el programa mismo que funciona, no una recolección empírica de datos o una teoría científica al modo de la física.

La física matemática ha sido hasta el momento la ciencia que ha servido hasta ahora como patrón de conocimiento para el resto de saberes académicos. Las Leyes del Movimiento de Isaac Newton, los famosos Axiomas, condensan una aportación clave de esta disciplina. Recordémoslos : "Law I. Every body continues in its state of rest, or of uniform motion in a right line, unless it is compelled to change that state by forces impressed upon it....Law II. The change of motion is proportional to the motive force impressed; and is made in the direction of the right line in which that force is impressed... Law III. To every action there is always opposed an equal reaction: or, the mutual actions of two bodies upon each other are always equal, and directed to contrary parts" (Newton, 1946:13).

Comparando estas leyes con la hipótesis más importante de la Inteligencia Artificial, según Newell y Simon podemos apreciar diferencias importantes.

La formulación newtoniana es una construcción explicativa de cómo funcionan unas determinadas fuerzas del universo físico. Su resultado son leyes explicativas, que descubren un fenómeno natural dado. La hipótesis del "Physical Symbol System" es un programa de una disciplina normativa y descriptiva al mismo tiempo, según Simon. Se basa en la "intención" (Newell) o la "creencia" (Simon) en que una máquina puede realizar cualquier tarea cognoscitiva que pueda realizar una persona. Esta máquina aún no existe pero se ve posible diseñarla.

El resultado de la primera es ciencia natural, el de la segunda, una "artificial science", según Simon . Se cree que el resultado final de esta será el saber como construir un sistema artificial que tenga las mismas capacidades cognitivas que el ser humano. Sistema que primero habremos tenido que construir. Es como si Isaac Newton tuviera que haber construido primero un universo con sus leyes, que intentaba probar, y, una vez finalizada la obra, comprobara de esta forma que , en efecto, eran ciertas.

En el primer caso el criterio es la verificación, o su falsación. En el segundo caso, el criterio es su validación, medida en términos ingenieros sobre si

funciona o no . "It works". Si en efecto se realiza, la hipótesis será válida.

De lo contrario, no. Newton no ha de ejecutar sus leyes, se limita a enunciarlas. El investigador de A.I. ha de poner en práctica sus hipótesis, como única forma de validarlas. Un enunciado científico puede ser verdadero o falso. No tiene sentido preguntarle eso a un programa informático. Pueden ser, incluso, correctas sus inferencias lógicas y al final no funcionar por estar mal planteado el problema a resolver.

Herb Simon y Allen Newell en su famosa conferencia "Computer Science as Empirical Inquiry: Symbol and Search" (1976) sostuvieron que ésta era una disciplina empírica, "an experimental science, but like astronomy, economics, and geology, some of its unique forms of observation and experience do not fit a narrow stereotype of the experimental method". (ibid:114). ¿Por qué no encaja en el estereotipo del método experimental?. El método de la ciencia experimental se pregunta: "¿qué existe y qué es?" (Kunh. 1965:97). El método de la computer science se plantea: 1. que sea lo que deseo ("wants"). 2. que es lo que existe y 3. cómo hacer para que sea lo que deseo a partir de lo que existe.

El primero método parte de una realidad anterior al científico. En el segundo caso, el computer scientist es previo a su "computer". William James, pragmatista norteamericano, expresó esta visión al afirmar: "La verdad se hace".

Y para hacerla hace falta un hacedor, un programador. El símbolo físico lo hace el computer scientist. No sólo construye la hipótesis, como todo científico.

Construye el propio resultado de la investigación. Volviendo a la comparación con Mendel, la diferencia entre Simon y Mendel es que éste no tuvo que construir el programa genético de los guisantes, se limitó a estudiarlo. Los computer scientists han de hacerlo para poder estudiarlo luego empíricamente. Esta simple inversión cambia toda la relación "Science&Technology" en otra "Technology&Science" al menos en este campo del saber.

A la hora de evaluar si esta hipótesis ha sido probada y por tanto es una teoría de Inteligencia Artificial, su primera teoría, las opiniones no son unánimes.

Para Ed Feigenbaum, por ejemplo, en la reunión celebrada en 1987 en honor de Herb Simon en CMU decía: "If we ask whether general intelligent action involving commonsense reasoning has been empirically demonstrated, the answer is not.

However, because I am inclined toward the dimension I called competence/performance, I assess the evidence as providing strong confirmation of the PSSH". (1989:178). Y concluía sugiriendo que se pasara a denominarla "

the Law of Intelligence of Physical Symbol Systems".

No obstante, si nos atenemos al hecho empírico, en 1991 los sistemas de símbolos físicos existentes en la actualidad no tienen aún esa capacidad de acción general inteligente, pero manifiestan determinadas conductas que sí lo son.

Existen son numerosos programas informáticos, resultado de más de 25 años de A.I., que exhiben acciones inteligentes parciales, especializadas. El intento más ambicioso es el de construir arquitecturas inteligentes integradas, sistemas capaces de aprender como SOAR, PRODIGY o THEO.

Todo ello no invalida la hipótesis para sus autores. Lakatos ya indicó que un programa de investigación no se cambia por que se encuentren pruebas que falsean sus hipótesis, como creía Popper, sino cuando se presenta un nuevo programa de investigación que supere al anterior.

2.3.4. LA MENTALIDAD PRAGMATICA NORTEAMERICANA Y MAS ALLA: EL CONOCIMIENTO-PROGRAMA.

Este programa de investigación de Simon y Newell para entenderse ha de situarse en el marco de un programa de conocimiento cultural más amplio que el de la propia ciencia. Necesitamos analizar sus bases culturales, su "situación" (L. Suchman. 1987), el contexto cultural de donde ha salido.

Newell ha definido la computer science como una "ciencia experimental". Simon opone su enfoque "empírico" al "lógico-formal". Pero el programa de investigación de la A.I. en CMU no viene del experimentalismo inglés que arranca con Bacon.

El programa de Newell y Simon entronca directamente con un bagaje cultural en materia de conocimiento, específicamente norteamericano. Por un lado, es un programa que se alimenta del pragmatismo, la contribución más específicamente norteamericana a la filosofía occidental. Esta filosofía ha sido tradicionalmente poco estudiada en Europa, pero sin cuyo conocimiento no se puede entender el significado de la tecnología en Estados Unidos, y en particular, de la "computer science and technology" en CMU.

En 1977, en su libro "La Nueva Ciencia de la Decisión Gerencial", el propio Herbert A. Simon se definía como un "radical tecnológico, un conservador económico y un pragmatista filosófico". De este mismo libro es su expresión: "En el campo de las computadoras la hora de la verdad es un programa que funciona: todo lo demás es profecía." (ibid.:8).

Desde finales de los 80s, apreciamos en Estados Unidos una ofensiva contra la identificación mente humana-ordenador caracterizada como visión racionalista y unilateral del conocimiento.

Así T. Winograd y F. Flores (1989) han elaborado, desde la propia comunidad de computer scientists norteamericana, una crítica al racionalismo de Simon.

Apoyándose en diversas escuelas europeas, principalmente la hermenéutica alemana de Heidegger y la filosofía del lenguaje anglosajona de Austin, colocan a Herb Simon dentro del programa filosófico de la "tradición racionalista". Desde este ángulo, intenta ofrecer una alternativa basada en el "diseño ontológico"(1989:217).

Esta opción dicotómica entre saberes, "razón" versus "diseño", lo volvemos a encontrar en el psicólogo Jerome Bruner (1990) que anuncia su desencanto y separación de la "revolución cognitiva" del que fue uno de los fundadores, oponiendo "information" a "meaning". Según este autor, la revolución cognitiva con su metáfora de la computación, ha dejado de lado el mundo del significado, de la polisemia, la metáfora y las conexiones connotativas.

Bruner viene sosteniendo que existen dos modos de pensamiento o "modes of thought", irreductibles uno al otro: "One mode, the paradigmatic or logico-scientific one, attempts to fulfill the ideal of formal, mathematical system of description and explanation..." (1986:12). El otro es el "narrative mode": "The imaginative application of the narrative mode leads instead to good stories, gripping drama, believable (though not necessarily 'true') historical accounts. It deals in human or human-like intention and action..." (ibid: 13).

También la antropóloga Lucy Suchman (1987), de Xerox PARC, opone también la "situated action" al "plan", forma de pensar característica, según ella, del racionalismo europeo.

Pero estas visiones dicotómicas son muy propias de la cultura filosófica europea, y no acaban de encajar en una cultura de la "computer science" norteamericana como la de CMU. Herbert Simon ha defendido en efecto la racionalidad, pero dentro de un marco cultural tecnológico que hace del "design" su forma de pensar nodal.

Bruner, en crítica al cognitivismo, se coloca en la tradición de Geertz, Goodman, y otros científicos sociales y filósofos constructivistas. Pero el cognitivismo norteamericano, criticado por estos de racionalista, proviene de una concepción ingeniera de la inteligencia, de una tradición constructivista,

no de una concepción racionalista.

Ninguna de estas críticas acierta a señalar que la cultura informática es un invento norteamericano, y que existe una estrecha relación de la misma con el programa pragmático, la filosofía nacional de este país. Esta filosofía, al menos en autores como William James, fue un intento de conciliar ciencia y religión, la clásica dicotomía de saberes de la cultura europea. Y el terreno que encontró para ello fue el de la acción humana, el campo de las "posibilidades no garantizadas" (W. James. 1984:234).

Se puede considerar el programa de Newell y Simon como un programa racionalista.

Simon, en efecto, defiende al "rational man". Pero Simon no es Aristóteles. En CMU, Simon tiene dos caras como el dios Jano: por un lado, ante los científicos, los matemáticos y los economistas, se le conoce por su crítica sistemática a la visión del "rational behavior" de Von Newmann y la economía clásica. Se le considera un teórico de la organización, que empezó su carrera y ha dedicado su vida a teorizar la "bounded rationality", los límites de la racionalidad: :

Rationality, then, does not determine behavior...Instead behavior is determined by the irrational and nonrational elements that bound the area of rationality.

The area of rationality is the area of adaptability to these nonrational elements...Hence , administrative theory must be concerned with the limits of rationality". (Models, 1991:88).

Este combate se ha traducido dentro de la A.I., Simon por su oposición al paradigma denominado "Logic" , y la defensa de los métodos heurísticos, los menos racionales y más empíricos, contra el enfoque formalista lógico y algorítmico al tema. Métodos que ha considerado los apropiados para el tratamiento de los "ill structured problems", o problemas que incluyen decisiones de política novedosas o mal estructuradas (Simon,1977: 45).

Sin embargo, Simon ha presentado también la otra cara: la del científico que ha introducido los métodos cuantitativos en la ciencias sociales desde los 40s. La del científico que ha intentado convertir el diseño ingeniero de arte en una ciencia. Se da la paradoja que ha recibido un Premio Nobel en economía siendo que ha dedicado su carrera profesional criticando esa disciplina y su "model of man", y oponiendo a la "mano invisible" el "decision making" de la conducta administrativa.

Esta mentalidad pragmática explica una notable capacidad de absorción que está demostrando la computer culture de CMU para integrar artistas, matemáticos,

psicólogos o managers en el universo computacional.

Charles Peirce fue el primero en establecer lo que William James consideraba el principio del pragmatismo. En un artículo titulado: "How to make our ideas clear?" (1878), aquél filósofo estableció que nuestras creencias eran reglas para la acción. Para desarrollar el significado de un pensamiento necesitamos determinar qué conducta es la adecuada para producirlo: tal conducta es para nosotros toda su significación.

En la tradición científica europea un pensamiento, en forma de teoría, hipótesis, etc. es significativo si se propone explicar racional o empíricamente un fenómeno de la naturaleza. Los problemas están planteados en forma de incógnita a resolver, sin certeza de que lo sean.

Aquí sin embargo, los problemas se formulan como "problem-solving", problema a resolver. La finalidad es su resolución. Lo que da sentido al problema es que se pueda resolver. Para ello, el problema es un "issue" que inmediatamente se transforma en una "task", una tarea a ejecutar.

Son propias de esta cultura, los "task force" u organizaciones para una tarea, para una operación. La red Andrew fue promovida por un Task Force on the Future of Computing.

En esta cultura, el mundo se ve como un "problem solving". Se parte de una brecha o "gap", una brecha entre un "desired state" del individuo y el "initial state" del que parte. Como señala John Hayes, profesor de psicología cognitiva en CMU y encargado de los cursos de problem solving en la facultad de Humanities and Social Sciences de CMU: " Whenever there is a gap between where you are now and where you want to be, and you don't know how to find a way to cross that gap, you have a problem". (1989:xii).

La tarea es la forma de resolver este "gap". Esta estructura de conocimiento la denomina Newell : "problem space", y es la estructura básica de la Inteligencia Artificial, y de la Computer Science.

Un programa de ordenador es esa secuencia de pasos dirigida a conseguir que la máquina realice la tarea asignada. En este marco se explica que hacer teoría se identifique con hacer programas. Por eso A. Newell puede considerar a SOAR, un programa de ordenador inteligente, como una teoría de la cognición humana.

En una primera etapa, el pragmatismo alentó el conductismo o "behaviorism" norteamericano. La acción se concibió como simple "conducta". Era la época del "learning by doing", de la "efficiency". Saber era hacer. Con todo, la Depresión

y sobre todo la II Guerra Mundial generó una distinción entre "behavior" y "planning". Se distinguió entre el nivel de planeamiento de la acción, y la propia acción. La era de Roosevelt y el New Deal introdujo la planificación en proyectos como el Tennessee Valley. La II Guerra Mundial profundizó este pensamiento estratégico. Por último, el Manhattan Engineering District educó a la élite de la ciencia en esta mentalidad planificadora, propia de la ingeniería.

En el campo de la ciencia social, psicólogos norteamericanos como Tolman (Purposive Behavior in Animals and Men, 1932.) empezaron a destacar la "purposive behavior". Línea que fue continuada en los 60s por G. Miller, E. Galanter y K. Pribram: "Plans and the Structure of Behavior" 1960. La cibernética de N. Wiener (Rosenblueth-Wiener-Bigelow: Behavior, Purpose and Teleology, 1943) fue la culminación de este periodo. El ordenador se convirtió en una "general-purposeful machine". El conocimiento pasó a considerarse no como una mera acción (behavior), sino como un programa de acción.

Desde esta óptica, los conocimientos lógico-matemáticos o empíricos de la cultura científica europea importados durante y tras la II GM han sido utilizados en este país para generar disciplinas como la computer science o la A.I., distintas a las antiguas disciplinas científicas europeas. Es tradicional en Europa reducir el fenómeno informático al concepto "información", olvidando la aportación específica de la computer science, la idea de "programming", la información procesada para ser ejecutada por una máquina. El "processing" no es el adjetivo sino el sustantivo en la expresión "information processing". Ello está tanto más dificultado, dado que en países como Francia, al "programming" se le denomina "logiciel".

En Computer Science, la lógica o la matemática se utiliza para construir un lenguaje de programación distinto al clásico lenguaje matemático de la ciencia europea. Como indica Newell: "Modern logic, of course, was integrally involved in the development of the digital computer....But logic did not enter A.I. at all as the logic of thought...Logic was part of the underlying technology of making mechanisms do things". (1982:8).

Es una matemática para cumplir tareas, donde un concepto básico es el de función matemática. Por otra parte las heurísticas o técnicas de descubrimiento ya no se utilizan como los griegos para resolver problemas de geometría sino para la toma de decisiones por el programa de ordenador. Las heurísticas del L.T. no eran

para buscar ("search") en el sentido empírico del término sino mecanismos de ayuda a la toma de una decisión para generar una acción. Incluso el mecanismo "If...then" de la A.I. tiene un doble sentido: uno, el de la inferencia lógica aristotélica y el otro que forma parte de la casuística procedimental propia del pensamiento normativo de la jurisprudencia. No se pretende inferir conocimiento descriptivo sino, ante tal o cual caso, emitir una orden que solucione un "impasse" de la máquina en su proceso de ejecución de un programa. Este saber no es sólo "performative", como ha señalado F. Lyotard en "La condition postmoderne". Es "programador". Su clave no es la "performance", sino el "programming". La cultura que inicia es la de la programación, la de una sociedad programadora, y en este sentido, prefigurativa.

2.3.5. "COMPUTER SCIENCE AS AN EMPIRICAL INQUIRY."

Este saber también permite el considerar la computer science como una ciencia empírica, pero no como disciplina previa y fundacional del ordenador sino secundaria e instrumental para su diseño.

En 1976, Simon y Newell, al recibir el premio de la ACM, el más importante en esta comunidad científica, definían computer science como una "empirical inquiry" : "Computer Science is an empirical discipline. We would have called it an experimental science, but like astronomy, economics, and geology, some of its unique forms of observation and experience do not fit a narrow stereotype of the experimental method...Actually constructing the machine poses a question to nature; and we listen for the answer by observing the machine in operation and analyzing it by all analytical and measurements means available...They are and we can open them up and look inside...As basic scientists we build machines and programs as a way of discovering new phenomena... (1976:114).

En efecto, una vez construídos se le pueden aplicar al ordenador los métodos de la ciencia empírica. Y ese ha sido la finalidad de gran parte del programa de Newell y Simon, considerar al ordenador como un "tool" para comprender la mente humana.

El problema es que hay algo que no encaja, según sus autores ("do not fit").

¿Qué es?. Que esta visión del ordenador no explica cómo se inventa el ordenador. Y por lo tanto no se puede entender el conocimiento del inventor del ordenador, y en consecuencia no se puede comprender "the human mind", al menos la mente de este inventor. Hay un agujero negro en este esquema: ¿qué tipo de conocimiento

humano es el que construye el ordenador y permite que a continuación los "basic scientists" lo analicen empíricamente?.

Este aspecto que no encaja en la teoría de la computer science como "empirical enquiry" es el que nos conduce a la hipótesis inicial de este trabajo:

"Knowledge is design". El "computer" no es un simple "fact" es un "deed", un acto humano, un acto libre. Y éste es el que provoca una invención tecnológica. Y esta invención permite a la ciencia entrar a continuación a realizar su labor analítica, que servirá para seguir inventando nuevos artefactos.

2.3.6. "FACTS" Y "DEEDS".

Los organismos biológicos también se rigen por fines, por programas. Existe un código genético. Newell considera, por ello, la biología como una "science of the artificial". (Entrevista.17 dic. 90), en tanto que disciplina que analiza organismos con fines, mediante los procedimientos de la ciencia empírica. Simon opina lo mismo para la ciencia de la administración. Y lo mismo se podría decir de todas las ciencias sociales, incluida la antropología.

Pero hay un aspecto que sigue sin reconocerse. El "científico del diseño" diseña. No sólo analiza organismos o sistemas que tienen un programa, o sino que hace esos programas y esos sistemas. Lo cual no es el caso para la ciencia social positiva, al menos, hasta ahora.

En 1991, en su Autobiografía Simon ha revelado qué política han seguido Newell y él, históricamente, frente a las críticas de sicólogos o filósofos: "We have adopted the policy (was it the anarchist Bakunin's or Sorel's) of "propaganda of the deed, not propaganda of the word". The best rhetoric comes from building and testing models and running experiments. Let philosophers weave webs of words; such webs break easily". (Models,1991:272).

El lema de la Royal Society inspirada en la filosofía de Francis Bacon decía: "Nullius in verba". Pero el empirismo inglés no consideraba central la acción humana en un sentido constructivista. Su tema es la experimentación para conocer mejor a la Naturaleza. Era un programa de ciencia natural, en primer lugar, y sólo en segundo lugar, de tecnológico. Sin embargo, la filosofía nacional de Estados Unidos no es el empirismo sino el pragmatismo: la acción humana es la que determina el medio, "hace la verdad". Su matriz no es científica sino ingeniera. Confundirlas es un error. El hecho es que Newell y Simon lo hacen sistemáticamente. La única explicación posible es para justificar

científicamente sus construcciones tecnológicas.

Años antes, Simon confesaba a su hija Barbara, en medio de la polémica nacional en Estados Unidos sobre el carácter moral de la Inteligencia Artificial, su actitud al respecto: " My own posture has been this: my scientific work, and that of other AI researchers, will, in the long run, determine how many of Man's thinking processes can be simulated. I believe that ultimately all of them can, but feel no great urge to try to prove that to others who feel differently. In science, it is the facts that give us the final answers". (1991:274).

La ciencia física europea, hasta Heisenberg, se regía por el principio determinista, según el cual existía un orden natural independiente del observador que tenía sus leyes propias, necesarias, analizables por el científico como hechos empíricos, o "facts", a partir de cuya observación y con ayuda de la matemática se obtenían enunciados generales en forma de leyes. El científico era ante todo un observador, un "investigador" de los secretos de la Naturaleza. En 1927, Heisenberg con su principio de indeterminación, demostró que la observación del investigador influía en la posición de partícula observada. Cuanto más claramente quedaba especificada la posición de la misma, menos podía fijarse su momento. Con todo, la disciplina no ha cambiado. La física matemática, y su visión del Universo, como realidad comprensible por medio de la razón y los sentidos, sigue siendo la contribución más importante de la civilización europea a la historia del pensamiento humano.

El problema es que, ahora, con la Computer Science entramos en el "artificial world". Y en él un científico como Simon o Newell "determinan" este mundo, o al menos así lo creen. Profesores de la escuela como Dario Guise al ser preguntado por las razones de su elección de Computer Science, indica la libertad que permite: " You are only limited by your imagination...Total freedom with real world. Your own limitation are you.". (Entrevista.15 de mayo.).

No obstante "ciencia del diseño" y ciencia empírica son dos programas de investigación que no acaban de encajar en esta visión. La negativa a analizar el mismo proceso de diseño del ordenador reduce la comprensión de la propia "human mind". ¿Por qué no se analiza?. ¿Por qué se es empírico para el resultado y no se entra en analizar el agente que inventa la máquina?. Al preguntar a Herbert Simon si guardaba notas de su proceso de invención, me dijo que no. Tampoco las tiene Newell. Lo mas aproximado son los papeles que por esas fechas publicaron.

2.3.7. "NATURAL WORLDS" Y "ARTIFICIAL WORLDS".

Ha sido un computer scientist el primer científico en destacar la diferencia entre los "mundos naturales" propios de las ciencias tradicionales como la física y los "artificiales" propios de la ingeniería y la computer science. (The Sciences of the Artificial. 1981). Esto desde un punto de vista científico occidental es incorrecto dado que todos los mundos artificiales son a su vez naturales. Pero el ingeniero puede contestar que lo contrario no es verdad, dado que no todos los mundos naturales son artificiales. Artificial para una cultura de matriz ingeniera significa "man-made", hecho por el ser humano, y tiene sentido dado que significa diferenciar lo que es propio del ingeniero, el diseño, en lo que se distingue del científico.

Podría afirmarse que también las sociedades humanas son "mundos artificiales", y por lo tanto son un objeto de estudio propio de este tipo de ciencias. Y así lo ve Simon, dado que él empezó su carrera como científico de las organizaciones. Sin embargo, hay algo más en esta división que escapa al mero científico social. El "scientist of the artificial" no sólo estudia esos mundos, también, y sobre todo, los diseña. Lo cual no es normal en la ciencia social positiva o empírica. Y no sólo los diseña en el sentido que construye el cuerpo de herramientas teóricas para entenderlo, los diseña en el sentido ingeniero del término. Esto es, no sólo los piensa o los simula, sino que los hace, "man-made world". Para Simon, los "natural worlds" son el objeto de las ciencias naturales, que se basan en el descubrimiento. Sin embargo, los "artificial worlds" son el producto del diseño humano. Son un "collective artifact", un terreno común para tecnólogos y científicos sociales interesados en el diseño.

Según este autor: "With goals and "oughts" we also introduce into the picture the dichotomy between normative and descriptive. Natural science has found a way to exclude the normative and to concern itself solely with how things are. Can or should we maintain this exclusion when we move from natural to artificial phenomena, from analysis to synthesis?. (1981:7-8).

Esta pregunta ha recibido en Simon respuestas contradictorias.

Por un lado, en The Sciences of the Artificial afirma en una nota a pie de página en la edición de 1969 y de 1981: "I may say that I hold to the pristine positivist position of the irreducibility of "ought" to "is"...This position is entirely consistent with treating natural or artificial goal-seeking systems as phenomena, without commitment to their goals". (1981: 8).

Por otro, en el Congreso fundación de Cognitive Science , en 1980 en la Universidad de California San Diego, ha sostenido : "En las ciencias artificiales, lo positivo (descriptivo) y lo normativo no están nunca muy distanciados. Así, por ejemplo, en economía, "el principio de racionalidad" se afirma a veces como una invariante descriptiva, otras como un consejo a los que adoptan decisiones... De modo semejante ,en sicología, la perspectiva según la cual los sistemas inteligentes son adaptativos, pueden aprender y evolucionan, no impide que los estudiemos para descubrir como mejorar su aprendizaje o su capacidad para resolver problemas...La inteligencia artificial, en particular, se dirige a metas normativas y tiene que hacerlo. En sus aspectos aplicados, se interesa no sólo por comprender la inteligencia sino por mejorarla. Quizá nos sea necesario reconocerlo explícitamente, hablando de ingeniería cognitiva y no sólo de una ciencia de la cognición...Los riesgos de confundir lo normativo con lo positivo son menores comparados con las perdidas que provocaría aislar a la ciencia de sus aplicaciones técnicas". (1987:36).

Esta contradicción se ha vencido, no obstante, históricamente en favor de una posición positivista, coherente con su visión de la computer science como "empirical enquiry", y su medio universitario. A la pregunta sobre los distintos saberes en CMU y donde se ubicaba personalmente entre la ingeniería, la ciencia o el arte, Herbert Simon respondió que en la ciencia. (Entrevista. 17/12/90). Lo que resulta paradójico con las citas anteriores, y sin embargo, ambas citas son del mismo autor.

Una visión más detenida del tema nos permite explicar estas contradicciones. Un programa pragmático no se rige necesariamente por una lógica binaria (verdadero-falso) de tipo aristotélico. Como ya hemos señalado, la estructura del "human problem solving" está compuesta de tres elementos: "desired state" (o conocimiento normativo), "initial state" (conocimiento descriptivo) y un tercer factor nuevo: la actividad humana deliberada que soluciona la contradicción, entre mundo normativo y mundo descriptivo, operando, reduciendo la diferencia entre ambos mediante su actividad deliberada. (o programada en el ordenador). La distinción de la ciencia clásica entre normativo y descriptivo tuvo sentido para liberar a la ciencia ("conocimiento descriptivo") de la tutela de la religión ("conocimiento normativo"). Es la aportación clave del pensamiento griego: la escisión del Logos respecto al Mythos.

Pero este esquema ya no sirve para explicar la cultura tecnológica. Y menos la

norteamericana. Esta cultura, como ya señaló Margaret Mead, es profundamente moral y al mismo tiempo científica.

¿Cuál es el "ethos", el núcleo central de este programa cultural?. Por un lado se hace una defensa clásica de la visión científica, por la cual, todo es naturaleza.

En 1977, Herb Simon en un significativo artículo a la revista Science, titulado "What Computers Mean for Man and Society", afirmaba: " Perhaps the most important question of all about the computer is what it has done and will do to man's view of himself and his place in the universe.... The definition of man's uniqueness has always formed the kernel of his cosmological and ethical systems. With Copernicus and Galileo, he ceased to be the species located at the center of the universe attended by sun and stars. With Darwin, he ceased to be the species created and specially endowed by God with soul and reason. With Freud, he ceased to be the species whose behavior was-potentially- governable by rational mind. As we begin to produce mechanisms that think and learn, he has ceased to be the species uniquely capable of complex, intelligent manipulation of his environment.

What the computer and the progress in artificial intelligence challenges is an ethic that rests on man's apartness from the rest of the nature. An alternative ethic, of course, views man as a part of nature governed by natural law, subject to the forces of gravity and the demands of his body. The debate about artificial intelligence and the simulation of man's thinking is , in considerable part, a confrontation of these two views of man's place in the universe. It is a new chapter in the vitalism-mechanism controversy". (1983: 199-200).

En 1987, con motivo de la construcción de Andrew, Simon volvía a insistir en esta alternativa: "We are learning to describe the human mind as an information-processing system. We are learning to describe the human mind as a kind of computer made out of neurons that is able to input information, output information, store information, manipulate information. I earlier referred to the computer as a brain and to the fact that some people find that unsettling. Why unsettling?. Darwin persuaded a lot of us, apparently not everybody in this country yet but a lot of us, that we are simply wrong in thinking that our species was created in any different way than other species were created. So we lost that sense of uniqueness...But there are responses other than denying that

computers can in fact do that...Why don't we try to get our sense of worth in this world, not in uniqueness, not in being apart from the rest of nature, but in being a general and integral part of nature". (1987:14-15).

Por otra parte se sigue admitiendo que el orden ha sido creado y que el ser humano es creador.

Desde una visión europea, esta lectura de Darwin es sorprendente. El Darwin que se enseña en Europa no es el que demuestra que todas las criaturas han sido creadas iguales, sino el que con su teoría de la evolución, hace innecesaria la tarea de crear especies, incluida la especie humana. La selección natural ha sido criticada por el creacionismo religioso por atreverse a explicar el origen de las especies sin hacer intervenir la voluntad de un ser trascendente al propio orden natural, que "crea" dicho orden. Fue la clásica disputa sobre el "design argument", que Darwin, zanjó. (Hurlburt III, R., 1965).

Uno de los términos que sorprenden en CMU es la utilización del término "creation" por los propios científicos. Parece no haber contradicción entre creación y descubrimiento. Herb Simon titula el capítulo de su Autobiografía relacionado con CMU: "Creating a University Environment for Cognitive Science and A.I.". (1991:248)

¿Cómo se explica esta conciliación?

2.3.8. " THE UNITARIAN ETHOS".

Descubrimiento científico, creación artística e invención técnica no están separados como modos de conocimiento radicalmente distintos en el programa de Herbert A. Simon.

Por contra trata de mantenerlos unidos y teorizarlos unitariamente. En una entrevista personal mantenida el 30 de noviembre de 1990, el Dr. Simon, ante la sugerencia que podían existir distintos tipos de saber, acaloradamente exclamó:" Discúlpeme, pero esta división de conocimientos me ha molestado. He dedicado mi vida a entender la unidad de la mente humana. Y en todos los sitios funciona igual: por búsqueda heurística".

Esta defensa ética de la unidad del hombre y la naturaleza, así como del hombre y el ordenador, sólo puede encontrar una explicación en el terreno de la propia ética.

No es un secreto en Carnegie Mellon, que el Dr. Simon es miembro de la First Unitarian Church de Pittsburgh, en la que participa, y en la que da charlas

sobre educación o sobre temas en los que es experto. En esto se emparenta con otros destacados científicos, políticos y autores norteamericanos, como Thomas Jefferson, Ralf W. Emerson, el pastor William E. Channing, o el histórico presidente de Harvard, Charles William Eliot. Razón y religión no son incompatibles para el unitarianismo.

El unitarianismo es una religión de tipo racionalista, que se basa en la doctrina de la "unity, or oneness, of God". Se denominan a sí mismos "liberal Christians" y creen en la libertad humana.

Denominado Arminianismo en el siglo XVIII, este credo fue el determinante de un importante movimiento contra el calvinismo y su doctrina de la predestinación que tuvo lugar en Nueva Inglaterra. Fue el denominado "The Great Awakening" (Wright: 1955). Boston fue el centro de lo que se denominó la "liberal Christianity". Y entre sus figuras destacaron Charles Chauncy y Jonathan Mayhew. La mayor parte de sus líderes procedían de Harvard. Para uno de sus críticos, el pastor Jonathan Edwards de Yale, centro de la oposición calvinista, el punto clave que dividió ambas doctrinas fue "the freedom of will". Para arminianistas como James Dana o Samuel West, el ser humano era un agente moral ("agent") con capacidad de iniciar acciones en el mundo físico. "The will" era el poder de elegir ("power of choice"). Sus acciones estaban determinadas por "moral incentives", más que por causas físicas. El estímulo para la acción era el "motive". Por ello, el ser humano era "self-determined"(Wright :107).

La objeción principal frente al calvinismo fue "the moral argument", planteado por el pastor William Ellery Channing, uno de sus reconocidos líderes unitaristas. Frente a la doctrina de la predestinación y la depravación humana propia del calvinismo, el unitarismo planteó que "the ultimate reliance of a human being is an must be on his own mind...Men who, to support a creed, would shake our trust in the calm, deliberate, and distinct decisions of our rational and moral powers endanger religion more than its open foes". (Channing, Unitarian Christianity: 47). Para Channing: " Conscience, the sense of right, the power of perceiving moral distinctions, the power of discerning between justice and injustice, excellence and baseness is the highest faculty given us by God, the whole foundation of our responsibility, and our sole capacity for religion" (ibid:52).

Uno de los puntos básicos del unitarismo de Channing es que Dios es "incomprehensible" al ser infinito. Pero no por ello es "unintelligible".

Conocemos muy poco de él. "Our knowledge of all is limited" (ib:49). ¿Cómo podemos ganar conocimiento?: "By witnessing the effects, operations, and expressions of these attributes. It is a law of our nature to argue from the effect to the cause, from the action to the agent, from the ends proposed and from the means of pursuing them, to the character and disposition of the being in whom we observe them". (ib.:50).

Rasgos culturales básicos de este programa unitarianista lo volvemos a encontrar en la cultura moral y tecnológica norteamericana, así como en el programa cultural de Herb Simon en la Inteligencia Artificial: "bounded rationality", "means-end analysis", "unity of mind".

En palabras del propio Simon: "The goal of my life has been understanding the unity of human mind" (Entrevista 30/11/90).

Allen Newell admitió que "existe un intento en CMU de hacer una síntesis en los distintos saberes" (Entrev. 17/12/90). Pero como escribió en un poema titulado : " How It All Got Put Together, A Story" dedicado a Simon en el 21 Simposio de CMU dedicado a Cognición, celebrado ese año en honor de su obra: " It is not all put together yet, And it never shall be, For that is the way of the world..." (1989: 445).

Este código ético descansa en dos pilares: la unidad de la mente humana, y la libertad humana. Por el primero se pretende resolver el clásico problema de la división de la mente y el cuerpo. Como declaró Herb Simon, con L.T. "solved the venerable mind/body problem explaining how a system composed of matter can have the properties of mind" (1990:190).. Por el segundo se mantiene abierta la vía de la innovación, al concebir contra el determinismo , que el propio científico es el que determina el alcance de sus trabajos.

Este socavamiento del determinismo calvinista por el arminianismo lo detecta también Daniel Bell al analizar la cultura norteamericana: "El calvinismo, aun en las primeras colonias norteamericanas, fue constantemente corroido a medida que nuevas doctrinas, como el arminianismo...trataron de reemplazar la predestinación absoluta por la elección condicional". (1977:65). La preocupación por la "decision making" sólo se puede explicar en un marco que posibilita dicha elección y donde las decisiones humanas cuentan.

Según este modelo, crear y descubrir se considera parte de una misma actividad intelectual. Ciencia, ingeniería y arte pueden verse como saberes equivalentes.

La discusión en la School of Computer Science sobre si esta disciplina es

ciencia, ingeniería u otra cosa carece de sentido para la mayoría de los entrevistados. Y tantas veces se ha planteado en las entrevistas, el entrevistado responde con que es ambas cosas, dejando al antropólogo europeo desconcertado.

La cultura norteamericana es parte de la cultura occidental y como tal participa de la visión científica del mundo. Pero, al mismo tiempo, esta arraigada la idea del "American Dream". La palabra "visionary" no tiene la connotación de charlatán en este contexto, sino que se valora como una característica positiva de líderes políticos o empresarios innovadores. Y la creencia en el "everything is possible" aún forma parte del "common sense" del ciudadano medio.

"Human Problem solving" no es sólo una mera técnica es un "ethos", un tipo humano ideal para el cual su forma de ver el mundo, conocer su esencia, se resume en poder construirlo, cambiarlo, transformarlo, operar con él. Para Newell, "Science is in the techniques". ("Desires and Diversions, 4-12-91). El criterio de validez no está puesto en los principios o la teoría que se aplica sino en los resultados que se obtienen. Por lo tanto es el "solving" lo que cuenta ante todo. Como afirmó Newell recientemente: "If a domain can't get beyond having just discover...that science is in fact in a preparadigmatic state. It's in a very early stage. My idea is that discoveries in physics, in chemistry, in biology all convert routinely into things you can do later, things that help you actually do things, like solutions ...build new devices, or build calculations that you can do."(ibid. 4-12-91).

La definición de qué es investigación para un pragmático como John Dewey es la siguiente: " La investigación es la transformación controlada y dirigida de una situación indeterminada en otra que es tan determinada en sus distinciones y relaciones constitutivas que convierte los elementos de la situación original en un todo unificado". (1950:123).

Esta búsqueda del saber aplicado, a la vista de un europeo, siempre ha aparecido como un rebajamiento de la ciencia, esto es, de lo que se entiende por saber fundamental en la cultura universitaria modelada por la Sorbona, Oxford o Cambridge.

Pero si analizamos antropológicamente el tipo de saber constitutivo de la cultura norteamericana encontraremos que la tecnología no se concibe fundamentalmente como mera aplicación de la ciencia, sino "profession". El ingeniero, "the engineer", es ante todo un "professional", con todo el

significado que este término tiene en una cultura protestante como ya Max Weber indicó en su *Ética protestante y el espíritu del capitalismo* (1985). La tecnología es una ética de la "professional culture". El lema de Andrew Carnegie: "My heart is in the work" es una máxima moral. Igual que el "man of bounded rationality", o el "human problem solving" de Herbert Simon. Son valores culturales, no meros conceptos científicos.

El propio significado del término "rationality" en Estados Unidos es distinto al que definió el racionalismo ilustrado europeo. "Rationality" en América es un término aplicado a la conducta humana, en primer lugar. Esta es "rational" si esta orientada a fines o "goals" y existe una concordancia entre estos y los medios para conseguirlos. Va asociado, pues, a una teoría de la elección, "choice", y estos conceptos basan una teoría de la decisión, "decision making". En este contexto la creencia en Dios puede ser "rational" pues puede considerarse útil para conseguir determinados fines.

Por contra el Racionalismo en Europa inicialmente fue una doctrina filosófica, según la concepción iniciada con Descartes, por la cual la razón humana era autosuficiente para la explicación del orden natural. Mientras la cultura norteamericana funda la racionalidad en la libertad de elección humana, el racionalismo europeo funda la libertad humana en la razón autónoma.

Esta ética para autores como J.W Smith y A.Leland Jamison (1961) tiene una base religiosa. De hecho durante siglos la separación ciencia-religión en Estados Unidos ha sido superficial, a diferencia de la Ilustración europea. Este fenómeno se ha denominado "superficial accommodation" (J.W. Smith: 413). Según esta teoría, el ajustamiento puritano a la ciencia newtoniana fue en cuanto al cuerpo empírico no en cuanto al espíritu metodológico de la ciencia. "The Puritan theologians never themselves seriously adopted the method of science, and never ceased to approach the problems of physics with a moralizing rather than an objective and inquiring spirit". (:414).

Margared Mead detectó este carácter eminentemente "moral" de la cultura norteamericana: "I have emphasized that we are a moral people; a people who believe that there is direct connection between effort and efficiency and rewards on this earth; and that we, therefore, need success, because as failures we must also brand ourselves as evil in the sight of God and Man." (1942:261).

La separación europea, especialmente instaurada por Max Weber entre "juicios de hecho" y "juicios de valor", la doctrina de la "Wertfreiheit", es débil en la

cultura universitaria norteamericana. En ésta, un científico puede y acaba siendo un diseñador de "public policy", aunque le llame "policy analysis". Y la diseña siguiendo una metodología de "problem solving" con una marcada tendencia ingeniera.

Por eso D. Boorstin llama a la cultura norteamericana: "The Republic of Technology"(1978). No es la fuerza de la religión, o de la ciencia, sino la de la tecnología la aportación más específica de esta cultura al mundo moderno. Pero este carácter nacional, aún no se ha elaborado en profundidad.

2.3.9. LA CULTURA DE LA INNOVACION EN CMU: "THE RENAISSANCE MAN"

Los años 90s se han iniciado con el final de la Cold War y el planteamiento de un New World Order por parte de la Administracion Bush. Estados Unidos, cultura que ha propuesto dicho nuevo marco internacional, se encuentra una crisis interna que afecta tanto a la economía, como a la educación, la sanidad y el conjunto de sus valores. Los estudios antropológicos sobre la cultura norteamericana son escasos. Se da la paradoja que en el país con más antropólogos del mundo (30.000 en 1990), sigue siendo válida la afirmación de Clyde Kluckhohn hecha hace 40 años: "Sobre la cultura de este país, en el sentido antropológico, conocemos menos que sobre la cultura esquimal" (1983:244).

La obra "And keep your power dry" (1942) de Margaret Mead fue de los primeros intentos para profundizar acerca del llamado caracter nacional. Durante la II Guerra Mundial, Mead, Bateson, Benedict, Kluckhohn y otros antropólogos de primera fila trabajaron para el Foreign Morale Analysis Division, una sección de la Office of War Information. Sus estudios realizados sobre el "national character" puede volver a ser útiles de nuevo ante la conflictiva situación de crisis nacional que se vislumbra.

El principio constitutivo de Norteamérica para Margaret Mead es el carácter puritano : "The essence of the Puritan character, the character which has reached its most complete development in America, is the mixture of practicality and faith in the power of God--or moral purpose" (1943:161).

M. Mead considera el lema del puritano O. Cromwell: " Trust in God, my boys, and keep your power dry", (1943:161) como la clave de la moral norteamericana. Esta confianza en el poder de Dios y a su vez, en el trabajo duro y en el esfuerzo humano, sería el resumen de esta ética.

Ya Max Weber había elaborado esta ética puritana referida al desarrollo del capitalismo. Robert Merton, siguiendo a Weber, estudió como esta concepción es la base del sistema de conocimiento científico-técnico propio del presente sistema cultural. La predestinación calvinista ("Trust in God") habría alentado la fe en el orden natural, y en la ciencia; y el esfuerzo humano ("keep your power dry") sería la base del esfuerzo tecnológico. El sistema Science and Technology partiría de esta base, y su conclusión sería el "innovation cycle": R&D, cuya última expresión es el Computer Science and Technology.

Siguiendo este modelo, la ciencia tendría como fin "la gloria del Creador", y la tecnología perseguiría "el alivio del estado del hombre", en palabras de Francis Bacon en su *Advancement of Learning*.

Pero estos autores no se detienen a explicar el hecho sorprendente de cómo una religión como la puritana, y en especial el calvinismo, con su dogma de la predestinación y pesimismo sobre la condición humana, puede ser la base de una cultura como la norteamericana que ha empezado a hacer de la libertad individual su principal lema, del inventor tecnológico una figura nacional y de su "problem solving mentality" una forma de pensar ampliamente aceptada, y que incluso, como indica Kluckhohn, aprecia el buen humor y el carácter optimista como rasgos del carácter nacional.

La ascesis del protestantismo reformado explica una filosofía de la acción ("ascesis"), pero no la idea de la responsabilidad de individuo ante su propio futuro. El calvinismo explica el individualismo, pero no el individuo libre. El protestantismo reformado explica la ética del trabajo, pero no el trabajo inventor, innovador. El modelo del empresario calvinista weberiano no explica el "entrepreneur" de Schumpeter, el empresario innovador norteamericano. Y menos aún el "entrepreneur" político como son los técnicos y militares de DARPA. Para profundizar en la figura del inventor norteamericano, de la relación del trabajo como ética con la invención y la innovación vemos preciso analizar no sólo el protestantismo calvinista sino la influencia del unitarismo norteamericano, secta de origen cristiano, que, aunque minoritaria, cobró una importancia decisiva en la forja del carácter norteamericano.

El rasgo cultural que nos parece clave para explicar esta cultura de innovación, propia de Norteamérica, y en particular de CMU, es la idea del "free will" tal como la encontramos en la religión racional del unitarianismo norteamericano. Esta religión entronca curiosamente con el Renacimiento humanista europeo, a

través de uno de los fundadores del unitarismo, el científico de origen valenciano Miguel Servet, y su doble polémica contra Calvino y contra la Iglesia Católica.

Este humanismo renacentista se alimenta de los ideales de la Paideia griega (Werner Jaeger), y del concepto de educación de los sofistas atenienses. El más importante de ellos, Protágoras, fue el primero en expresar la máxima: "El hombre es la medida de todas las cosas", máxima que curiosamente repite Herb Simon al final de una de las reuniones fundacionales del Machine Learning: "Gaining a deeper understanding of human learning will continue to provide important clues about what to imitate and what to avoid in machine learning programs...Here as elsewhere, Man seems to be the measure of all things" (1983:36).

En varias entrevistas hemos podido observar a ingenieros de informática como Rich Stern, expertos músicos a la vez, o ingenieros como Granger Morgan con una profunda preocupación ética y política. En una entrevista con Herb Simon tenida el 30 de Noviembre de 1990, éste afirmó que " the model of man of CMU is Leonardo da Vinci". Según él, este campus con su interdisciplinariedad entre diversos saberes, artes, ciencias, ingeniería y humanidades se había inspirado en este tipo humano polivalente del Renacimiento, una época en que tecnología y humanidades podían convivir en una misma persona, que destacaba a su vez en la construcción de ingenios militares.

Esta figura no encuentra explicación en un país si nos ajustamos estrictamente al modelo weberiano, o incluso al modelo de Robert Merton sobre "ethos" de la ciencia. El patrón de la ciencia no es un "model of man" sino un orden de la naturaleza al que se ha de adaptar el "model of man". El sistema de conocimiento es "Science and Technology", primero entendemos el orden natural mediante la ciencia y luego lo transformamos mediante la tecnología.

Pero en CMU, como antes en el M.I.T., el orden de conocimiento está invertido. Primero fue un instituto tecnológico y la ciencia sirve en ellos a la tecnología. Este hincapié en el orden artificial es lo que curiosamente lleva a introducir las humanidades en el marco de conocimiento pero no ya no en el sentido antiguo sino unas humanidades que tienden a estar ligadas a problemas prácticos, esto es, "ingenieras", informatizadas.

Poner el "computer" en lugar de la "nature" como objeto central de estudio en un campus invierte el orden tradicional del conocimiento de la presente cultura.

Del ordenador se pasa a estudiar el artífice del mismo, el "human problem solver" y su sistema de saber "el problem solving".

De la ingeniería del ordenador se pasó a Inteligencia Artificial, de ésta a la "cognitive engineering", y posiblemente a la ingeniería social o cultural, en suma, a un tipo de saberes basados en una común base metodológica, el diseño, que se puede reemplazar los métodos inductivos-deductivos de la ciencia como sistema de conocimiento de las culturas tecnológicas.

2.4. CAMBIOS EN LA A.I. DE CMU.

2.4.1. EDWARD FEIGENBAUM Y EL "KNOWLEDGE ENGINEERING": O COMO SE RENUEVA LA A.I..

El programa de investigación de Newell y Simon empezó como hemos visto con la invención del Logic Theorist, que era sustancialmente un razonador lógico mediante procedimientos de búsqueda heurística. Pero este programa no tenía conocimiento del mundo empírico. El siguiente paso significativo de la Inteligencia Artificial lo daría un discípulo de Herb Simon, Edward A. Feigenbaum.

Su programa DENDRAL (1965-1983) construido en la Stanford University era un sistema experto que permitía inducir hipótesis de estructuras químicas a partir de los datos espectrales y fórmulas químicas proporcionados por los quimicofísicos.

El sistema experto es un programa de ordenador que puede proporcionar realizar funciones que requieran un conocimiento especializado como el propio del médico, del ingeniero o del arquitecto. Está compuesto por dos partes fundamentales: la base de conocimientos y el procedimiento de inferencia.

La primera a su vez consistía de dos partes: en la primera, el knowledge engineer incluía los hechos de la área o dominio en cuestión, organizados en forma de datos empíricos y teorías, el material que componen las llamadas bases de datos, y en la segunda se introducían lo que Feigenbaum denomina el "saber heurístico", "el saber de la práctica y el enjuiciamiento correctos. Es el saber experimental, el 'arte de hacer suposiciones acertadas', que un especialista humano adquiere después de años de dedicación". (1983:95).

Una aportación clave de Feigenbaum fue separar en un sistema inteligente lo que denominan base de conocimientos ("knowledge base") del mecanismo de inferencia ("inference engine"). Como afirma Bruce Buchanan, uno de los pioneros con

Feigenbaum de la

"knowledge engineering": "This has become the fundamental organizing principle of all successful work on expert systems".(Dec.1985:2).

Ello permitía centrar la búsqueda de Inteligencia Artificial en lo que denominan "knowledge". Dentro de él, el "saber heurístico" era para Feigenbaum, "lo mas difícil de conseguir, porque los especialistas--o cualquier otra persona--raramente se dan cuenta en que consiste". (1983:95). Según D. Lenat, el método seguido por Feigenbaum, Lederberg y Buchanan para construir DENDRAL fue ese: " "What they did ...was to talk to chemists and find out heuristics, rules of thumb, that led them to solve problems...And gradually, by talking with them, they got rules of chemical topology, mass spectography, first principles in chemistry, nuclear magnetic resonance, and so forth" (1989:37).

La antropóloga D. Forsythe ha destacado el hecho que la labor del "knowledge engineer" es similar a la del antropólogo al realizar el trabajo de campo y extraer del informante, hablando con él, las pautas culturales de su comunidad, el conocimiento compartido de la misma.

En cualquier caso, este saber llamado "heurístico" exige una investigación antropológica mas detenida en el futuro, dado que parece ser clave en el origen de la A.I. y en su desarrollo.

Tradicionalmente, la ciencia occidental ha admitido dos métodos para la obtención de conocimiento verificable, el que provenía de los sentidos, que se traducía en hechos empíricos de los que se inducían leyes generales, y el que provenía de la razón, que por procedimientos lógicos derivaba consecuencias particulares. Una combinación de racionalismo y empirismo ha sido la que ha caracterizado la ciencia moderna.

Pero la "heurística" como saber de la práctica no se ha considerado tema de estudio, dado que entra en el terreno que Judea Pearl (1984) denomina " rules of thumb, educated guesses, intuitive judgments o simply common sense", terreno que no es facilmente formalizable. Quizá se aclararía haciendo una investigación específica de caracter antropológico sobre su naturaleza, dado que al parecer es fundamental para construir sistemas expertos y como veremos más adelante, la ausencia de "common sense ", es decir, la ausencia de buenas heurísticas, limitan su desarrollo.

¿Qué aportó Feigenbaum?. Ante todo la creación de una nueva disciplina en la A.I.: la "knowledge engineering" o ingeniería del conocimiento. Fue el primero

que intentó plantear claramente que la A.I. debía dirigirse a finalidades prácticas y resolver problemas del mundo real.

Su principio, tomado del dictum baconiniano, "Knowledge is power", permitió conectar la A.I. con el mundo industrial y hacerla operativa, sacándola del mundo meramente académico, en el que se había quedado, después de su primer surgimiento en laboratorios como la Rand Co..

Nos interesa la figura de Feigenbaum por su papel innovador en la cultura de la A.I. y en especial, nos interesa en relación con el programa cultural de Newell y Simon. El "knowledge engineering" se saltó un principio del programa de Newell y Simon según el cual la A.I. avanzaba primero gracias al descubrimiento de principios generales de cómo funciona la mente humana, principios que luego se aplican en la A.I. como ingeniería. Feigenbaum empezó por hacer ingeniería, como habían hecho Newell, Simon y Shaw en 1955.

Durante años CMU no consideró trabajar en sistemas expertos y aún hoy en este campo es un centro atrasado en comparación con Stanford. No obstante, Feigenbaum en 1989, en un seminario en honor a Herbert Simon, llamaba a su antiguo maestro: "the first Knowledge Engineer" (1989:173).

La aportación de Feigenbaum continuó la línea de investigación originariamente iniciada por Newell y Simon con el L.T., pero detenida más tarde al anteponer la investigación psicológica sobre los principios generales de la resolución de problemas en la mente humana. La A.I. no avanzaba por la ciencia cognitiva sino por un nuevo salto adelante en el diseño de sistemas informáticos, que se valían de conocimientos científicos. No como se preconizaba tras un avance científico fundamental sino por una línea de hacer más compleja la teoría tecnológica, abriendo nuevos campos de ingeniería.

Los sistemas expertos se unían a la ciencia, pero de otra forma: convirtiéndola en tecnología. Así, Feigenbaum logró para construir DENDRAL la colaboración de Josuah Lederberg, profesor de genética y también premio Nobel como Herbert Simon. Mientras Simon, tras inventar el L.T. se había instalado en la psicología como científico cognitivo, el profesor J. Lederberg pasaría de profesor de genética a ingeniero del conocimiento.

La A.I. nació con Simon y Newell antes que fueran psicólogos o que la ciencia cognitiva se hubiera formado. En realidad nació al convertirse en ingenieros de su propio conocimiento para así introducirlo a la máquina, y poder hacer Inteligencia Artificial. Y la comprensión de cómo inventar un sistema

inteligente la que dió luces de lo que sería una sicología de la mente humana, entendida por tanto como "human problem solving".

El ciclo de innovación en la A.I. es primero tecnología, segundo, estudio empírico de los sistemas inventados , y tercero, producir una tecnología más compleja. Y esta tecnología empezó con ingenieros del conocimiento humano. Y se renovó con Feigenbaum, Lederberg y Buchanan, por la ingeniería, de nuevo.

2.4.2. SPEECH UNDERSTANDING: CAMBIOS EN EL PROGRAMA DE INVESTIGACION EN LA AI DE CMU.

CMU no inició la nueva línea de "expert systems", pero esta orientación hacia una A.I. aplicada influyó en un cambio en la trayectoria de la A.I. en CMU, reforzando lentamente su cariz pragmático e ingeniero. El cambio se empezó a introducir a principios de los 70s con el programa denominado Speech Understanding Systems.

El profesor Raj Redd, proveniente de Stanford y en la actualidad unos de los líderes fundamentales de la School junto con Newell y Simon, fue pionero en dicha área. El y su equipo consiguió inventar los primeros sistemas de comprensión del lenguaje hablado continuo, el HARPY y el HEARSAY II. Estos sistemas fueron el resultado de una iniciativa tomada por un grupo de científicos encabezado por Allen Newell que en 1971 recomendaron a ARPA el comienzo de un proyecto de cinco años "towards the demonstration of a large-vocabulary connected speech understanding". (SUS 1976-77:1). Se fijaron requerimientos concretos: "The system was required to accept connected speech from many speakers based on a 1000 word vocabulary task-oriented grammar, within a constrained task". (ibid.:1). El sistema tenía que cometer menos del 10% de errores.

En una competencia nacional, HARPY y HERSAY-II desarrollados en CMU consiguieron la mejor realización, al final de los 5 años.

El profesor Reddy, miembro de la National Academy of Engineering, obtuvo su grado de ingeniería en el Instituto Tecnológico de Madras en la India. Se formó en Stanford y su programa de investigación desde el principio siguió una orientación marcadamente ingeniera en A.I.: diseñar un sistema tecnológico nuevo, según las especificaciones requeridas, en los plazos previstos y sin salirse del presupuesto. El problema a resolver: el "speech problem", conseguir que la máquina entienda el lenguaje hablado continuo, una barrera que dificulta

el uso masivo del ordenador más allá de su uso por los expertos.

La comprensión del lenguaje hablado por la máquina se inició diferenciando diversas funciones que ésta debía cumplir: 1. Verificar que las sentencias oídas tienen sentido y son plausibles. Lo que requiere conocimiento semántico ("semantic knowledge"). 2. Comprender el contexto del discurso e interpretar las sentencias que reconoce de acuerdo con dicho contexto. Lo cual requiere conocimiento pragmático ("pragmatic knowledge").

Para realizar estas funciones la máquina debía realizar varias acciones, las dos más significativas: 1. buscar una construcción plausible de la frase hablada, y 2. realizar una hipótesis de que intentaba decir que el interlocutor cuando no podía reconocer la frase en cuestión, en suma, interpretar la intención de una frase determinada. Ello permitió que este proyecto sirviera a su vez para avanzar los "knowledge-intensive systems".

Ello se consiguió mediante el diseño de un sistema que comprendía múltiples "fuentes de conocimiento" ("knowledge sources" o KS) distintas: semántica, pragmática, sintáctica, léxica, fonémica, fonética, y otras. Todas estas fuentes comunicaban entre sí en una base de datos global o "blackboard". El dispositivo funcionaba de la siguiente forma: una vez que una frase era reconocida, la fuente de conocimiento que es relevante empezaba a operar a fin de encontrar la mejor interpretación lo más rápidamente posible, reduciendo el número de hipótesis incorrectas que eran generadas y probadas.

Para ello se utilizaba una nueva heurística inventada, una estrategia para dirigir la búsqueda denominada "beam search", por la cual en el gráfico de búsqueda, se descartaban todas las alternativas generadas alrededor de mejor camino en cada punto de decisión, a excepción de un haz ("beam") de ellas, precisamente las más débiles ("weakest"), y por ello, las que tenían mayor potencial de mejora. Esta técnica según Reddy fue clave para hacer de HARPY, el mejor sistema de comprensión del habla humana continua en su época.

Nótese que el método "hypothesize-and-test" utilizado aquí para que cada fuente de conocimiento interpretara la frase, es un clásico método de la ciencia experimental, pero con una salvedad, aquí está utilizado como un instrumento del método de diseño que sirve para construir el conjunto del sistema, de forma análoga a como Simon utilizó la lógica de Russell para diseñar el L.T.. Se diseñó un sistema de comprensión del habla que formulaba hipótesis para interpretar dichos mensajes. Igualmente, la búsqueda heurística es un

instrumento al servicio del diseño del sistema. La tecnología se ha servido de la ciencia experimental para inventar.

Este proyecto fue planteado según una forma de trabajo propia del ingeniero: "The success of the speech understanding research effort is all the more interesting because it is one of the few examples in AI research of a five year prediction that was in fact realized on time and within budget". (op. cit.:8).

Este programa tras más de 20 años de investigación aún no ha conseguido sus objetivos finales como el propio Reddy ha admitido: "Recognition of unrestricted spontaneous continuous speech appears unsolvable at the present" (1988:10), pero sistemas de reconocimiento de voz más sencillos están ya en el mercado. Una de las mayores dificultades es la comprensión de la ambigüedad del lenguaje hablado continuo. Lo que se han conseguido han sido sistemas de reconocimiento del habla como SPINX, que usan alrededor de 1000 palabras y tiene un nivel de acierto del 94%.

Dada estas dificultades, el Speech Understanding Systems, fue una de las primeras áreas que se interesó en cómo utilizar conocimiento incompleto, impreciso y parcial para resolver problemas, el mismo tipo de conocimiento que se pretende ahora conseguir en la denominada segunda generación de sistemas expertos.

La importancia del Speech Understanding Systems en la cultura de CMU fue el permitir un nuevo avance a la A.I. en una línea de construcción de sistemas que manifestaban una conducta inteligente más compleja. Ya no sólo razonaban según búsqueda heurística como el LT. Ahora empezaban a entender el lenguaje hablado continuo.

La A.I. de CMU ya no era sólo el programa de investigación de Newell y Simon. Esta orientación más ingeniera y estratégica de Raj Reddy, y en parte de Newell, fue acelerada a principios de los 80s, por una iniciativa de DARPA, la Strategic Computing Initiative (SCI).

2.4.3. LA "STRATEGIC COMPUTING INITIATIVE" DE DARPA Y EL ROL ESTRATEGICO DE CMU.

CMU, y en especial, su área de Inteligencia Artificial, ocupa un lugar estratégico dentro de la cultura de "computer science" en Estados Unidos. Por estratégico hay que entender un lugar decisivo para mantener la superioridad tecnológica norteamericana, y su primera aplicación, la superioridad militar.

Desde los años 80s, a la tradicional competición con la URSS, USA vió nacer una creciente competición tecnológica con un nuevo adversario: Japón, y en segundo lugar, la C.E.E. Una línea decisiva de esta competición fue la construcción de una nueva generación de ordenadores, los ordenadores inteligentes o "Machine Intelligence Technology".

La inmensa mayoría de ordenadores existentes hoy en el mundo no contienen inteligencia sino tan sólo memoria. Almacenan gran cantidades de datos e información, que son capaces de procesar en forma de cálculos matemáticos o recuperación de dicha información para tareas como tratamiento de textos o llevar la contabilidad.

Incluso las redes informáticas, que tanto auge han tenido en los 80s, son redes no inteligentes. Sus servicios más usuales son el correo electrónico y la transferencia de archivos. La introducción de gráficos o imágenes en los actuales ordenadores tampoco los hacen inteligentes. El actual boom del multimedia, acompañado una tecnología óptica como el CD-ROM que permite acumular enormes memorias de datos e imágenes, es la más reciente tecnología de un modelo de informática basada en el mero almacenamiento de información que ha dominado durante los primeros 25 años, y que ya resulta obsoleto. Su modelo empresarial ha sido el de la IBM y "los siete enanitos", el tipo de empresa que simbolizan este tipo de informática que ha encontrado en el Silicon Valley su Meca.

Desde los 80s se está abriendo paso un nuevo modelo de informática, denominado en Estados Unidos, "Machine Intelligence Technology", y en Japón, "Fifth Generation Computer Systems", basados en la generación de conocimiento y en una mayor potencia de procesamiento de información.

Este nuevo tipo de ordenadores no sólo almacenan información, en las tradicionales bases de datos, sino contienen bases de conocimientos que almacenan también un conjunto de heurísticas y reglas de inferencia que le permiten generar nuevo conocimiento experto. Según Edward Feigenbaum, padre de los sistemas expertos: "Estos ordenadores podrán aprender, asociar, hacer deducciones, tomar decisiones y comportarse en general de un modo que siempre hemos considerado coto exclusivo de la razón humana". (1983:21). Las empresas creadoras de sistemas expertos como Carnegie Group, DEC parcialmente, u otras basadas en ordenadores que procesan en paralelo como Thinking Machines representan este nuevo modelo de informática incipiente. Determinadas redes dentro de Internet son el lugar de trabajo habitual de los grupos de

investigadores del Strategic Computing, siguiendo el modelo iniciado en los 70s por Lynn Coway y Carver Mead, de "red de mentes". (Feigenbaum, 1983:1963). En 1982, Japón se adelantó a Estados Unidos lanzando la iniciativa nacional del Fifth Generation Computer Project, fundando su Instituto de Tecnología de los Ordenadores de Nueva Generación (ICOT), proyecto centrado, según su director Kazuhiro Fuchi en la "parallel inference machine " (PIM) o la tecnología de la inferencia simbólica procesada por computadores en paralelo. (1991: 22)

Esta iniciativa fue promovida por el MITI, el Ministerio japonés de industria, contando con el apoyo de las principales empresas japonesas de ordenadores, que han concentrado una selección sus mejores cerebros durante 10 años en dicho laboratorio nacional, y con un presupuesto de 450 millones de dólares.

Un año más tarde, en octubre de 1983, DARPA, influenciado por esta iniciativa, lanzaba su Strategic Computing Program. Según el documento oficial del DARPA: "The overall goal of the program is to provide the United States with a broad base of machine intelligence technology that will greatly increase our national security and economic power". (Strategic Computing Initiative.1983:10).

Este programa ha quedado oscurecido durante años por el macro programa Strategic Defense Initiative (SDI), popularmente concido como "Guerra de las Galaxias". Esta iniciativa la puso en marcha el Presidente Reagan en marzo de 1983, inicialmente con un mega presupuesto de 26.000 millones de dólares para los cinco primeros años.

La tecnología principal que este programa se proponía producir era una tercera generación de ingenios nucleares, ante todo, rayos láser y rayos de partículas subatómicas, y en segundo lugar, una generación de proyectiles inteligentes, relativamente lentos.

En medios informáticos, se consideró que el mayor problema técnico de este programa era la enorme complejidad del software necesario para dirigir toda la eventual defensa militar. Pero la SDI no era esencialmente un programa de investigación en informática, y menos en informática estratégica, ni contó con el apoyo de la comunidad de "computer scientists". Estaba pensado globalmente dentro de la tecnología nuclear para competir militarmente con la URSS ante un eventual ataque atómico. Ha sido el ultimo proyecto del periodo denominado Guerra Fría.

El Strategic Computing de DARPA, por contra, fue pensado teniendo en cuenta, no a la URSS, sino a Japón como competidor tecnológico y económico, y su proposito

ha sido crear una "new-generation computing technology".

Según este plan, el rasgo diferenciador de la nueva generación de ordenadores sería su capacidad de exhibir inteligencia como los humanos: "In contrast with previous computers, the new generation will exhibit human-like, 'intelligent' capabilities for planning and reasoning. The computers will also have capabilities that enable direct, natural interactions with their users and their environments as, for example, through vision and speech". (op. cit.: 1).

Raj Reddy, líder de la investigación en Speech Understanding en CMU denomina a estas máquinas: "3-G machines", ordenadores con un poder de procesamiento de un gigaops (mil millones de instrucciones) por segundo, con memoria primaria de un gigabyte y una anchura de banda de otro gigabyte al menos. En suma, un superordenador del tamaño de una estación de trabajo. Esta nueva generación de ordenadores permitirá la creación de redes informáticas nacionales e internacionales para transmitir y recibir tanto información escrita como imágenes de alta definición, sonido y voz, en suma, "Speech, Image, Language, and Knowledge, SILK" (Raj Reddy, 1991:467).

Para Raj Reddy: "By the turn of the century, it appears possible that a low cost (e.g. costing less than \$1000) super computer could be accessible to every man, woman and child in the world. Using such a system, AI researchers should be able to create a personalized, intelligent assistant which would use voice and vision for man-machine communication, tolerate error and ambiguity in human interaction with machines, provide education and entertainment on a personalized basis, provide expert advice on day-to-day problems, make vast amounts of knowledge available in active form, and make ordinary mortals perform superhuman tasks leading to new discoveries and inventions at an unheard of rate. Believe me or not, such a system would help the illiterate farmer in Ethiopia as much as the scientist in U.S.A. or Japan...We would then witness a true revolution." (1988:AI Magazine:19).

El epicentro de esta nueva tecnología es la "Machine Intelligence Technology".

Dentro de ella, DARPA se ha planteado desarrollar los siguientes campos:

"Understanding Natural Language Expressions", "Information Fusion and Machine Learning", "Planning and Reasoning", "Vision and Visual Image Generation" y "Speech Recognition and Production". (1983:8). Pero no es ésta la única tecnología a desarrollar.

El plan preveía un avance en las arquitecturas en paralelo ("Computer

Architectures: Methods for exploiting concurrency in parallel systems"), así como en microelectrónica, ("graphics display/image generation" y "distributed communications").

La metodología del programa consistía en construir varios experimentos piloto y desde las aplicaciones desarrollar la tecnología: "building on a selected set of intelligent computing capabilities that are ripe for development in the near term...The objective is to evolve these capabilities into a broad base of new generation technology and to demonstrate specific applications of the new technology to solving a number of critical problems in Defense". (1983: 14). La aplicación debía proveer un efectivo tirón ("pull") para la generación de la nueva tecnología. Las tres aplicaciones que se proponían era un vehículo inteligente autónomo, un asistente inteligente de piloto, y un sistema inteligente de dirección de combate.

DARPA ha actuado de coordinador de la investigación entre universidades, industrias y gobierno: " Engineers and scientists from many disciplines must collaborate in new ways in an enterprise of very large scope. A framework must be created for the effective, adaptive planning of the discovery and development processes in this enterprise". (Ibid. :10).

Este modelo de innovación, empujado por la figura de innovadores o "entrepreneurs" con cargos públicos, es el que dió nacimiento al ordenador hace 25 años. Es un modelo distintos al estudiado por Schumpeter y su empresario innovador o "entrepreneur". La innovación no viene ahora desde abajo, fundamentalmente, sino top-down, de arriba abajo. El político o militar innovador, usuario de tecnología, junto con profesores universitarios tecnólogos, con visión estratégica y cargos de "advisor" en dichos comités, deciden qué tecnología alentar para mantener su superioridad en ese campo. Raj Reddy en 1988 dirigía un comité de planificación estratégica patrocinado por DARPA para aconsejar al Presidente de Estados Unidos y al Congreso a cerca del futuro desarrollo y aplicación de la "computer science and technology".

DARPA inició con el Strategic Computing en 1983 la investigación en esta " Machine Intelligent Technology" con un presupuesto de 600 millones de dólares para los 5 primeros años. La Information Processing Techniques Office (IPTO) era la oficina responsable. El Programa preveyó la creación de 10 "new 'computing-technology communities' and another five to ten 'applications communities'". (ibid: 63). Cada una de ellas se componía de al menos 100

personas. El plan se planificó con una duración entre 8 y 10 años, y un costo de 150 millones de dólares al año.

La redes electrónicas mantendrían esta comunidades separadas en comunicación: "Special and unique arrangements will be considered to establish an effective research community by leveraging existing computer tools and communications systems. Electronic mail and electronic bulletin boards are the simplest examples." (ibid: 62).

El papel de la universidad en todo este programa ha sido proporcionar las ideas más avanzadas en materia de inteligencia artificial: "There will be a selection of ideas on especially difficult topics from a set of several dozen leading contenders. The most advanced artificial intelligence ideas that seem ripe for developing will be exploited with heavy university involvement". (ibid. :64).

Este ha sido, y es, el papel solicitado a CMU como uno de los centros estratégicos, líderes en Inteligencia Artificial.

La aportación de CMU al Strategic Computing Program ha abarcado un conjunto de proyectos y líneas de investigación coordinados en distintos campos. En Artificial Inteligencia, hay fundamentalmente tres grandes proyectos dirigidos respectivamente por Raj Reddy, Takeo Kanade y Allen Newell: "SPOKEN LANGUAGE UNDERSTANDING", "IMAGE UNDERSTANDING" y "INTEGRATED ARCHITECTURES FOR INTELLIGENT SYSTEMS". Entre sus resultados están el NAVLAB, un vehículo autónomo desarrollado por el equipo dirigido por el profesor Kanade; el SPHINX, un sistema de reconocimiento de voz para ayuda de pilotos, con un margen de error del 6% de las frases, construido por Kaifu Lee y el equipo del profesor Reddy y por último, las tres arquitecturas inteligentes, SOAR, PRODIGY y THEO en construcción, y dirigidas por los profesores Newell, Carbonell y Mitchell, respectivamente.

En Programming System y en Computer Systems, la investigación relacionada con esta iniciativa estratégica se ha centrado en "Parallel Computing", y ha sido dirigida hasta 1991 por H.T. Kung y Rich Rashid, con los programas "NETWORK-BASE MULTICOMPUTERS" y "VERY LARGE SCALE OPERATING SYSTEMS" y cuyos resultados han

sido los ordenadores , Warp, iWarp y Nectar, y el sistema operativo Mach.

Este conjunto de proyectos, organizados alrededor de un sistema experimental específico ejercen entre sí una sinergia. Esta búsqueda de "total system effort" y el organizar los proyectos en torno a un sistema experimental,son dos

características propias de la cultura de investigación de CMU : "

DARPA-supported research in the Carnegie-Mellon University environment tends to be organized around specific experimental systems that attempt to attain particular objectives, e.g., the development of a multi-processor system or the demonstration of an image understanding system...A major strength of the CMU environment is the synergy resulting from the close cooperation and interdependence of various research efforts despite their diverse foci".

(CMU-CS. Proposal 1980-83:1-3).

Dentro de estos programas estratégicos, el de Machine Intelligence, se ha desarrollado en CMU un nuevo campo de investigación denominado Machine Learning, uno de los campos más nuevos de la A.I., aunque ya va a cumplir una década.

2.5. EL MACHINE LEARNING DE CMU: HACIA UN "ENGINEERING APPROACH".

2.5.1. LA SEGUNDA GENERACION DE INVESTIGADORES DE ARTIFICIAL INTELLIGENCE DE CMU.

La Strategic Computing Initiative de DARPA fue promovida para responder, principalmente, al desafío de Japón. A ello contribuyó que la A.I. gracias a los sistemas expertos y los avances en otras áreas como comprensión del lenguaje hablado y de la imagen había avanzado lo suficiente para construir sistemas inteligentes operativos.

No obstante, ese proyecto se basa aún en lo que constituye la primera generación de sistemas operativos de la Inteligencia Artificial, encabezada por los "expert systems".

En 1987 había en Estados Unidos aproximadamente 1.500 sistemas expertos funcionando y varios miles en desarrollo. Según el Computer Science and Technology Board, las dos áreas de aplicación que tienden a dominar son la manufactura y las finanzas, donde los sistemas expertos sirven ayudando en la tarea configuración de la programación o de diagnóstico.(Dertouzos,1988:53).

No obstante, a pesar de sus reconocidos méritos, estos sistemas han recibido críticas de no ser capaces de representar el sentido común ("common sense") y no ser capaces de aprender. Contienen un conocimiento experto formalizado pero se ven incapaces aún de razonar con el sentido común de un niño y tener su capacidad de aprendizaje.

Una nueva generación de computer scientists, la segunda generación tras la de

Newell y Simon, y una intermedia representada por Feigenbaum, se han planteado la resolución de estos problemas. Esta segunda generación en términos culturales correspondería a lo que en Estados Unidos se denomina "babyboomers", o la generación nacida tras la II Guerra Mundial, entre 1945 y 1965, en la época de mayor expansión económica y militar de Estados Unidos. Esta generación está adquiriendo su madurez cuando se anuncia un periodo de declive de dicha supremacía, y cuando se impone un enfoque más pragmático en la resolución de problemas del "real world".

La A.I. de CMU, y en particular su programa sobre Machine Learning, encabezado por Jaime Carbonell y Tom Mitchell, y en el que también trabaja Allen Newell con su sistema SOAR, están tratando de diseñar máquinas que aprendan, en particular PRODIGY y THEO. El punto central de este programa de investigación para J. Carbonell y Pat Langley es: "to develop methods and techniques to automate the acquisition of new information, new skills, and new ways of organizing existing information". (Langley, Carbonell, CMU-CS-84-108).

El Computer Science and Technology Board, una organización del National Research Council que agrupa a científicos destacados del campo, denomina a esta segunda generación de AI: "deeper cognitive systems"(Dertouzos, 1988: 56). Y una de sus formas más avanzadas es el Machine Learning.

CMU fue pionera en este campo a principios de los años 80s. Las razones que llevaron a ello no fue un avance sustancial en la comprensión de cómo aprenden los seres humanos, sino la necesidad de superar los problemas que se habían detectado en los sistemas expertos: 1. su conocimiento era "brittle", frágil, superficial, y 2. su adquisición, el trabajo de extraerlo de la mente del experto, extremadamente costoso produciéndose en este punto un verdadero cuello de botella o "bottleneck".

En este sentido, el grupo de promotores de esta nueva tecnología del Machine Learning, J. Carbonell, R. Michalski y T. Mitchell escribían en 1983: "In response to the difficulties of encoding ever-increasing volumes of knowledge in modern AI systems, many researchers have recently turned their attention to machine learning as a means to overcome the knowledge acquisition bottleneck". (1983:1 CMU-CS-83-135). El programa de investigación del Machine Learning es pues una consecuencia directa del Knowledge Engineering de Feigenbaum, y trata en principio de resolver el principal problema que se encuentra la tecnología de los expert systems: el "knowledge acquisition", o la adquisición de

conocimientos por el sistema experto. En lugar de plantear su adquisición mediante la transmisión de conocimientos de un experto, se optará por proporcionar un conjunto de técnicas al sistema para que pueda aprender autónomamente.

2.5.2. J. CARBONELL, R. MICHALSKY, T. MITCHELL: "THE OBJECTIVES OF MACHINE LEARNING".

Para conocer este programa de investigación recurriremos a conocer los objetivos, las finalidades, los "goals" que se marcaron en 1983 los promotores del campo: Jaime Carbonell de Carnegie Mellon, Ryszard Michalski, de la University of Illinois at Urbana-Campaign y Tom Mitchell, entonces en la Rutgers University, que pasaría en 1986 a CMU.

Este nuevo programa de investigación, iniciado a principios de los 80s, se organizaba en torno a tres focos primarios de investigación: 1. "Task-Oriented Studies- the development and analysis of learning systems oriented towards solving a predetermined set of tasks (also known as the 'engineering approach')". 2. "Cognitive Simulation--the investigation and computer simulation of human learning processes (also known as the 'cognitive modelling approach')" y 3. "Theoretical Analysis--the theoretical exploration of the space of possible learning methods and algorithms independent of application domain".

(CMU-CS-83-135).

Obsérvese que los objetivos eran en primer lugar ingenieros ("task oriented studies"), en segundo lugar se planteaba el estudio empírico sobre cognición humana simulada mediante el ordenador ("computer simulation of human learning processes"), y en tercer lugar, un nuevo tipo de teoría dirigida a explorar no ya los actuales métodos de aprendizaje sino los posibles ("possible learning methods"), que, en cualquier caso, se planteaba tras la necesidad práctica ("practical necessity") de construir sistemas de aprendizaje aplicados ("applied learning systems").

Los autores partían de un problema marcado por una necesidad práctica, "a practical necessity", y una posibilidad, "a possibility".

1º. La necesidad práctica: "At present, instructing a computer or a computer-controlled robot to perform a task requires one to define a complete algorithm for that task, and then laboriously program the algorithm into a computer. These activities typically involve a tedious and time-consuming effort

by specially trained personnel. Present-day computer systems cannot truly learn to perform a task...". (Michalski, Carbonell, Mitchell, 1984:4).

2°. La posibilidad: "Machine learning research strives to open the possibility of instructing computers in such new ways, and thereby promises to ease the burden of hand-programming growing volumes of increasingly complex information into the computers of tomorrow. The rapid expansion of applications and availability of computers today makes this possibility even more attractive and desirable". (1984:4).

3°. La estrategia o vía para su solución, era el conjunto de focos de investigación que se planteaban abordar: a). "Task-Oriented Studies". b). "Cognitive Simulation". c) "Theoretical Analysis".

El primer foco participa de rasgos culturales propios de la investigación en CMU: Task-Oriented Studies, esto es, "the development and analysis of learning systems to improve performance in a predetermined set of tasks". Primero, "development", después, "analysis".

Podríamos analizar este planteamiento siguiendo el "problem solving" codificado por Newell y Simon, como "problem space" y compuesto de tres componentes: "desired state", "initial state" y "strategy" que transforma el estado inicial en el deseado.

Lo nuevo de este enfoque es que pretendía tener un enfoque orientado primariamente a resolver necesidades prácticas, al modo de la ingeniería.

En el primer número de la revista "Machine Learning", Herbert Simon en un artículo denominado "Why should Machines Learn?" (1984), objetaba excéptico contra la investigación en Machine Learning arguyendo que aunque tiene sentido, podríamos encontrar que tales programas no pudieran ser construídos: "The difficulty may be intrinsic in the task; human learning, though slow, may be close to optimally efficient...It may turn out that there aren't procedures more efficient than these very slow ones that human beings use" (1984:27-35).

Las prioridades de su investigación las dejaba explícitas: 1. "I would give a very high priority to research aimed at simulating, and thereby understanding, human learning...". 2., "I would give a high priority, also, to basic research aimed at understanding why human learning is so slow and inefficient, and correspondingly, at examining the possibility that machine learning schemes can be devised that will avoid, for machines as well as people, some of the tediousness of learning". (ibid:35). Por último, planteaba que su finalidad

prioritaria era en torno a programas que hacían descubrimientos : "my final priority is research on discovery programs--programs that discover new things". Campo ahora denominado Machine Discovery.(Valdez-Perez, 1990).

Analizando con detenimiento el texto vemos que se afirma como la más alta prioridad la investigación para comprender la mente humana, y al mismo tiempo se afirma como prioridad final la investigación en programas informáticos que realizan descubrimientos. Ello puede no ser una contradicción en el caso que se considere a la mente humana como un programa informático que hace también descubrimientos.

En cualquier caso esta visión escéptica sobre el Machine Learning contrasta con la de Carbonell, Michalsky y Mitchell. Mientras que para Simon la comprensión del aprendizaje humano es el objetivo principal, para esta nueva generación es un medio para desarrollar "applied learning systems".

A su vez, Machine Learning es también para estos autores un campo teórico, consideran "Machine Learning as a Science", pero en una acepción distinta a la de Simon: " An equally basic scientific objective of machine learning is the exploration of alternative learning mechanisms, including the discovery of different induction algorithms, the scope and limitations of certain methods, the information that must be available to the learner,...the creation of general techniques applicable in many tasks domains. There is no reason to believe that human learning methods are the only possible means of acquiring knowledge and skills. In fact, common sense suggests that human learning represents just one point in an uncharted space of possible learning methods...Most theoretical work in machine learning has centered on the creation, characterization and analysis of general learning methods, with the major emphasis on analyzing generality and performance rather than psychological plausibility" (1984:5).

Lo novedoso de este texto es que se utiliza el término "science" en una acepción totalmente distinta a la tradicional. La ciencia clásica, la física, la química, etc. no estudia partículas atómicas o estructuras moleculares "alternativas" o "posibles" sino las existentes. El principio de incertidumbre de Heisenberg no modifica la tarea de la ciencia sino que la hace más difícil al admitir que el observador modifica el campo observado. Por otro lado, la reciente física del caos, admite un principio no determinista en el estudio de la naturaleza. Pero es ciencia del Machine Learning persigue otro objetivo.

Lo que los autores de esta ciencia del Machine Learning pretenden es considerar

también científico la exploración de mecanismos alternativos de aprendizaje:

"Whereas theoretical analysis provides a means of exploring the space of possible learning methods, the task-oriented approach provides a vehicle to test and improve the performance of functional learning systems". (ibid:5). En esto sintonizan con un objetivo similar de toda la computer science. Como afirma el informe "The National Challenge in Computer Science and Technology": "Theoretical computer scientists seek to understand all possible architectures or algorithms, which computer scientists create themselves". (Dertouzos,1988:60). Esa ha sido la tarea de la tecnología: el diseño de mundos posibles, no de como son las cosas sino de como podrían ser, o "how they might be" (The Sciences of the Artificial, 1981:xi). La diferencia es que ahora esta empezando a entenderse esta actividad como un conocimiento teórico en sí mismo, no un mero quehacer o "doing". Y este nuevo conocimiento se le denomina con un término antiguo, el de "ciencia", que ya no corresponde a su anterior significado. Se está conservando un término para designar contenidos del todo distintos.

Pero este texto de Carbonell, Michalski y Mitchell contiene una tradicional visión que dificulta poner en marcha la anterior innovación, y explica en parte el por qué se conserva el término citado: identifica como hecho dado e inmutable el "human learning process", considera la invención de procesos de aprendizaje alternativos como un campo ajeno a los "human learning methods", es decir, considera que la innovación en métodos de aprendizaje ha de dejar fuera los métodos humanos, dado que considera a éstos en gran parte inmutables, y objeto de estudio para la ciencia cognitiva, no para la A.I.

En parte no les falta razón. Las técnicas de aprendizaje fundamentales, no únicas, de los estudiantes de este computer-intensive campus siguen siendo la "lecture" medieval dada por el profesor a sus alumnos, con algo más de participación de éstos, complementada con el correo electrónico. Lo cual explica que una mera comprensión de dichas formas de aprendizaje no sirvan de mucho para avanzar la A.I..

Pero el que estas técnicas de aprendizaje humano sean anacrónicas no significa que en el campo de los "human learning methods" no pueda innovarse con métodos ingenieros.

El problema se podría volver al revés: ¿Puede avanzar el Machine Learning si no se innova radicalmente en lo que podríamos llamar el Human Learning?. ¿Cómo puede innovarse en human learning?. ¿Es preciso primero entender científicamente

como aprenden los estudiantes para luego mejorar su aprendizaje , o más bien es al contrario, avanzando en el diseño de nuevas formas de aprendizaje humano se darán nuevas luces a la ciencia cognitiva y al Machine Learning?.

Si no se avanza en esta dirección lo que puede ocurrir es que Machine Learning no haga más que repetir métodos de aprendizaje ya conocidos adaptándolos a la máquina. Con lo cual su avance es problemático.

Veamos algunos ejemplos.

2.5.3. "BASIC RESEARCH" EN MACHINE LEARNING, PRODIGY PROJECT, THEO PROJECT, SOAR PROJECT: EL NUEVO "KNOW WHAT".

Las tres arquitecturas integradas que están en construcción en CMU son SOAR, dirigida por Allen Newell, PRODIGY, dirigida por Carbonell y THEO, dirigida por T. Mitchell. Estas tres arquitecturas forman el principal programa de "Basic Research in Computer Science" denominado: "Integrated Architectures for Intelligent Systems (1990-93), presentado a DARPA para su financiación. (CMU-SCS- Basic Research in CS, 1989).

Este programa esta presidido por una finalidad común : "One ultimate goal of A.I. research is to understand how to create an agent capable of general intelligent action in the world, with the same openness, flexibility and power as we see in educated humans" (1989: 7-1). A destacar en el texto la combinación entre la función de la ciencia, "understand", y la del arte, "to create", en suma, la combinación de la tecnología.

Los resultados de esta "basic research" serán según los autores los siguientes: "The basic scientific results of this research will be a technical understanding of what types of total system organizations are capable of integrated intelligent behavior, as well as an understanding of which aspects of the total system belong in the architecture. This include an understanding of what problems remain unsolved and what difficulties stand between the current generation of systems and the next generation of more effective systems". (ib.:6-1).

Es decir, los resultados científicos serán una mejor comprensión de la tecnología que estos mismos tecnólogos habrán inventado. Ya no se trata de sólo un nuevo "know-how", un aumento del saber-como-hacer sistemas tecnológicos. Es un nuevo "know-what", qué nuevos sistemas se pueden construir capaces de comportamiento inteligente. Y es por este nuevo conocimiento que son capaces de hacerlo por lo

que obtienen un nuevo saber-como o "know-how". No pretenden pues una mera "performatividad" (Lyotard) sino un conocimiento teórico de una nueva realidad, pero no ya dada, sino posible y que ellos mismos construyen.

"The direct products of this research are software systems of two kinds: integrated systems and software support. The integrated systems are the Prodigy, Soar and Theo systems. The software support are the programming languages, software systems and tools used to implement the Prodigy, Soar and Theo systems". (ibid.:6-1). Estos productos son los que se exportan y países que los compran creyendo que compra tecnología, cuando en realidad sólo se adquiere así la mitad del proceso, aquella que no se puede reproducir dado que no se tiene el conocimiento para ello. La clave de la tecnología está en el conocimiento como invención, en el saber que productos nuevos se pueden hacer, y cómo hacerlos. Y este conocimiento no puede vivir separado de organización, de actividad socialmente compartida, en suma, de cultura. Este nuevo conocimiento no son libros sólo, sino los investigadores que los hacen, los equipos que construyen el software, las agencias que los financian, los otros investigadores que utilizan sus productos, en suma, todo un sistema cultural tecnológico o tecnocultura, sin el cual no puede haber "conocimiento".

Para un investigador de la A.I., PRODIGY Y THEO son "integrated architectures" para un tecnoantropólogo son proyectos culturales. Por tanto los denominaremos PRODIGY PROJECT y THEO PROJECT.

Me detendré en estos dos últimos, dado que son las que están siendo construídas dentro de un enfoque centrado en el Machine Learning. SOAR es un sistema más antiguo, aunque quizá más maduro, que responde al programa inicialmente denominado Machine Intelligence.

2.5.5. " THE PRODIGY VIEW" Y LA "COGNITIVE ENGINEERING".

La orientación ingeniera antes analizado en el MACHINE LEARNING de J. Carbonell y T. Mitchell de CMU, inspira también el proyecto PRODIGY . Se evidencia en el papel "Designing an Integrated Architecture: The PRODIGY View" (Carbonell, et al. 1990) escrito en 1990 por todo el equipo compuesto en esas fechas por Jaime G. Carbonell, Yolanda Gil, Robert Joseph, Craig A. Knoblock, Steve Minton y Manuela Veloso. Este "paper" fue presentado a una conferencia de Cognitive Science, ese mismo año.

En el se plantea, en el resumen final, el siguiente conflicto: "PSYCHOLOGICAL

VALIDITY vs. COGNITIVE ENGINEERING". Los autores afirman: "The PRODIGY project strives to produce a useful, scalable, and maintainable reasoning and learning architecture. Where this matches human cognition, it is so by accident, by the limited imagination of the PRODIGY designers, or perhaps because the human mind has indeed optimized such aspects of cognition. In all others aspects, the goal is at reengineering cognition the way it ought to be, in order to be most useful in problem solving, planning, and learning". (1990:6).

Ese mismo año 1990, Jaime Carbonell daba una charla en Barcelona denominada "Artificial Intelligence: Past, Present & Future" (VIA-90)., en la que tras trazar un recorrido por la breve historia de la disciplina acababa con cuatro "basic messages": "1. AI is an unquestioned success at handcrafted solutions. 2. AI must move from a craft into an engineering discipline. 3. AI must live in the world of classical informatics. 4. AI must strive for standard software environment". (VIA-90).

PRODIGY, como sistema informático, y como proyecto cultural, es una concreción de esta filosofía. Según la propuesta de "basic research" antes citada, PRODIGY es : "a general problem solver combined with several learning modules".

(Ib.:7-3). El "problem solver" es un mecanismo que resuelve problemas en diferentes dominios de aplicación como planificación de la actividad de un robot, planificación logística y otros, mediante una búsqueda de alternativas en un "problem space". Recordemos que "problem space" es una forma de representar los problemas en la cultura de computer science de CMU, según la cual, estos pueden verse como un mapa en el que están representados el estado final o "desired state" y el inicial "initial state" y la resolución del mismo consiste en decidir cuál es un camino satisfactorio que lleve al robot o al agente informático a alcanzar el estado final a partir del inicial. Esta representación inspirada en como se representa una operación militar sobre el mapa parece adecuada para representar la actividad de sistemas informáticos.

Pero la novedad de PRODIGY es que a este mecanismo de resolución de problemas se le han añadido varios módulos de aprendizaje "learning modules", varias formas mediante las que el "agent" puede usar la información que recibe y usarla de dos formas, o bien para aprender "control rules" o reglas de control o normas que guían la acción general del agente o bien "domain knowledge" o conocimiento empírico sobre el dominio de actuación.

Estos módulos son de varios tipos. A diferencia de SOAR, que sólo utiliza un

tipo de aprendizaje denominado "chunking", PRODIGY y THEO utilizan distintos tipos. En concreto PRODIGY usa principalmente tres: 1. "Explanation-Based Learning" (EBL), 2. "Derivational Analogy", y 3. "Learning-by-Experimentation".

El primer módulo, EBL, desarrollado principalmente por Steve Minton, consiste en un método analítico ("analytical method") de aprendizaje de reglas de control o "control rules", que permite al sistema aprender de ejemplos al poder conceptualizarlos y explicar ("explain") su adecuación para tomar determinada decisión. El sistema no sólo resuelve problemas sino que puede explicar el por qué ha elegido este camino y no otro al disponer de un conocimiento sobre su propio sistema de resolución ("problem solver") y sobre el dominio específico de aplicación. En este sentido, el sistema puede aprender de su propia experiencia.

Un segundo módulo es el denominado "ANALOGY", desarrollado principalmente por Manuela Veloso. Este mecanismo de analogía se basa en el uso de problemas similares resueltos con anterioridad para resolver nuevos problemas. El "problem solver" graba las justificaciones para cada decisión tomada durante el proceso de búsqueda y son usadas para la reconstrucción de la solución de situaciones problemáticas análogas.

Por último, se está desarrollando ahora un nuevo mecanismo, "EXPERIMENT", principalmente por Yolanda Gil, que consiste en aprender nuevo conocimiento del dominio en cuestión o "domain knowledge" que sea incompleto o mal especificado. Para ello se está desarrollando un método denominado "operator refinement method" para activar una experimentación reactiva o "reactive experimentation" cuando el sistema obtiene consecuencias distintas a las esperadas de una acción. Este mecanismo sirve para adquirir conocimientos empíricos de área de actividad en cuestión. Se inspira en cómo adquiere conocimientos "real-world learners ranging from children at play to scientists at work. (Carbonell, Gil: 1987:1).

Lo curioso de este último módulo es que está instrumentalizado dentro de un sistema tecnológico de conjunto. El método experimental, propio de la ciencia, sirve aquí para diseñar una arquitectura informática integrada. La experimentación es un subconjunto del diseño como forma de aprender del conjunto del sistema.

Teóricamente, podría resultar interesante extraer un nuevo módulo de aprendizaje basado en cómo el equipo de PRODIGY ha aprendido a hacer PRODIGY. Este nuevo módulo podría llamarse "learning by design", o "learning by possibilities" dado que es la actividad en la que están todos ellos comprometidos, según hemos visto

en el papel que firmaron como equipo en 1990.

Bajo la dirección de Jaime Carbonell, distintos estudiantes de doctorado de la escuela han desarrollado cada uno de los módulos dando lugar a diferentes tesis doctorales en Computer Science. Así, Steve Minton tuvo una contribución destacada en el diseño de el EBL. Su tesis o "dissertation" se denominó: "Learning Effective Search Control Knowledge: An Explanation-Based Approach" (1988). Manuela Veloso, desarrollo su investigación sobre "Learning Analogies by analogy" (1989) y Yolanda Gil la está desarrollando ahora sobre "Learning-By-Experimentation".

La financiación de estas tesis ha corrido a cargo de contratos con DARPA y con una Ph.D. Scholarship de algunas empresas como AT&T Bell Laboratories.

2.5.6. EL PROYECTO THEO: "TOWARD A LEARNING ROBOT".

Tom Mitchell es, junto con Carbonell, uno de los fundadores de la disciplina Machine Learning.

Formado en Stanford y llegado a CMU en 1986, Mitchell es el responsable principal de THEO, una arquitectura inteligente integrada que también exhibe funciones de aprendizaje. Su principal aplicación ha sido servir de sistema de control de un robot móvil provisto de sensores que puede responder tanto a situaciones de rutina como reaccionar ante cambios del medio.

THEO usa varios "learning mechanisms", que usa en diferentes situaciones. Estos incluyen "chunking" también usado por SOAR, "Explanation-based learning" usado por PRODIGY y otros como "statistical learning" o "caching of slot values". (1989. Basic Research: 7-5).

La aplicación más directa de esta arquitectura ha sido en el área de Robotics. Theo está construido con una orientación central hacia esta área, dentro de la cual se encarga del mecanismo de control del robot.

Una de las áreas de esta investigación se desarrolla en el programa AUTONOMOUS PLANETARY ROVER, iniciado en 1987, en el Robotics Institute de CMU. El robot inteligente AMBLER ha sido una de sus primeras realizaciones. Este un robot que la NASA ha previsto para la exploración de Marte a comienzos del próximo siglo, ahora dentro del programa SPACE EXPLORATION INITIATIVE un programa estratégico para los próximos 20 años que el Presidente Bush anunció el 20 de julio de 1989, y que marca la nueva política de este país en materia espacial.

Este es un denominado "collaborative project", realizado mediante acuerdo con la

NASA, que abarca varios profesores de Computer Science , Robotics como Takeo Kanade, Reid Simmons y Red Wittacker de Carnegie Institute of Technology. Su construcción se ha realizado en el Robotics Institute de CMU, en concreto en el Field Robotics Center, y para su construcción se ha precisado de la colaboración de tres equipos distintos que han resultado los tres problemas de este tipo de robots: 1. "Perception", 2. "Cognition", y 3." Manipulation". (CMU-Robotics Ins-1987).

El paso clave de este proyecto ha sido el pasar de Machine Learning como mero mecanismo cognitivo a un sistema más complejo que incluye, no sólo cognición sino visión y manipulación. Ello ha llevado a Tom Mitchell a colaborar con un computer scientist, Takeo Kanade, especializado en Computer Vision,y con un ingeniero Red Whittaker experto en construir robots capaces de manipular objetos y desplazarse por el "real world". David Pahnos, director adjunto del Field Robotics Center, denomina este enfoque "integration, synthesis.", habiendo formado un "integration group" que se reúne semanalmente (Entrevistas, 20 Mayo 1991, 7 junio).

La actividad autónoma de aprendizaje en un mundo como el de Marte, ha requerido construir robots con visión, manipulación y desplazamiento. Y eso el mundo sólo de la A.I. sólo no lo puede proporcionar. Para Steve Shafer del Robotics Institute: "For decades, computer scientists have been studying toy problems such as compiling computer programs, playing chess, designing parallel computers, and finding convex hulls. Now, under the impetus of robotics, computer scientists are trying to represent the physical world in computer hardware and software..."(1990:295).

El hecho nuevo de este proyecto es que MACHINE LEARNING ha salido a la calle y el robot ha empezado a aprender del "real world".

Si Machine Learning es el área más compleja de la A.I., el proyecto de Learning Robot, es uno de los mas complejo de Machine Learning, dada la diversidad de campos para su construcción. El resultado es un Machine Learning que puede actuar con cierto grado de autonomía en el mundo real.

Lo curioso es que cuando el Machine Learning pasa a diseñar máquinas que aprenden se encuentran con los ingenieros que son los encargados de realizar la síntesis: el robot inteligente, siendo la inteligencia una subconjunto de un sistema mecánico, en suma, de una máquina. W. Whitaker ha sido el que ante la opinión pública ha quedado como el principal responsable de AMBLER. El New York

Times, Business, del 21 de abril de 1991, dedicaba un artículo al proyecto denominado, "The Creature That Lives in Pittsburgh", en el que W. Whittaker, aparecía en solitario junto al robot.

2.5.7. FREDERICK REIF : EL "HUMAN COGNITIVE ENGINEERING" EN EL CENTER FOR DESIGN OF EDUCATIONAL COMPUTING.

Otra línea de investigación que tiene relación con esta orientación de la A.I. hacia un "engineering approach" la desarrolla el profesor Frederick Reif en el Center for Design of Educational Computing, y la denomina: "Human Cognitive Engineering" (1980), o más recientemente "Artificial Natural Intelligence" (1991)

En 1980, Frederick Reif, en un congreso celebrado en CMU sobre "Problem Solving and Education" planteó que existía lo que él denominó un "gap" entre la A.I. y la Ciencia Cognitiva: mientras la primera era normativa y descriptiva, la segunda sólo era descriptiva. Para resolver el "gap" proponía lo que llamaba una "human cognitive engineering": "This neglected domain of prescriptive human cognitive engineering might be called 'human cognitive engineering' (or 'human knowledge engineering', if one adapts a term used by Feigenbaum to denote applied work in complex artificial intelligence). Human cognitive engineering would thus take into account the special capabilities and limitations of human cognitive functioning (e.g., highly developed language and pattern recognition capabilities, but significant memory limitations) in order to address questions such as the following: Can one design or invent modes of human information processing that can specifically enhance human cognitive performance...?"(1980-44). La premisa para ello es que, en efecto, dicha mejora es posible, lo que significa que el aprendizaje humano no está en su punto de casi optimización, como afirmaba antes Simon.

Esta "human cognitive engineering" se está utilizando para averiguar y guiar las formas de resolución de problemas de los estudiantes de física. Frederick Reif es físico de profesión y su interés educativo está centrado en como utilizar esta tecnología al servicio de mejorar el bajo nivel de los estudiantes de esta ciencia. Esta línea de investigación requerirá un estudio etnográfico con más detenimiento en el futuro, pues apunta en directo hacia la ingeniería cultural, entendiendo la educación como el método de implementar los diseños culturales que tienen los antropólogos para el cambio de las comunidades culturales, como ya

se evidenció en proyecto Vicos (Holmberg, 1956a: 17, "The role of education").

2.5.8. CAMBIOS EN LOS MODELOS EDUCATIVOS DE CMU.

Esta orientación de la A.I. hacia un "engineering approach" en cuanto método, y una dirección hacia el "learning", hacia el aprendizaje, la educación, tiene conexión con tendencias más profundas en Carnegie Mellon.

De hecho, cada avance de CMU, desde su fundación en 1900, se ha identificado con un avance del "engineering method" y la generación de un nuevo modelo de "education". Cada reforma ha significado una extensión del "design" y la "problem solving mentality" a nuevas áreas.

Andrew Carnegie en 1900 introdujo la idea de la "professional education" como alternativa a la educación universitaria tradicional por crearla separada del "real world", y elevando al "technical education" a rango universitario. Este modelo educativo se basaba en el método llamado "learning by doing", una aplicación directa del primitivo pragmatismo norteamericano.

Robert Doherty, en los años 30s, abordó una reforma educativa introduciendo elementos de una "liberal education" como complemento de la anterior: fue el denominado "Carnegie Plan", que sirvió de modelo educativo para una generación de ingenieros norteamericanos. A su vez codificó el "problem solving" como "engineering method". Líderes de la institución como el Provost Dunlap Smith, incluso fue más allá como hemos visto en la primera parte de la tesis tratando de hablar del "problem solving" como un "basic form of scholarly thought" para el conjunto disciplinas del campus.

Herbert Simon, discípulo de Dunlap Smith, continuaría este intento uniéndolo con el ordenador, los años 60s y 70s, introduciendo con Allen Newell el "human problem solving". Se realizó un congreso sobre "Problem Solving and Education" que culminó en los años 80s, con la red Andrew y la "computer science education", para el conjunto del profesorado y estudiantado. En ese periodo, Simon teorizó sobre la posibilidad de transformar el "design" ingeniero en una ciencia.

Al calor del Andrew Project, se plantearon diversos intentos en CMU por constituir el diseño en un "core curriculum" para todo el campus, intentos que no fructificaron. (Achenbaum, W.A. 1985).

Pero los años 90s, una nueva reforma se está planteándose no sólo en los campus sino en el conjunto de la sociedad norteamericana. El modelo de investigación y educación denominado Science and Technology está en crisis. Una orientación

hacia la ingeniería y la cultura de la innovación se discute de nuevo. Y CMU podría considerar el "design", como método de aprendizaje e investigación común. Una innovación en métodos educativos ayudaría a disciplinas como la A.I, y el Machine Learning a dar un nuevo salto adelante.

Esta fase coincide con un movimiento similar que ha comenzado en la "higher education" en Norteamérica en los principales campus como Stanford, Cornell o Michigan (Grassmuck, The Chronicle of Higer Education, Sept 12, 1990). Se considera la necesidad de una mayor "globalization" de los campus, una mayor atención a la diversidad étnica y cultural y una mayor atención a la educación de los "undergraduate".

Si ponemos estos periodos dentro de un contexto más amplio apreciaremos que coinciden a grandes rasgos con ciclos alternativos de prosperidad y crisis económico-social en Estados Unidos, cada uno de los cuales se ha resuelto con una extensión del mundo tecnológico ,en un sentido amplio, esto es, no sólo de tecnologías primarias como la electricidad o la informática sino de tecnologías secundarias, como el marketing, o la planificación estratégica.

La primera reforma de Andrew Carnegie, coincidiría con la Progressive Era, 1890-1920s, donde floreció la economía de las grandes corporaciones, basada en la tecnología de la electricidad, el automóvil y los métodos del trabajo en cadena. En política, coincidió con administraciones republicanas.

A este periodo, le siguió una etapa marcado por la Gran Depresión (1930s-1950s). En el cual, el anterior modelo industrial entró en crisis, y la administración del país pasó a manos de los demócratas. Ambas etapas conforman un primer ciclo económico-social.

En los años 60s, se inició un nuevo ciclo económico de gran prosperidad basado en la tecnología informática que ha hecho crisis a finales de los 80s, bajo la Administracion republicana. Ha sido en CMU el periodo dorado de la computer science, marcado por la Administracion Cyert y el liderazgo de Newell y Simon. Y en los 90s se ha abierto la etapa de recesión económica, y de crisis del actual modelo industrial, basado en la informática y las nuevas tecnologías cuyo futuro es incierto, al coincidir con el final de la Guerra Fría y una reestructuración internacional profunda.

Estos ciclos vienen a corresponder a los llamados "long waves cycles" analizados en los años 20s por el economista ruso Kondratieff (1984), y estudiados más tarde por Schumpeter(1939), Kutnetz (1953) , y Volcker(1978).

El programa de Andrew Carnegie y los primeros presidentes establecieron la figura del "professional engineer". La reforma de Doherty abrió paso al "professional manager". Se formó al ingeniero en técnicas de administración industrial y en la dirección de personas. Los años de Cyert han producido un "computer scientist", que ha ocupado en parte el papel del "professional engineer" en la era informática. Los años 90s pueden dar lugar al nacimiento del diseñador de organizaciones informáticas, ya a escala internacional. Para ello hará falta una nueva generación de tecnologías secundarias, o segunda generación de "high tech".

Si la primera generación se basó en la informática, la biotecnología, los nuevos materiales y otras la segunda generación de nuevas tecnologías puede partir de una "tecnoantropología", una "tecnosociología", una "tecnosicología"...

Tras la II Guerra Mundial, una de las aportaciones más novedosa en el campo de ciencia tradicional ha sido iniciar la construcción de un nuevo campo denominado "cognitive science". Es el campo más reciente de un sistema de conocimiento que empezó como ciencia de la naturaleza con la astronomía de Copernico y Kepler, la física de Galileo y Newton, continuó con la selección natural de Darwin, con las ciencias sociales de Adam Smith, Comte, Wundt y Taylor y ha culminado con las ciencias cognitivas. Sucesivamente Italia, Inglaterra, Francia, Alemania y tras la II Guerra Mundial Estados Unidos fueron los países que tuvieron el protagonismo en estos campos.

La ciencia cognitiva fue iniciada en la mayor época de esplendor científico y técnico de Estados Unidos, en los años 50s y 60s, y sigue ejerciendo en el un papel de liderazgo. Hasta ahora la historia más completa de la "revolucion cognitiva" es la escrita por Howard Gardner, profesor de sicología de Harvard. Es una historia que empieza en el programa de investigación de Platon en el Menon y considera por este orden las disciplinas fundamentales que la componen: filosofía, sicología, inteligencia artificial y lingüística (Gardner, 1987:53). Y secundariamente, antropología y neurociencia. Según este esquema, la creación de dicho campo habría estado presidido por el programa de la filosofía tradicional y de la ciencia.

Pero se da una anomalía, una curiosa ironía, en este caso: el modelo inicial de inspiración de la actual ciencia cognitiva, a diferencia de la iniciada con Platon, no ha sido el funcionamiento de la mente humana, sino el de los procesos cognitivos del ordenador. La ciencia cognitiva moderna ha nacido inspirada en el

Computer Science y la Inteligencia Artificial, no al revés. El ciclo de innovación en este campo es de la tecnología a la ciencia.

Como afirmaba el psicólogo George Miller, uno de los fundadores de la psicología cognitiva, en un seminario celebrado en CMU en homenaje a Herbert Simon: "The irony that it took a machine to arouse psychologists to an active interest in mental processes has been frequently noted. Cognitive psychology broke free from behavioristic psychology in the 1950s largely due to the growing recognition that mental phenomena can be analyzed using concepts that were developed to analyze electronic communication and control systems. In part, this realization was the result of the development of servomechanisms, which played a central role in cybernetics theories, and in part a result of increasing mathematical sophistication, thanks to mathematical theories of information games, and learning, but

mostly it was consequence of increased familiarity with the new computers and with the kinds of information processing of which they are capable". (1989:146).

El propio H. Gardner admite la "inspiración proveniente de las computadoras".

Según este historiador de la ciencia cognitiva: "Los estudios cognitivos fueron legitimados gracias al advenimiento de las computadoras (que en sí mismas representaban una conducta de resolución de problemas) y al auge de la teoría de la información (que suministró una base objetiva para establecer los elementos componentes del lenguaje o de los conceptos)". (1987:137).

La A.I. dentro de la ciencia cognitiva ha jugado un papel de inspiración. Si la ciencia cognitiva avanza por la A.I., y ésta avanza por la "knowledge engineering", podemos suponer que un nuevo avance de esta última disciplina provendrá no de una comprensión general de la mente humana, sino de considerar el conocimiento humano como diseño, de considerar el conocimiento del ingeniero, en particular, y ampliar la ingeniería a las áreas humanas como la psicología, la lingüística, la antropología o la neurociencia, en suma, construyendo una ingeniería psicológica, lingüística, antropológica o neurológica, en suma "ciencia social tecnológica" según la expresión de Popper. (1987:60), o lo que sería más ajustado denominar, "una nueva ingeniería".

3. EL AREA DE "PROGRAMMING SYSTEMS" DE LA SCS.

3.1. QUÉ TIPO DE INVESTIGACION SE REALIZA.

3.1.1 ¿ QUIEN FORMA PARTE DE "PROGRAMMING SYSTEMS" EN CMU?.

Programming Systems es la segunda área más influyente en la School of Computer Science de CMU. Constituye el grupo más numeroso de dicha escuela después de la de Artificial Intelligence. Es la otra área fundacional del antiguo Department of Computer Science iniciado por Perlis, Newell y Simon en 1965.

Durante los 25 primeros años, los directores del Departamento han provenido de esta área: Perlis, 1965-71, Traub 1971-78, y Habermann, 1978-1991. (Traub realizaba también Theory). Todos ellos eran matemáticos de origen. Una razón que podemos encontrar de esta no-correspondencia entre el área mayoritaria del departamento, la Artificial Intelligence, y , el carácter de su dirección, es que a escala nacional, el Programming sí representa la corriente mayoritaria en la inmensa mayoría de Departamentos de Computer Science. Y dentro de ella juegan un papel clave los matemáticos aplicados.

Como explica Allen Newell en "Intellectual Issues in the History of Artificial Intelligence" (1982), entre 1955 y 1965 se desarrolló un conflicto entre dos visiones del ordenador: "Symbols versus Numbers". La primera defendida por los que Newell denominada "scientist in AI" sostenía que los ordenadores eran máquinas que manipulaban símbolos. La segunda, el "the digital-computer field", sostenía que manipulaban números, pudiendo ,cualquier cosa, incluso las instrucciones, codificarse en números.: " The bulk of the computer field, and all its responsible opinion-leaders, simply adotep the view that computers are number manipulators. There was no attempt to argue against the view that computers are symbol manipulators...The result of this clash of views was to isolate AI within computer science, but one with a special point of view that made it somewhat suspect, indeed somewhat radical".(1982:10).

Sin embargo, en CMU, en particular en lo referente a la investigación la relación entre A.I. y Programming Systems ha sido la inversa. En todas las propuestas ("proposals") realizadas a DARPA desde los años 70, el área prioritaria de la misma ha sido Artificial Intelligence y sólo secundariamente las que correspondían a Programming Systems. Las propias dificultades de la A.I. para pasar su investigación a tangibles resultados ha permitido que Programming durante los años 80s cobrará más fuerza en el Departamento. Se ha reforzado el peso de los matemáticos y lógicos en el área de "Theory", se han conseguido realizaciones importantes en sistemas operativos como MACH, y por último, se ha conseguido iniciar un master en Software Engineering que lo dirige esta área.

Programming Systems es un área de investigación formada por un total de 28 profesores e investigadores, y un conjunto de 43 estudiantes en el curso 91-92.

De estos profesores 7 pertenecen también al área de Theory.

"Programming Systems" y "Theory" son las dos áreas dentro de este Programa de Doctorado con más directas interrelaciones. De hecho, los paradigmas que se desarrollan en el área de "Theory", ("Algorithms", "Complexity Theory", y "Logic") se dirigen a fundamentar teóricamente el trabajo realizado principalmente en "Programming Systems". Por contra, la Artificial Intelligence encuentra su fundamentación en CMU sobre todo en psicología, y, en menor medida, en la lógica.

Aquellos profesores que supervisan el mayor número de estudiantes eran en el curso 1989-90: Thomas Gross (9), Peter Lee (8), Rich Rashid (8), (que en 1991-92 ha pasado a dirigir el principal laboratorio de investigación de Microsoft dejando la escuela), M. Satyanarayanan (6), Jeannette Wing (6), y Doug Tygar (6).

Esta área del Doctoral Program en Computer Science influye en el Information Technology Center , de igual forma que Artificial Intelligence influye en Robotics y Machine Translation. En concreto, varios computer scientists de la misma: Jim Morris, Mahadev Satyanarayanan, John Howard y otros fueron los principales encargados de construir el software de la red Andrew, el producto principal del ITC. (1986). Por su parte, Mary Shaw, veterana profesora de programming está asociada a su vez con el Software Engineering Institute, y ha sido una de las principales encargadas del curriculum de la School.

3.1.2. LA INVESTIGACION "BASIC" Y "STRATEGIC" EN "PROGRAMMING SYSTEMS".

Programming Systems es un área principal de investigación en la escuela, como la A.I. En las sucesivas proposals del "basic contract" que CMU tiene con DARPA para financiar la investigación en "Information Processing", esta área ha desarrollado una investigación en "Software Technology" (1976-78), "Distributed Processing" y "Programming Technology" (1980-86), "Reliable Distributed Systems" y "Programming Environments" (1987-90).

De Programming Systems partieron los trabajos de Alan Perlis en lenguajes de programación como ALGOL 60 ,en esa misma década, así como los trabajos de Nico Habermann en ADA, el lenguaje de programación standard del Department of Defense, y su línea en ambientes informáticos o "environments", de ayuda a la

programación ,en los años 70s que culminaron en el proyecto GANDALF(1980-1990). También fueron protagonizados por esta área los trabajos en "distributed processing" que condujeron a la red Andrew y ,por último, el citado proyecto MACH en los 80s, dentro de la Strategic Computing Initiative de DARPA. En la actualidad, los proyectos de investigación principales en esta área son los que se siguen presentando dentro del programa que DARPA financia y se incluyen dentro de dos grupos: "BASIC" y "STRATEGIC", estos términos no se corresponden con lo que se entiende por "basic" y "applied research" tradicionalmente.

Los proyecto de investigacion "BASIC" del area de Programming Systems son dos:

1. "OBJECT MANAGEMENT" (1989-1993) (CMU-SCS-Basic, 1989). En este programa participan Nico Habermann , Jeannette M. Wing, Edmund M. Clarke Jr., Eugene Rollings, Mahadev Satyanarayanan, y Doug Tygar. Es un programa que se considera "basic research". Está dedicado al diseño de una red informática nacional que permita el almacenamiento, acceso y obtención de un complejo sistema de información por un extensa red de usuarios. "We envision a future nationwide network of persistent objects that will support storage, access, and retrieval by diverse users. Today, files and databases are the only types of persistent objects that we can manage on a large scale. We propose to build a highly scalable, distributed system that provides secure and reliable, yet efficient access to persistent objects spanning a broader range of types". (Object Management. 1989:1-1).

"Object Management" es una investigación destinada a poder manejar este sistema de "objetos" informáticos. Para poder avanzar en el diseño, este proyecto tiene previsto construir un prototipo experimental, una arquitectura denominada Odyssey, que soporte un almacén distribuido y el acceso a diversos "objects".

Relacionado con este proyecto de software, esta el proyecto de Theory, "ERGO: Semantically Based Program-Design Environments".(1989-91) (CMU-Department CS-1988),en el que participan Peter Lee, Frank Pfenning, John Reynolds, Gene Rollins y Dana Scott. Es un proyecto para aumentar la habilidad de los programadores para desarrollar y mantener un software correcto, adaptable y eficiente. Para ello, el enfoque que se sigue se denomina: "Design-Based Approach" (DBA) (op. cit.:1). La novedad que aporta es que recoge no sólo las especificaciones del programa sino también las decisiones que el programador

realiza a lo largo del proceso, "a body of design information that describes how an implementation is related to the original specification" (ibid:1), esto es, recoge el programa del propio programador.

2. "CREATING GRAPHICAL APPLICATIONS" (CMU-SCS- Basic Research, 1989), dirigido por Brad Myers, y con la colaboración de Dario Guise.

Este es un proyecto dirigido a facilitar la tarea de programación creando interfaces gráficas interactivas que permitan una manipulación directa de los objetos representados en ellas por el usuario. El sistema a construir continua el GARNET project, iniciado por Myers con anterioridad.

Los objetivos del proyecto son claros: "We propose to build an environment in which all aspects of graphical, highly interactive user interfaces can be specified without programming. Instead, user interfaces will be specified using graphical direct-manipulation techniques and a very-high-level specification language. " (1989: 1-1). En suma, se trata de construir como el anterior proyecto un "artificial environment", en concreto preocupandose de facilitar la interacción "human-computer". El papel del "computer scientist" es claramente el diseño de estos ambientes artificiales.

Todos estos proyectos se consideran "basic research", y a diferencia de lo que se entiende por "basic research" tradicionalmente, esto es, una investigación dirigida como primer fin al "understanding of the subject under study", según definición aceptada por la National Academy of Science, la National Academy of Engineering y el Institute of Medicine, (National Academy of Sciences, 1989: 2-22), persiguen directamente el diseño de sistemas informáticos. Así el proyecto Object Management afirma: "During the proposed contract period, Carnegie Mellon University will conduct a broad program of basic research in distributed, type-specific object management. The principal tasks of our work will be to:

- Design and build a Common Lisp framework that will support multiple computationally-intensive satisfaction engines...
- Design and build components that fit into the framework in support of a software development laboratory...
- Design, build, and evaluate an experimental prototype of an architecture that supports large-scale object distribution.
- Use the framework and its components in the above distributed prototype. This

use will include demonstrating a distributed version of the software development application."(1989:5-1).

Estas son las tareas de un programa de "basic research" en Programming Systems en CMU.

Un segundo gran proyecto de investigación, éste dentro del programa de DARPA denominado "STRATEGIC", es el denominado: "RESEARCH ON PARALLEL COMPUTING: VERY LARGE SCALE OPERATING SYSTEMS" (1989-93) (CMU-SCS- Research on Parallel Computing, 1989). En este proyecto, dirigido por Rich Rashid, (al menos hasta 1992) participan otros profesores como Eric C. Cooper, Scott E. Fahlman, Alessandro Forin, Robert Sansom, Peter A. Steenkiste y Hideyuki Tokuda. Su objetivo es desarrollar un "systems software environment called Terra OS which will address problems created by new computer technology in the 1990s". (1989:1-1). Específicamente, TerraOS está pensado como sistema operativo de sistemas de grandes redes de multiprocesadores, como la que se está construyendo en CMU, denominada NECTAR, dirigida por H.T. Kung. Este sería un "operating system environment" que perfeccionaría el sistema operativo MACH desarrollado en CMU en 1985

El objetivo del proyecto es permitir a grandes multiprocesadores, supercomputadores y colecciones de ordenadores dispersos geográficamente operar no como una colección de autónomos sistemas con interconectividad, como es el caso en la actualidad, sino como sistema orgánico único, "a single system organism".

Ambos programas, BASIC y STRATEGIC forman parte de la STRATEGIC COMPUTING PROGRAM de DARPA antes estudiado. De las tres aplicaciones que se preveían en dicho programa: "an autonomous vehicle", "a pilot's associate", y "a battle management system (BMS)", la más compleja y decisiva era esta última, dado que consistía en un sistema informático inteligente para ayudar en la gestión de una campaña militar. Esta aplicación requería según el programa citado: "infinite planning and reasoning processors, vast knowledge and database management systems, perhaps no vision systems, but highly complex distributed, survivable communication systems" (Strat. Comp. Init.1983:12). Proyectos como GANDALF, GARNET y MACH han contribuido al diseño de estos sistemas de gestión de bases de datos y conocimiento, y a la construcción de sistema de comunicación distribuidos altamente complejos. Analizemos uno de ellos, el proyecto MACH

3.2. EL PROYECTO MACH: LA "STRATEGIC RESEARCH".

3.2.1. LA DEFINICION TRADICIONAL DE "APPLIED RESEARCH".

La definición oficial aceptada por la National Academy of Science y la National Academy of Engineering de Estados Unidos es ésta: "Applied Research is the systematic study where the primary aim of the investigator is directed toward gaining knowledge or understanding necessary for determining the means by which a recognized and specific need or commercial objective may be met."

(NAS,1989:2-22). Ello la diferencia de la actividad de "desarrollo":

"Development is the systematic use of the knowledge or understanding gained from research, directed toward the production of useful materials, devices, systems, or methods, including design and development of prototypes and processes".(ib:2-22).

Pero como veremos, este proyecto no se ajusta a estas clasificaciones. No tiene como objetivo principal el "understanding" en el sentido científico, y ni siquiera pretende ser un prototipo al modo de un diseño experimental. Se concibe como un producto, "a product", pero no de carácter comercial, sino un producto de carácter estratégico, distribuido gratuitamente al conjunto de empresas, universidades e instituciones públicas del país con el sólo requerimiento de una licencia de CMU.

Mach quizá sea el éxito más notable en la historia de Programming Systems de CMU. Este nuevo sistema operativo fue desarrollado entre 1984-85 por un equipo de 20 profesores, estudiantes, y programadores dirigidos por el profesor Rich Rashid.

Este ha sido uno de los resultados de un amplio programa concebido como "applied research" en "supercomputer technology" iniciado en 1983 por Allen Newell, Raj Reddy y H.T. Kung. (CMU-Proposal for Research on Supercomputers, 1983).

En CMU, el proyecto MACH se considera un éxito ("success") en transferencia tecnológica. De este proyecto afirmaba el hasta hoy decano de la escuela, Nico Habermann: "Another example of Carnegie Mellon's contributions to computer science is the Mach system, which is very successful and useful in the design of architectures and the development of large programming systems. It has been accepted by various industries and been taken as the standard by the Open Software Foundation. It's going to be the kernel of many parallel architectures and standards of architectures. That is indeed a success story." (Habermann,

Carnegie Mellon Magazine, 1990: 56).

El profesor Rich Rashid ,su diseñador principal, fue el organizador del pasado Symposium de la SCS, en septiembre de 1990, celebrando el 25 aniversario del departamento. Su nombre se estaba barajando como próximo decano de la escuela, cuando éste comunicó su decisión de aceptar el cargo como director del principal centro de investigación de Microsoft, la principal empresa de software del país, en Seattle.

Rich Rashid era "professor" de la School Computer Science de CMU. En 1974 se graduó cum laude, ("with honors") en Matemáticas en Stanford University. En 1980, recibió el doctorado en Computer Science en la Universidad de Rochester, de donde pasó a Carnegie Mellon. Nació en Iowa en una familia de inmigrantes cristiano libaneses, llegada a Estados Unidos hacía dos generaciones. Según sus propias palabras: "I'm the first generation going to University". (12-3-91).

Sus áreas de investigación en Computer Science eran éstas: "My interests are in the areas of operating systems, file systems and languages for distributed and parallel processing, computer architectures and distributed Artificial Intelligence. My major project focus in the Mach Project, in which we are trying to build a new kernel foundation for distributed and multiprocessor software development". (CM-CS-Faculty Research Guide, 1991-92:64).

3.2.2. MACH PROJECT : EL DISEÑO DE UN NUEVO SISTEMA OPERATIVO ESTRATEGICO.

Este nuevo sistema operativo diseñado fue, en efecto, diseñado en CMU entre 1984 y 1985 por un equipo de profesores, estudiantes, programadores y personal técnico.

Este sistema operativo culmina 10 años de experiencia en el diseño e implementación de sistemas operativos tanto en CMU, el sistema Accent, como en la University of Rochester , el sistema RIG , de donde provenía Rich Rashid, incorporado a CMU en 1979. El propio Habermann había explorado este campo en el Department of Computer Science con su libro "Introduction to Operations System Design" publicado en 1976.

MACH tiene un conjunto de rasgos que lo definen como un producto propio de la "computer culture" de CMU:

1. Es un proyecto de investigación encaminado a producir un programa que funciona, un "system that performs very well". Como dirá Rashid: "That comes

back to the long-standing philosophy at Carnegie Mellon that, in the system area, we not only come up with good ideas and design and build prototype systems, but we build things that work and that other people can use..."

(Rashid, Carnegie Mellon Magazine. Fall 1990).

2. Es un programa construido con apoyo de DARPA, dentro de la Computing Strategic Initiative, iniciada en 1982, para promover una nueva generaci3n de ordenadores: la Machine Intelligence Technology, y los ordenadores en paralelo.

3. Este proyecto aport3 un rasgo nuevo en la cultura de Computer Science de CMU: desde el principio se pens3 para tener impacto en las principales industrias del ordenador, en la 3rea del desarrollo de sistemas operativos para arquitecturas en paralelo. Se concibi3 como un modelo de "technology transfer". (Rashid ,1990: Mach, a case study: 421).

Hasta ese momento la filosof3a de los proyectos de CMU hab3a sido la de "concept demonstration" m3s que la de "system engineering". Los sistemas inform3ticos serv3an para demostrar nuevos principios te3ricos pero no para ser empleados en el "real world". MACH se realiz3 con otra perspectiva, la perspectiva ingeniera: se pens3 para ser transferido a la industria, para resolver "real world problems". No se construy3 como prototipo sino como sistema para uso pr3ctico en las empresas, y no en alguna de ellas, sino en el conjunto de la industria del ordenador.

MACH es un "operating system" (OS), esto es, "the master program that controls all the computer's resources (memory, devices, processing time) and allocates them among all the computer's users and programs". (Phillip Miller.1987:12).

Es el programa que regula las relaciones entre el usuario y la m3quina. Contiene las instrucciones que definen como el ordenador ha de interactuar con el usuario, y viceversa, define las reglas b3sicas de utilizaci3n del sistema inform3tico: los c3digos de acceso al ordenador (password, ...), la forma de manipular los archivos (files), la manera de acceso al resto de programas del sistema, el funcionamiento de la red, etc.

Inicialmente cada fabricante desarroll3 su propio sistema operativo,(VMS, MVS, MS-DOS) siendo dif3cil que un programa que funcionara en un sistema operativo lo hiciera en otro de una empresa distinta. En los a3os 80s. el sistema UNIX fue un primer intento de llegar a un sistema abierto , "open system standards", que permitiera que un mismo programa funcionara en distintas m3quinas de distintos fabricantes. Varios grupos nacionales e internacionales intentaron definir

nuevos entornos informáticos "abiertos", tratando de preservar la continuidad con los sistemas operativos desarrollados durante los 60s y los 70s.

En 1983, La Defense Advanced Research Projects Agency (DARPA) empezó a financiar una gran iniciativa, el Strategic Computing Initiative, para el desarrollo de una nueva generación de ordenadores, los llamados "high performance parallel computers". Como explica Rich Rashid: "This goal could not be reached, however, without a common software base which could be used on a wide range of new architectures and provide a single system software environment for application designers" (1990: 413). MACH se concibió como un instrumento técnico al servicio de un cliente, DARPA y la industria, que precisaba resolver la incompatibilidad de sus sistemas informáticos.

3.2.3. EL "ALTERNATIVE APPROACH" DEL PROYECTO MACH.

¿Cómo ha conseguido MACH ofrecer un sistema operativo que permite la colaboración de distintos tipos de arquitecturas informáticas?. La razón es que el equipo de diseñadores del mismo, ha partido de un enfoque distinto, nuevo. Entre los principales miembros de este equipo están a parte de Rich Rashid, Avadis Tevanian Jr., David B. Golub, David L. Black, Eric Cooper, y Michael W. Young.

En lugar de construir el sistema operativo en un solo bloque, como los sistemas tradicionales, UNIX o VMS, y ofreciéndose alternativo a los mismos, ha construído un sistema operativo "kernel" , una especie de enchufe universal en el que más de un entorno de sistemas operativos puede conectarse.

Sus dos rasgos únicos son: 1. "Hardware independence", ésto es, la construcción de un sistema capaz de funcionar en múltiples tipos de máquinas. y 2.

Compatibilidad con los restantes sistemas operativos. Es una "enabling technology" (Rachid, 1990 Carnegie Mellon Magazine: 42), un sistema operativo general.

Rashid explica así su nuevo enfoque: "Because of its modular design, Mach offered an alternative approach which allowed systems builders to separate those parts of the operating system which control the basic hardware resources--often called the operating system 'kernel'--from those parts of the operating system which determine the unique characteristics of an operating system environment , for example a particular file system interface. The advantage of this scheme is that it can allow more than one operating system environment to be implemented

on the same hardware/software base so that machine dependent software need be written only one for each new architecture. It also allows, at least potentially, for more than one operating system environment to be supported in 'native mode' simultaneously on the same hardware". (1990, Mach, A case study, : 416).

Estas ventajas se vieron reforzadas por la decisión del equipo dirigido por Rashid de distribuir libremente todo el software desarrollado por el MACH Project a cualquier universidad, compañía u otra organización que lo quisiera usar, según Rashid "the single most important decision" del proyecto.

UNIX hasta el momento era el sistema operativo con más posibilidades de convertirse en el sistema estándar para la nueva generación de arquitecturas en paralelo. UNIX es un producto que pertenece a la ATT, la principal empresa telefónica de Estados Unidos. MACH se concibe no como una alternativa comercial a este sistema, sino como un sistema operativo que podríamos denominar "público", que permite que los sistemas comerciales sean compatibles con él, y que, a través de él, lo sean entre sí. . Es una expresión de las posibilidades de la colaboración Universidad-Administración-Empresas en una área en la que hasta ahora los intereses se planteaban de forma disyuntiva.

3.2.4. "TASKS, THREADS, PORT, MESSAGE": NUEVAS ABSTRACCIONES QUE FUNCIONAN.

MACH es un sistema operativo que aporta nuevas características, nuevas funciones, que lo hacen distinto a su principal competidor, el sistema UNIX. Estas funciones son definidas por Rashid como "basic abstractions" o abstracciones básicas. Y son cuatro: "a task", "a thread", "a port" y "a message".

Las dos primeras funciones nacieron de la ruptura de una única abstracción del sistema operativo UNIX, " a process", en dos funciones denominadas "task" y "thread". "A task", es una abstracción ,que corresponde a esa función de "process", es una unidad básica de recursos del sistema informático (procesadores, memoria virtual...). Por su parte "a thread" es la especificación de una operación en esa "task". El objetivo de esta división de abstracciones es permitir que múltiples operaciones se puedan realizar en paralelo en una única tarea o "task". Ello significa que la función denominada "thread" funciona como un programa independiente contraoperando en una unidad básica de recursos.

Esta nueva función la denomina R. Rashid tanto "abstraction" o "basic function" (1986:39), como "mechanism" (1990: 415) y se convierte en un instrumento clave para el funcionamiento de multiprocesadores y arquitecturas en paralelo, manteniendo al mismo tiempo la compatibilidad con el sistema UNIX. Pero el invento no consiste en el mecanismo en sí que ya se conocía con anterioridad, sino en su aplicación dentro del sistema UNIX como tarea diferenciada. Un sistema de "multi-threading" había sido inventado por varias generaciones de diseñadores de sistemas operativos como Pathway de Tandem. Los denominados "developpers" de MACH reconocieron el valor de esta construcción alrededor de 1984 cuando estaban trabajando en otro sistema anterior a MACH, el llamado ACCENT.

La tercera abstracción introducida fue "a port". Esta es la abstracción de comunicaciones básica de MACH. "A port" es un canal de comunicación que permite el intercambio de mensajes.

Como explica Rashid: "The Mach abstractions were chosen not only because of their simplicity, but also for performance reasons." (Unix. Review, 1986:39).

Mary Shaw y William Sherlis, profesores de la SCS, (éste en la actualidad con destino en DARPA) en su "Mathematics Curriculum and the Needs of Computer Science"(1982), han venido sosteniendo la necesidad de basar la "computer science education" en un mayor conocimiento de las matemáticas, en particular, de su poder de abstracción. Tradicionalmente, por abstracción se ha entendido en matemáticas, la habilidad de captar la estructura esencial de una situación compleja: "to glean the essential structure from a complex situation" (Lynn A.Steen. 1983:99).

Esta capacidad se ha considerado uno de los rasgos básicos de la "mathematical maturity". La geometría euclideana se consideraba uno de los ejemplos de esta capacidad de abstracción: de los números y objetos se pasaba al formalismo de variables y definiciones.

Igualmente, la habilidad de sintetizar, de crear nuevas ideas a partir del uso efectivo de las antiguas, habilidad expresada en la deducción lógica, o el razonamiento indirecto, se ha considerado la otra marca tradicional de la "mathematical maturity".

El problema es que en "computer science" esta capacidad de abstracción no está separada de la capacidad de generación de nuevas estructuras mediante la síntesis, sino que forman un mismo proceso. El "computer scientist" ha de no

sólo captar la esencia de una estructura existente sino de crear una nueva estructura abstracta. Esto también lo hace el matemático. Pero en el caso del "computer scientist" esa estructura además ha de funcionar, pues se trata de una estructura abstracta que ha de dirigir las operaciones de un ordenador. Es una nueva función abstracta y a la vez operativa. Y esto es algo más de lo que hacen los matemáticos.

"A task", "a thread", "a port", "a messages", son las nuevas funciones que el equipo de Mach introdujo en su sistema operativo, las nuevas abstracciones que hacen funcionar al sistema.

Computer science parte de esta unión de conceptos de la matemática y la ingeniería en un solo cuerpo teórico. La matemática se ha hecho ingeniera. Como diría Perlis: " What will be new is the attachment of algorithms and languages to a machine--the electronic digital computer--which is the agent commanded to perform (to do,to accomplish) algorithms expressed in languages" (1972:2).

La "abstracción" no es la esencia de la "computer science", su clave es la abstracción que funciona, dado que ha de formar parte de un sistema mecánico. Son nuevas funciones, en tanto que son nuevas capacidades de actuación que se le introducen al ordenador, lo que redundara en nuevas posibilidades para el usuario. "It's an enabling technology" según Rashid. (Carnegie Mellon Magazine.Fall 1990). Una tecnología que hace posible otras.

3.2.5. EL DISEÑO Y LA GESTION DEL MACH PROJECT .

Mary Thompson es la Project Manager de MACH. Tiene una experiencia de 13 años en CMU, y anteriormente otros 13 en el MIT. Según ella: "Developping a operating system like MACH it takes a lot of time. Only a few group of univerities can do that. In european universities like Cambridge in United Kingdom, they don't have the resources to do that. They must focus only on the theoretical aspects of computing". (Entrevista personal, 11 junio 1991).

En su momento de máximo desarrollo, el proyecto Mach requirió un equipo de 20 personas, la mitad de ellas " professional staff", es decir, programadores, encargados de conseguir que el sistema operativo pasara de las ideas a convertirse en un "product". En palabras de R. Rashid: "We generate products, not only ideas". (Entrevista personal: 12-3-91).

Este equipo, tan grande como un departamento de un "college" de CMU, contaba aún en junio de 1991 con 16 personas bajo la dirección de Rich Rashid, la

colaboración de otros dos profesores del departamento: Alessandro Foring y Brian Bershad. Tenían a su cargo a 7 personal de staff y 6 estudiantes de doctorado.

Un conflicto planteado en el equipo de Mach, según Mary Thompson, fue los diferentes enfoques de los graduate students y el staff. Mientras que los primeros pretendían introducir innovación en el sistema, los segundos tendían a sacar el trabajo adelante, cumpliendo con las especificaciones y los plazos.

Esta labor, según la Project Manager de MACH, se veía entorpecida a su vez porque los estudiantes, sometidos a las clases, no producían lo que había que producir a tiempo.

En cualquier caso, y a diferencia de otros sistemas operativos desarrollados por el departamento que nunca llegaron a la industria, MACH ha conseguido, según Mary Thompson, su objetivo: "to be useful for the people". Y en primer lugar para sus patrocinadores, DARPA y la Digital Equipment Corporation.

La construcción de MACH dió lugar a tres tesis doctorales en Computer Science escritas por A. Tevanian, M. Young y D. Black, que contenían las propias contribuciones de estos estudiantes al proyecto. La de Avadis Tevanian, Jr. titulada "Architecture-Independent Virtual Memory Management for Parallel and Distributed Environments: The Mach Approach"(1987), comenzaba así: "This thesis describes the design and implementation of a new, portable, memory management system and evaluates it by direct comparison with commercially developed memory management systems....The implementation and evaluation of Mach, described in this thesis, demonstrates that it is possible to build a virtual memory system that exhibits: -architecture independence..., -multiprocessor and distributed system support, ...advanced functionality".(1987:1).

Una tesis para el grado de Philosophy Doctor (Ph. D.) basada en demostrar que el sistema de memoria virtual de MACH era posible, y cuya prueba no era otra que el citado estudiante lo había construido.

3.2.6. EL PROYECTO MACH COMO "TECHNOLOGY TRANSFER BY DESIGN".

En el 25 Aniversario del Computer Science en esta universidad, Rich Rashid se encargó de publicar el libro final de la conmemoración. En él, el propio Rashid presentó una ponencia titulada: "MACH, A Case Study in Technology Transfer"(1990).

En ella, explicaba la historia del "MACH Project" destacando su carácter de tecnología transferida con éxito a la industria. Su modelo de transferencia

tecnológica lo denomina "technology transfer by design" , y difiere del tradicionalmente empleado en el medio académico.

Desde hace años existe una creciente preocupación por la transferencia tecnológica en Estados Unidos, y en Europa. Esta preocupación proviene del hecho de que tal transferencia no funciona o funciona deficientemente. El modelo del que se parte es el denominado "unique innovation engine", (Dertouzos.1988 CSTBoard). En él , las "universities" y su "basic research" ocupan el primer lugar en la generación de innovaciones. Esta , luego, se transfieren a las "venture companies" y a las "mature companies".

Pero veamos cuál es el modelo del MACH PROJECT. Rich Rashid en dicho papel lo resume en un conjunto de rasgos:

1.) Mach Project se apoyó, en efecto, en una sólida base previa de investigación

"We began our work from a solid base of previous research experience".

At the time the Mach project began, I and other members of the project already had as much as 10 years of experience in the design and implementation of network operating systems. This experience led to simpler interfaces and more rapid and effective implementation" (1990:412).

En efecto, existe en su origen una investigación básica pero basada en el diseño, esto es, " the design and implementation" de sistemas operativos. Ya hemos visto con anterioridad que la "computer science" realiza un tipo de "basic research" distinta a la de la ciencia tradicional. Existe una investigación básica científica y otra tecnológica, también básica, que tiene por objetivo el diseño.

2.) El sistema Mach se diseñó con un fin explícito de transferencia tecnológica.

"We had as an explicit goal of the project the transfer of technology to other research groups and industry.

This was due in part because the project began as a collaboration with government and industry and in part because our earlier experiences with operating systems had led us to believe that we could only successfully validate our research ideas through extensive use by application programmers" (1990:412).

MACH no ha salido del "basic research" ni siquiera tecnologico, sino de un proyecto de la denominada "strategic research", concebido con una clara orientación desde el principio de ser transferido al "real world", y por tanto,

con unas características distintas a otros proyectos estratégicos.

La visión que la universidad tiene una tecnología ya elaborada, nacida de su misión de crear conocimiento y presta a transferirla a la industria, no se corresponde con la experiencia de MACH.

El sistema operativo MACH no existía como tecnología preexistente al acuerdo entre el equipo de R. Rashid, la DEC y DARPA. Fue un resultado de este proyecto conjunto, y se pudo transferir porque se concibió desde el principio con esta finalidad. Como explica Rashid: "From the earliest days of the Mach project, it was envisioned to have an impact on the industrial and research development of operating systems for parallel architectures..." (1990:413).

Antes se tuvo el objetivo de transferir la tecnología y luego se construyó ésta ad hoc. Como afirma Rashid: "Technology transfer can happen by accident, but is more reliable when done by design". (1990:420).

3.) El sistema Mach se planeó, en directo, como "product", no como "prototype", ni como "demo" (demostración).

Pero hay una razón más profunda planteada por Rashid. Este equipo creía que sólo se podían validar sus ideas a través de su uso por la comunidad de programadores: "we could only successfully validate our research ideas through extensive use by application programmers" (Rashid, 1990:).

Ello introduce una visión distinta de la tecnología a la imperante hasta hoy en la SCS. Hasta ahora, los sistemas informáticos construidos en Computer Science se regían por el principio de "concept demonstration". Servían para ejemplificar nuevos conceptos. La "demo" es un término usado en esta cultura y que significa esta actividad de demostrar un sistema construido, ejemplifica el hecho que funciona, "it works".

En explícito, se prefería este enfoque al "system engineering", que es el sistema de demostración de la tecnología de los ingenieros, ligado más a la construcción de un producto comercial, y cuya pieza básica es el prototipo.

Pero el enfoque de MACH está pensado para que funcione en el llamado "real world": "We generate products, not only ideas" (Interview. 12-3-91). No sólo no pretendía ser una "demo", sino tampoco un "prototype" sino un "product".

La tecnología así vista se valida si y sólo si funciona y transforma el "real world". La norma de Rashid era clara: "Don't build prototypes. Code that both embodies new ideas and that works and works well is the best form of publication

and most convenient vehicle for technology transfer" (1991:420.). En suma, ideas, diseños que funcionen, y funcionen bien.

Pero el término "product" hasta ahora se ha asociado a producto comercial. Pero MACH no es un product en el sentido comercial, pues su objetivo no es que se compre sino que se utilice, y para ello la forma adoptada ha sido distribuirlo gratuitamente. Si hubiera aparecido como otro producto comercial, las empresas lo habrían considerado un competidor y habrían ofrecido sus productos, con lo cual el último perjudicado habría sido la propia empresa al no disponer de un sistema operativo compatible. De esta forma, concibiéndose como producto estratégico, no comercial, se ha podido adoptar por un sector más amplio que el mero mercado: el sector de productores y usuarios de informática en un sentido amplio.

Ello indica que determinados productos informáticos ven frenado su desarrollo cuando se consideran sólo como productos comerciales . Esta informatica practica en CMU no sólo es un "product" es un producto estratégico, y así es cuando se desarrolla al máximo como el proyecto Mach indica.

4.) El equipo de Mach consiguió que este sistema proporcionara más prestaciones que los otros sistemas, con igual o mejor "performance", manteniendo la compatibilidad con ellos.

"We met important needs at a time when more traditional systems software platforms did not". (Rashid, 1990: 412)

Este sistema daba más prestaciones que los existentes en el mercado, pues permitía utilizar un solo sistema de software en diversas plataformas de hardware.

"Mach Project" tuvo libertad para poder construir un sistema operativo independiente de los diversos sistemas de hardware de las compañías pero compatible con ellos.

" We recognize the need to provide binary compatible with earliers systems"(Rashid, 1990: 412).

Este proyecto empezó como un proyecto conjunto entre DEC y CMU con ayuda de DARPA, pero la finalidad del equipo no se limitó a servir a los planes de una sola empresa: "In the fall of 1983, DEC approached CMU with a proposal to do joint research on a 32-processor VAX multiprocessor system. Digital's efforts were to be concentrated in the area of multiprocessor hardware design ...CMU's

contributions to this project were expected to be in the area of operating systems, measurement and performance evaluation...At the same time, DARPA was beginning to fund a major initiative in high-performance parallel processing--the Strategic Computing Program". (1990:413).

En la entrevista realiza a R. Rashid, este hizo hincapié en que su preocupación era "how the system would be used". (12-3-91). Pero este usuario no se limitaba a una sola empresa. Al estar financiado dentro de un programa de DARPA, el proyecto podía presentarse al conjunto de la industria norteamericana: "We have an advantage: not to work with a single company" (Rashid. Interview 21-5-91). Una de las consecuencias de ello fue el que MACH no fue patentado sino que su software el equipo lo decidió distribuir "free available for all people. This is our decision".(Interview 12-3-91)

5.) El equipo de MACH introdujo una denominada "agressive distribution policy". (1990:420).

El proyecto se empezó en Octubre de 1984. En un año y medio la primera versión de Mach funcionaba en un procesador VAX 11/784. Desde el principio el equipo distribuyó el sistema entre los propios computer scientists de CMU a fin de convertir a MACH en el "primary computing environment for the school".

En agosto de 1986, aparecía en la UNIX Review un artículo de Richard Rashid, "Threads of a new system", donde presentaba públicamente el proyecto.

En enero de 1987, empezaron la distribución de Mach a grupos fuera de CMU. En Mayo de 1988, cerca de 100 sitios fuera de CMU usaban MACH. En septiembre de ese mismo año, la compañía NeN anunciaba que basaría su nueva estación de trabajo en una versión de Mach. En noviembre de 1989, la Open Software Foundation (OSF), un consorcio de grandes fabricantes de computers anunciaba planes de basar su primer sistema operativo en una versión de Mach, y al año siguiente varios miembros de este consorcio anunciaban que Mach sería el sistema operativo de varios productos que se preveían en el mercado para 1991.

Rashid acababa su papel en el 25 Anniversary del departamento con esta frase: "Be aggressive in marketing new ideas and systems". (1990:420).

Conclusiones del caso MACH como transferencia tecnológica:

1. La innovación es una idea-producto, no una idea sólo. Se apoya en un tipo de investigación básica basada en el diseño, pero sale fundamentalmente de lo que

se denomina "applied research".

2. La transferencia de Mach se planeó antes de diseñar el sistema operativo. La "technology transfer by design" se adelanta en este caso a la propia "research". Se inventa lo que se quiere transferir. No al revés.

3. La empresa productora es usuaria de este tipo de tecnología que tiene un carácter estratégico, pues va orientada a solucionar un problema no de la empresa sino del conjunto de usuarios de informática, un sistema operativo compatible con todo tipo de hardware.

4. El equipo de Rashid produjo no una idea técnica viable sino un producto que funcionaba. Pero la prueba que funcionaba es que era usado por la comunidad de programadores. La tecnología es una actividad cultural y su validez la proporciona su uso por la comunidad en cuestión a la que se dirige. Si no se utiliza, no es válida.

El problema en la denominada "technology transfer" es que puede que las universidades tengan menos tecnología de la que se cree para transferir. Con lo que el "technology transfer" ha de empezar planteándose como "technology design", la generación de una tecnología que esté pensada para ser utilizada en el "real world". Y aún más. Se precisa que las universidades se planteen como centros de diseño de sistemas, productos, organismos innovadores para la sociedad.

En las mismas jornadas del 25 aniversario, otro antiguo profesor de CMU, Sam Fuller, hoy responsable de los programas de investigación de la empresa DEC, hablaba del mismo tema que Rashid: "Technology Transfer". Si bien señaló la necesidad de una reevaluación del tema, el esquema que empleó para su estudio seguía la tradicional secuencia: UNIVERSITIES--INDUSTRY-PRODUCTS.

Pero Mach Project ha introducido una nueva secuencia:

"UNIVERSITIES"-- "STRATEGIC PRODUCTS"-- "SOCIETY", esto es, (Universidades, administración pública, compañías).

Determinados equipos de CMU empiezan a construir nuevos productos que distribuyen agresivamente no sólo a las empresas, al conjunto de ellas, no solo al Departamento de Defensa o la administración pública, sino también a su propia universidad y otros centros de investigación . Estos productos tienen un distinto carácter, se rigen por criterios no comerciales sino de utilidad general o estratégica como MACH

3.2.8." TERRA OS" PROJECT : EL FUTURO DE MACH .

Este proyecto de diseño de este sistema operativo , que permite a ordenadores de distinto tipo trabajar en red, se ve continuado ahora con el proyecto de CMU:

"VERY LARGE SCALE OPERATING SYSTEMS" (1990-1993), financiado también por DARPA, dentro del mismo programa de Strategic Computing.

El equipo inicial formado en 1989 estaba formado por Rich Rashid, Eric Cooper, Scott E. Fahlman, Alessandro Forin, Robert Sansom, Peter Steenkiste y Hideyuki Tokuda.

Este proyecto pretende construir un nuevo sistema operativo, denominado TERRA OS, construido sobre el propio MACH, para hacer frente a los problemas de la tecnología informática en la década de los 90s: "Ultimately, our vision of TerraOs is one of an operating system environment layered upon an advanced version of the Mach kernel. This environment would allow large multiprocessors, supercomputers and geographically dispersed collections of computer systems to operate not as a collection of autonomous systems with interconnectivity (as is the case today) but as a single system organism". (CMU-CS, Research on Parallel Computing, 1989.1-1).

Este proyecto cobra sentido si lo situamos en un contexto más amplio. Estados Unidos está poniendo en marcha un proyecto denominando "NATIONAL RESEARCH AND EDUCATIONAL NETWORK", (NREN), inicialmente aprobado por el Congreso y el Senado, para construir una red nacional que una a toda la comunidad de investigadores y educadores en Estados Unidos mediante una red de fibra óptica y basada en ultrarapidos supercomputadores.

CMU está construyendo la red Gigabit NECTAR que pretende ser un experimento piloto de este proyecto.

Ello plantea enormes problemas de compatibilizar sistemas informáticos muy diversos, desde ordenadores personales a supercomputadores y multiprocesadores.

Desde arquitecturas Von Neumann a sistemas de computación en paralelo.

Hace falta un sistema operativo que proporcione una base para el funcionamiento de dicha red. Son diversos los esfuerzos internacionales que se están dando para construir ese tipo de sistema operativo.

En Estados Unidos, hay en la actualidad cinco proyectos que tratan de resolver dichos problemas: los sistemas SPRITE y DASH en la University of California at Berkeley, el V KERNEL en Stanford, el LOCUS en la University of California at Los Angeles, y el QUICKSILVER OS que desarrolla el Almaden Research Center de

IBM. En Europa hay dos proyectos similares, el AMEOBA, de la Universidad de Amsterdam, y el CHORUS iniciado por el INRIA de Francia, primero y continuado por Chorus System, Ltd.

Este nuevo sistema de CMU continúa la misma orientación hacia la industria iniciada por MACH: "The commercial success of the Mach kernel has placed us in a unique position of direct contact with the hardware development organizations of many companies and given us access to information about their development plans (both research and commercial). We are currently intimately involved with both major UNIX systems groups...and with a large numbers of individual companies...We hope to use our position of access and trust to greatly speed the transfer of Terra Os technology to industry". (1989:6-1).

Así mismo, MACH está siendo utilizado por otros proyectos en la propia School of Computer Science como el grupo de Speech Understanding y el que trabaja en " New Concepts for Advanced High Definition System" (1991-93), basado en la fusión de la tecnología informática , el video, y las telecomunicaciones, "tres mundos tecnológicos distintos que tienden ahora a converger" según Angel Jordan, director de este último proyecto.

En resumen, Mach ha conseguido el propósito que pretendían sus creadores, ser una "enabling technology", a cambio ha producido cambios importantes en el tipo de investigación realizada en la School, y continúa otros aspectos de la vieja filosofía de CMU. En palabras de Rich Rashid: "That comes back to the long-standing philosophy at Carnegie Mellon that, in the systems area, we not only come up with good ideas and design and build prototype systems, but we build things that work and that other people can use...We really do use our own stuff, and we view that as being an important attribute of our work". (Rashid, CMmagazine, 1990:42).

Esto significa "research" para este computer scientist de CMU.

3.3. LA DENOMINADA "BASIC RESEARCH IN PROGRAMMING SYSTEMS": "GANDALF", "ERGO", "TYPES IN PROGRAMMING".

3.3.1. LA DEFINICION TRADICIONAL DE "BASIC RESEARCH".

Según la definición comúnmente aceptada por la National Academy of Science y la National Academy of Engineering de Estados Unidos : "Basic Research is a systematic study where the primary aim of the investigator is directed toward

fuller knowledge or understanding of the subject under study, rather than a practical or commercial application thereof" (NAS, 1989:2-22).

En esto no se ha cambiado desde la definición de qué era investigación pura hecha por Vannevar Bush en 1945: "Pure research is research without specific practical ends. It results in general knowledge and understanding of nature and its laws. This general knowledge provides the means of answering a large number of important practical problems, though it may not give a specific solution to any one of them" (1945:81).

Se mantiene la idea que la teoría no es práctica, y se basa en la comprensión de las leyes de la naturaleza. Y que ésta es la tarea más propia de la Academia. A su vez, la práctica no es una investigación propia de la academia sino de la industria, y su lugar es el laboratorio industrial.

Esta visión de la investigación ha tenido consecuencias para Estados Unidos que sólo ahora, con el auge de Japón, se empiezan a apreciar. Igual que a principios de siglo le ocurrió a Estados Unidos respecto a Europa, el motor actual de la innovación en Japón, no parece su potencia científica sino tecnológica.

Pero en "Computer Science", Estados Unidos aún mantiene una clara superioridad. Veamos que significa "basic research" en esta área de Programming Systems de CMU. En ella, la "naturaleza" a estudiar la hace el propio "computer scientist" y éste no puede dejar de resolver los problemas que esta construcción.

Un programa de "basic research" en CMU es GANDALF, dirigido por Nico Habermann, y que es continuado ahora con el proyecto "OBJECT MANAGEMENT" (CMU-CS- Basic Research, 1989), dirigido por J. Wing y Nico Habermann. Este último proyecto está incluido dentro de la propuesta enviada a DARPA como "BASIC RESEARCH IN COMPUTER SCIENCE", en respuesta a su Announcement 89-05 de 28 de Diciembre de 1989.

3.3.2. "BASIC RESEARCH IN PROGRAMMING TECHNOLOGY": GANDALF PROJECT.

Durante los últimos 10 años, entre 1980 y 1990, un equipo de computer scientists de CMU dirigido por Nico Habermann, decano de la School of Computer Science hasta 1991, ha realizado un proyecto de investigación denominado GANDALF. El proyecto ha investigado "software environments that are interactive, integrated, task-specific, and have direct manipulation user interfaces" (1990:69).

Nico Habermann es "professor" de la School of Computer Science y hasta 1991, decano de esta escuela. Bajo su dirección, dos cambios importantes sucedieron en

la Computer Science en CMU: 1. El departamento se convirtió en colegio universitario, separándose del de Ciencias. y 2. Se constituyó el Software Engineering Institute (SEI), fuera del Computer Science de CMU.

Nico Habermann es doctor en Applied Mathematics por la Technological University de Eindhoven, Holanda, país donde nació en 1932. Llegó a Carnegie Mellon en 1969, colaborando con Alan Perlis inicialmente en la área de lenguajes de programación, y más tarde en ingeniería del software: Según afirma:

"My research interest is in software engineering and in the design and implementation of programming languages, operating systems and programming environments" (CMU-CS . Faculty Research Guide, 1991-92:34).

No deja de ser curioso que el principal interés del decano de la School of Computer Science sea una ingeniería: la ingeniería del software. Habermann fue a su vez, el primer director del SEI.

Los dos proyectos principales en los que en 1991-92 estaba envuelto eran: GANDALF y CS-EDU. Ese último es un proyecto educativo: se trata de basar la educación introductoria en "programming" en un sistema de construcción y análisis de sistemas de software.

El Gandalf System es un "ambiente" especializado en la tarea de construir otros "ambientes" de programación. Sirve para tres funciones: 1. Apoya al diseñador de estos "ambientes" proveyéndole de instrumentos denominados "editors", el mecanismo que permite al usuario crear estructuras como archivos, bases de datos,... 2. Coordina el proceso de generación comprobando la consistencia de las descripciones de dichos mecanismos. 3. Controla la evolución de dicho "ambiente". Los resultados son que este sistema reduce drásticamente el tiempo requerido para producir un "ambiente" de programación integrado.

Esta investigación arrancó a mediados de los 70s, bajo la denominación de "Software Technology" en CMU. Según la propuesta enviada a la entonces ARPA para el periodo 1977-78: "The long range goal of software technology, and the CMU program in particular, is to increase our ability to predictably produce high quality hardware/software systems." (CMU-CS. 1977-78:13).

Las implicaciones prácticas de la tecnología del software, según el texto citado, eran inmensas. Casi el 70% del presupuesto del Departamento de Defensa en informática por aquellas fechas se dedicaba al software y sólo el 30% al hardware. Y se preveía que para 1985, el software llegara a representar el 90%.(ibid.:11).

Esta tecnología se componía de dos partes: **"the principles and knowledge (the know-how) to be used in producing software, and the tools (primary software systems themselves) used in the production of software --compilers, debuggers, editors, design systems,etc".** (ibid.:11).

El problema planteado en la tecnología del software se resumía en una palabra: **"complexity"**. El ordenador funcionaba con una extremadamente complicada combinación de operaciones concretas muy simples. La tarea del programador era simplificar esa complejidad mediante la creación de operaciones abstractas dirigidas a ejecutar determinadas tareas. Ya hemos visto en el MACH PROJECT algunas de estas "task-specific abstractions".

En la Proposal for Information Processing Research, sometida a DARPA, para el periodo 1980-83 se proponía continuar la "basic research in programming technology", con varias aplicaciones. Entre ellas ya se proponía un "advanced programming environment" para el lenguaje ADA, el que preconiza el Department of Defense para unificar sus diferentes sistemas informáticos.

Segun la propuesta: **"Programming Technology (PT) is concerned with all aspects of the construction of hardware/software systems. Specifically, it aims at obtaining 'high quality systems':systems that are produced on time, that are produced to budget, that are correct, that are efficient, that can be maintained and enhanced, and that are tolerant to their human users".** (CMU-CS. 1980-83:5-1).

Se trataba pues de una investigación básica pero de carácter distinto a la clásica en física o química. Era una investigación sobre una tecnología, y el objetivo de la misma era la construcción de dicha tecnología. Esta tarea se planteaba con un carácter normativo y claramente prefigurativo, en el sentido de Margaret Mead.

" Programming Environments of the future must have two properties not found in current systems. -The environment must provide an integrated set of system design and maintenance tools. - The environment must support the collaboration of many programmers and the coexistence of many software modules over a long period of time" (1980:5-5).

Si comparamos la noción de "medio ambiente" de las ciencias naturales y la de la "computer science", en la que se inspira, vemos diferencias claves. La teoría de la selección natural explica que los seres vivos sobreviven por su mejor adaptación a las variaciones del medio en el que se desarrollan. Pero en el caso

de los "programming environments" es distinto. Estos sistemas son del todo artificiales, los construye el "computer scientist". Este tiene la iniciativa en su diseño, construye el artefacto mecánico y con él construye una nueva naturaleza artificial.

Es claro que estos "environments" son también naturales, no son algo "sobre-natural" o "anti-natural" y también evolucionan, como el resto de ecosistemas, pero apreciamos que el ser humano tiene un papel clave en la iniciación de los mismos y en su evolución. De igual forma que la tecnología se adelanta a la ciencia en "computer science", el diseñador se adelanta a su diseño. Las relaciones natura-cultura se invierten, y pasan a ser cultura-natura.

De momento, estos "programming environments" tienen aun una influencia reducida en la vida del ciudadano corriente. Pero al igual que las redes de carreteras, teléfonos o los propios sistemas urbanos, estos "environments" pueden ser la infraestructura básica de las diferentes culturas en un próximo futuro y es conveniente que en su diseño participen, desde el principio, alguien más que "computer scientists".

3.3.3. GANDALF PROJECT : DISEÑANDO HERRAMIENTAS PARA LOS SOFTWARE ENGINEERS.

Pero lo más interesante del GANDALF PROJECT es que se concibe como un proyecto para diseñar herramientas de programación para los software engineers. ¿Qué significa esto?. Significa que la "computer science" se concibe a sí misma en estos proyectos como un instrumento al servicio de la ingeniería del software, reconociendo así que la relación denominada Science&Technology está invertida en este campo.

Hasta ahora la tecnología era, y es, considerada por la ciencia como el saber encargado de producir herramientas, "tools" para avanzar el conocimiento de la naturaleza. Aún hoy macroproyectos como el High Performance Computing and Communications (1992) de la National Science Foundation y otras agencias federales, pretende construir una nueva generación de ordenadores ultrarrápidos al servicio del avance de la ciencia, la biología y su proyecto Genoma Humano, la física y su estudio de la dinámica atmosférica y la biosfera de la Tierra, y otros.

Incluso en CMU, psicólogos cognitivos como Herbert Simon usan el ordenador como

herramienta para el descubrimiento de como funciona la mente humana.

Pero en estos proyectos de investigación básica de Computer Science como GANDALF, la relación se ha invertido. La ciencia sirve a la tecnología construyéndole las herramientas, los ambientes y lenguajes de programación necesarios para realizar la tarea sustancial: diseñar un software adecuado. En este campo, la que tiene la responsabilidad de avanzar el contenido de la disciplina es la tecnología, y quien construye herramientas para ayudar en esta tarea es la ciencia.

Esta relación la volveremos a ver en Dana Scott y su "denotational semantics" que no deja de ser una técnica de programación. Y la encontraremos de nuevo en Computer Systems, en investigaciones como la de Randall Bryant que construye dispositivos de validación lógica que ayudan al diseño de circuitos integrados a gran escala. (VLSI) que realizan los ingenieros electrónicos.

La ciencia en la "computer culture" de CMU complementa, ayuda a la tecnología, siendo un instrumento para acelerar el proceso de innovación, no su motor fundamental.

En la propuesta a DARPA de 1984-86 (CMU-SCS- 1984-86), el proyecto GANDALF introdujo una nueva dimensión, la del "knowledge-based environment", los ambientes informáticos podían contener conocimiento experto

Recordemos que la tecnología de programación se dividía en dos partes. En primer lugar, el "know-how" o los conocimientos sobre como programar (las metodologías , los conceptos de programación, los algoritmos a ejecutar). En segundo lugar, las herramientas o "tools" para programar.

Los investigadores de GANDALF empezaron a considerar la posibilidad de introducción de conocimientos expertos en el ambiente de programación para facilitar la labor del usuario: **"We want to explore the kind of expert knowledge about system building that can be included in a programming environment and in what respects system development can be a cooperative effort of environment and user"**. (1984:5-2). A destacar aquí la forma en que aparece este cambio: como una posibilidad nueva, un "can be". Las innovaciones surgen inicialmente como una posibilidad u "oportunidad", como ya vimos con Logic Theorist y otros proyectos. El problema era introducir un mecanismo que permitiera al diseñador de los llamados "environments" definir y comprobar la denominada "semantics" de las estructuras abstractas que él introducía en el sistema. Para ello, se elegía un "better approach". En lugar del "procedural approach" que complicaba la tarea

del diseñador se optaba por "descriptive mechanisms" como los de la técnica denominada "mathematical (denotational) semantics" desarrollada por el lógico matemático Dana Scott, profesor de CMU desde 1981, donde N. Habermann y Merrick Furst le ofrecieron un puesto.

Se trataba de introducir lo que los autores del GANDALF project llamaban " **a entirely new attitude towards the very role of a programming tool in the program creation and validation process. It should be emphasized, however, that the development of practical semantically based tools is a long-term goal requiring new theoretical insights about programming-language theory. Nevertheless, it is a goal that is essential to pursue**". (CMU-CS-Proposal 1984-86.:5-5).

El cambio era sustancial. Se trataba de pasar de concebir el programa informático de un "procedure" a una "derivation". El método tradicional de construcción de los programas había sido presidido por la frase: "GOTO". Se trazaba un gráfico ("flowchart") que describía la secuencia de pasos a seguir para la ejecución de la tarea, con la que se resolvía el problema. De ahí el término, "go-to". El programa terminaba cuando se cumplían las especificaciones u objetivos a cubrir.

El nuevo enfoque partía, por contra, de las especificaciones que debía cumplir el programa y por derivación se desarrollaban las implementaciones sucesivamente. Este tipo de programación se denominaba "inferential programming", dado que se concebía como una sucesión de inferencias derivadas de los objetivos a cumplir - El primero primaba la eficiencia ("efficiency") el segundo la claridad ("clarity"). Ambos procedimientos eran técnicas al servicio del mejorar el "programming".

GANDALF se pasaba a apoyar en un nuevo proyecto de investigación denominado ERGO PROJECT, presentado a una agencia distinta, la Office of Naval Research, para el periodo (1984-88) y dirigido por W.Scherlis y Dana Scott, cuyo objetivo era la construcción de un sistema experimental de "inferential programming", en concreto para construir los denominados "Semantically Based Programming Environments".

De esta forma, el programa de investigación básica inicialmente llamado "PROGRAMMING TECHNOLOGY" se subdividía en dos áreas distintas. En la "proposal" de 1987-90, se pasaban a denominar "PROGRAMMING ENVIRONMENTS" y "REASONING ABOUT PROGRAMS", la primera dirigida por Nico Habermann y la segunda por Dana Scott y William Scherlis.

De forma paralela, a principios de los 80s se había puesto en marcha la "SOFTWARE INITIATIVE", un programa del Departamento de Defensa, dirigido a resolver dos grandes problemas a los que se estaban enfrentando los sistemas de defensa no sólo de Estados Unidos, sino de toda la OTAN : la enorme complejidad de los sistemas de software y su elevado costo. El concepto de "programming" ya era insuficiente para abarcar el problema planteado en esos sistemas informáticos. Se imponía hablar ya de "software", y , más aún, de "complex software systems", así como de la educación de profesionales para el diseño y mantenimiento de dichos sistemas. Nació el "software engineering".

CMU firmaba el 28 de diciembre de 1984 un contrato con el Departamento de Defensa para establecer el Software Engineering Institute, como centro federal de investigación y desarrollo con la misión de abordar la solución de esos problemas.

Analicemos separadamente estos nuevos programas de investigación, que se pueden concebir como partes del área de "Programming Systems".

3.3.4. ERGO PROJECT : EL "DESIGN-BASED APPROACH".

En julio de 1983, Dana Scott y William Scherlis escribían un artículo titulado : " First Steps Towards Inferential Programming" (1983) publicado por el Department of Computer Science de CMU.

En él, se proponía la nueva investigación sobre la siguiente premisa: **"Our basic premise is that the ability to construct and modify programs will not improve without a new an comprehensive look at the entire programming process"**. (1983:2).

Se partía, pues, de pretender mejorar la habilidad de los programadores para construir y modificar programas informáticos. Para lo cual, estos profesores proponían un "research program" cuyo propósito inicial era comprender el proceso de programación: **"A initial goal of research must be to discern the basic structural elements of the process of programming and to cast them into a precise framework for expressing them and reasoning about them. This understanding must be embodied in a logic of programming--a mathematical system that allows us not only to reason about individual programs, but also to carry out operations (or transformations) on programs and, most importantly, to reason about the operations"**. (1983:2).

Scherlis manifestaba que su línea de investigación iba dirigida a comprender:

"understanding of the programming process--the methods by which we invent, specify, implement, adapt, and reason about programs and systems"(Scherlis. GSCS:41). Hasta ahora los programas de investigación se habían dirigido en Programming a mejorar esta actividad, mediante el diseño de nuevos lenguajes o ambientes. Eran programas aplicados.

Ahora se pretendía investigar como se diseñaban esos mismos programas. Sólo en psicología cognitiva, Herbert Simon y Allen Newell estaban intentando comprender también la mente humana vista como "problem solving", y no estaba centrada en la actividad de programación en especial sino en cualquier actividad de resolución de problemas. Con esta nueva línea de investigación cobraba fuerza el denominada "Logic Metaphor" (H. Simon, 1990:459) en Computer Science de CMU.

Scherlis y Scott veían el "programming" como una lógica, y de ahí se podía considerar a los programadores como lógicos derivando un programa de sus especificaciones originales mediante procedimientos de inferencia lógica. En el futuro, los programas según estos autores serían: " descriptions of program derivations".(1983:23).

William L. Scherlis era associate professor en Computer Science de CMU donde llegó en 1980, sirviendo en la actualidad en DARPA. Recibió la graduación en Applied Mathematics de la Harvard University en 1974 y el doctorado en Computer Science en Stanford en 1980.

Desde el principio, este profesor centró su investigación en "the practical and theoretical aspects of programming languages and systems, with a focus on the development of advanced environments to support the design and adaptation of software systems". (CMU-CS- FRG. 1989-90:73).

Por su parte, Dana Scott es Hillman University Professor of Computer Science, Mathematical Logic and Philosophy en Carnegie Mellon, a donde llegó en 1981. Se graduó en Mathematics en Berkeley en 1954 y se doctoró en la misma especialidad en Princeton University cuatro años más tarde. Es una autoridad internacional en estas áreas, siendo nombrado miembro de la National Academy of Sciences en 1988. Sus intereses son múltiples, en concreto en computer science van dirigidos, según sus propias palabras: **"to the development of denotational semantics of programming languages and the mathematical foundations of a suitable theory of computability"**. (CMU-CS-FRG. 1991-92:70). Su llegada a Carnegie Mellon reforzó la tendencia a una mayor formalización de la Computer Science, animando la construcción de una denominada "Logic Community" (Carnegie Mellon, Graduates

Studies in Computer Science, 1991-92:9) entre miembros los tres departamentos a donde este profesor está adscrito: Mathematics, Philosophy y Computer Science. En 1984-88 se desarrolló una primera fase del ERGO PROJECT, subtítulo "Foundations of Program Derivation/Research for Semantically Based Programming Environments". De 1989 a 1991 se ha desarrollado una segunda, donde el proyecto ha variado algo su título. Ahora se denomina: "Semantically Based Program-Design Environments". Se introducía un cambio, un nuevo enfoque denominado "Design-Based Approach", que no estaba en la propuesta inicial. Se trata de una nueva forma de abordar el paradigma del "inferential programming".

El equipo de ERGO formado por Peter Lee, Frank Pfenning, John Reynolds, Gene Rollins y Dana Scott, en 1988 en un papel conjunto lo explicaban. En el proceso de derivación de los programas por inferencia, no sólo se recogerían para su estudio las especificaciones y como se cumplían sino también las decisiones del programador o "design information" (CMU-CS- Ergo Project, 1988:1) que éste adoptara a lo largo de ese proceso. Estas decisiones no quedan normalmente representadas explícitamente en los programas, pero son claves para la adaptación, el análisis, la verificación y otras actividades en la vida de un sistema de software.

Este programa que tenía por objeto " a sound understanding of the process of programming", y descubría un aspecto oculto en las distintas teorías de programación: **"During the process of software development, a programmer makes many decisions. Conceptually, these decisions constitute a body of design information that describes how an implementation is related to the original specification or set of requirements...Unfortunately, descriptions of design decisions are not generally represented explicitly in programs, either as formal structures or informal comments, and thus are usually lost or forgotten...The primary need is to capture design information at the appropriate points during program development, and then to show how to use this information in an effective way".** (ibid1988:1-2).

En la propuesta realizada para la nueva fase de ERGO PROJECT a DARPA y a la Office of Naval Research ese mismo año, se incluía ya este enfoque y el proyecto de construir un denominado "engineering framework": el Experimental Program-Design Environment (EDPE). Su misión de "retention, formalization, and reuse of design information". La comprensión del proceso de programación, de la "logic of programming" localizaba una información hasta ahora perdida que

denominaba "design information", para lo cual era preciso construir un instrumento ingeniero para recogerla.

Esta investigación, de continuar, puede llevar a la conclusión que la actividad de programación es una actividad de diseño, esencialmente.

Simultáneamente, para el periodo 1987-90, Carnegie Mellon presentaba a DARPA dentro de su Proposal for Information Processing Research, una investigación básica sobre "Reasoning about Programs". Esta investigación era a la vez una investigación del area de Programming Systems y de Theory en el Department of Computer Science de CMU, llevada a cabo por profesores que trabajaban en ambas áreas como Stephen D. Brookes y Dana Scott.

Se afirmaba en la propuesta: **"The overall aims of the project involve basic foundational research into theoretical issues in the semantics of programming concepts, motivated by practical needs. In particular we want to develop semantically based tools for reasoning about programs including such issues as concurrency and abstraction"**. (1987-90:6-1).

Este era un programa de "basic research" sobre teoría de los lenguajes de programación o "programming-language theory".

La noción de "theory" es propia del sistema de conocimiento científico, como sistema cultural dominante en la actualidad. Las culturas primitivas y sus sistemas de conocimiento mitopoéticos, se basan en representaciones fundamentales que denominamos mitos, forma socialmente aceptada de organizar sus principales sistemas de creencias. Las teorías científicas son estas formas para las culturas científico-técnicas. Con todo, el fin perseguido por ambas es el mismo: explicar el mundo para poder dominarlo. Para Popper, **"Las teorías científicas son enunciados universales,...redes que lanzamos para apresar aquello que llamamos " el mundo": para racionalizarlo, explicarlo y dominarlo"** (1962:57). La meta del hombre de ciencia teórico es, según Popper: **"Encontrar teorías explicativas (si es posible verdaderas); es decir teorías que describan ciertas propiedades estructurales del mundo que nos permitan deducir, valiéndonos de condiciones iniciales, los efectos que se trata de explicar"**. (1962:59).

Pero, ¿qué significa "theory" para las computer cultures?. Significa lo mismo que para las culturas científicas restantes?. Ya hemos visto la diferencia de "mundos" que separan las ciencias naturales de lo que Simon denomina "sciences of design". Existirá también una diferente concepción de "theory" en ambos

mundos?.

De entrada, no puede explicarse un mundo que no existe, la teoría de los mundos artificiales tendrá que estar precedida por el diseño de dichos mundos. Ello lo volvemos a comprobar en este proyecto de "programming language theory".

Centrémonos en sus dos temas fundamentales de investigación : " fundamental programming language concepts" y "theoretical foundations".

Empezaremos por este último. El marco de referencia que se propone para desarrollar la fundamentación teórica de los lenguajes de programación se denomina: "Constructive Logic". (CMU-CS- Proposal 1987-90: 6-17).

Esta orientación proviene de la matemática constructiva, utilizada por los matemáticos aplicados, que sostiene que cualquier legítimo objeto matemático debe ser construído en una serie finita de etapas, comenzando un pequeño número de objetos primitivos, y procediendo paso a paso. Este enfoque está en directo conflicto con el platonismo que sostiene que los objetos matemáticos existen independiente de nuestras mentes, que han de ser descubiertos, no inventados. William F. Aspray, Jr. (1980) ha destacado la influencia de esta matemática constructiva en los fundadores de la computer science, Alan Turing y J.Von Neumann. Igualmente influye en la "constructive logic" de este programa de "Reasoning about programming" de CMU.

Según los autores del mismo: **"Logic cannot be used in a 'pure' form entirely without reference to intended applications. Programming language semantics, for example, already sets out preferred data types of objects, and proofs must be oriented to these kinds of defined structures. This causes the logician to take into account the explicit design of the programming language in working out proof strategies"**. (1987-90:6-29). El lógico se hace diseñador, adoptando un valor de la cultura tecnológica del ingeniero.

Ello empieza a cambiar el sentido de la "theory". De concebirse según la ciencia empírica , como un conjunto de enunciados que explican una realidad dada, estos investigadores pasan a darle una nueva función: **"The point is this: it is not true that theory can serve only descriptive purposes. Theoretical insights are more appropriately applied in a prescriptive role--as a means for achieving simplicity rather than alleviating complexity. It is our belief that this type of role must be adopted before the benefits of theoretical research can be felt extensively by the programming language community"**. (op. cit.1987-90:6-3).

Es bajo este marco de "constructive logic" que se trabaja en el proyecto para

construir una teoría unificada de los lenguajes de programación: **"The second area of research involves knowing how, under the framework of constructive logic, a unification can be achieved of the methods of recursion theory, proof theory, type theory, domain theory, and denotational semantics...This unification will have critical consequences for designers of systems that reason about programs"**. (1987-91:6-29).

Pero el enfoque del proyecto parte no de la "constructive logic" en general, sino una aplicación de ésta a la "computer science", en concreto, su aplicación a los lenguajes de programación, donde Dana Scott realizó junto con Christopher Strachey su contribución fundamental: la técnica de "denotational semantics" para la descripción de programas informáticos.

Dicha técnica consiste en construir definiciones semánticas en lenguaje matemático ("denotational") y manipular esas definiciones sobre lenguajes y programas informáticos. Este enfoque semántico formal acerca de los lenguajes de programación era una innovación. Hasta entonces había prevalecido el enfoque sintáctico, preconizado en lingüística por Chomsky, por el cual se analiza un lenguaje por su estructura sintáctica, libre de contexto y significado. El enfoque semántico se basa, por contra, en asignar significados o valores a las expresiones o programas en el lenguaje.

Dana Scott afirmaba ante la concesión del Turing Award, en 1977: **"From the late 1960's my own work has concentrated to give a better conceptual understanding of programming languages...What I try to do is to make some results from logic which seem to me to be relevant to computing comprehensible to those who could make use of them"**. (1977:635) y terminaba diciendo: **"The method of semantical definition does in fact work"**.(ib: 641).

En el texto no queda claro si se trata de una técnica o de una teoría que permite comprender mejor los lenguajes de programación, o de ambas cosas a la vez.

Dentro de este enfoque, "Reasoning about Programming" pretendía el desarrollo de su otro gran tema de investigación: los conceptos fundamentales de los lenguajes de programación : "type structure", "abstraction mechanisms (procedures, modules, abstract data types)" "concurrency", "nondeterminacy", "interference". (op. cit.: 6-4). Un principal concepto matemático que se planteaba desarrollar el proyecto era el de "types", concepto que se consideraba útil para construir lenguajes de programación. Esta investigación ha dado lugar a un nuevo proyecto

denominado "TYPES IN PROGRAMMING". (CMU-CSD-Basic Research in CS: 1989).

3.3.5. EL "TYPES IN PROGRAMMING PROJECT" : LA DENOMINADA "TYPE THEORY" COMO INSTRUMENTO DEL "PROGRAMMING".

TYPES IN PROGRAMMING (1990-1993) (CMU-CSD-1989). es uno de los cinco programas en "basic research" planteados por CMU a DARPA para el periodo en curso. Los otros ya los hemos citado: dos en Inteligencia Artificial, : "Integrated Architectures for Intelligent Systems" e "Image Understanding" y dos más en Programming Systems: "Object Management" y "Creating Graphical Applications". El equipo que lo lleva adelante está formado por Dana S. Scott, John Reynolds, Daniel Leivant, Robert Harper, Peter Lee y Frank Pfenning.

Las áreas de interés de Dana Scott ya las hemos descrito con brevedad anteriormente.

Un investigador destacado del equipo es John Reynolds , Professor of Computer Science en CMU. Es doctor en física teórica por Harvard, en 1961. Según el mismo indica en la Faculty Research Guide de 1991-92:

"My research centers on the design of programming languages and languages for specifying program behavior, mathematical tools for defining the semantics of such languages, and methods for proving that programs meet specifications". (CMU-CS-FRG, 1991-92:66).

Esta investigación la divide en dos partes:

"At present, my main interest is type theory .The last fifteen years have seen the discovery of a variety of type systems that have vastly enlarged our notions of what types are and how they might be used. My goal is to understand the semantics of these systems and to find a general concept of type of which they are all instances". (ibid.: 66).

"A second area of interest is the design, definition, and proof methodology of languages, such as Algol, that combine imperative constructs with a powerful procedure mechanism. This has led to the design of Forsythe, and extremely general and uniform Algol-like language that encompasses object-oriented programming" (FRG 1991-92:66).

Si nos fijamos en estos dos últimos objetivos, éste sería un programa de investigación tradicional como el de la física, primero descubrir ("what the types are"), luego utilizar esos descubrimientos para diseñar ("design languages"). Pero estos dos objetivos están precedidos por otro objetivo que J.

Reynolds lo sitúa en primer lugar, y por el cual su investigación se centra en el diseño de lenguajes de programación. El esquema no sería DESCUBRIMIENTO-DISEÑO, sino DISEÑO 1-DESCUBRIMIENTO-DISEÑO 2. Por su parte, Daniel M. Leivant, associate professor en CMU y doctor en Lógica por la Universidad de Amsterdam desde 1975, está especializado en aspectos fundacionales del computing y en "constructive mathematics": **"My research interests are in the foundations of programming and of computing."**(FRG, 1990-91:47).

Otro investigador del equipo es Robert Harper, Research Computer Scientist en CMU, y doctor en Computer Science por Cornell desde 1985. Según afirma Harper: **"My principal research interest is in the application of logic and semantics to the practice of program development. The fundamental tenet of my research is that the methods of mathematical logic may be usefully applied to problems in computer science, providing the conceptual basis for the development of reliable and elegant software systems"**.(FRG 1991-92:35).

Por su parte, Peter Lee, Assistant Professor de CMU , doctor en Computer Science por la Universidad de Michigan, e investigador también del proyecto ERGO, afirma:

"My research interests are in the design and implementation of programming languages. More specifically, I am interested in languages and compilation techniques based on principles from formal semantics and type theory. I specially like finding ways to make such languages and techniques practical to use in real world applications". (FRG. 1991-92:44).

Por último, Frank Pfenning, Research Computer Scientist de CMU y doctor en matemáticas por esta universidad desde 1987, afirma:

"My primary current research interest is type theory and its application to language design, in particular in the areas of logic programming and functional programming. I have also retained an interest in logic, automated theorem proving, formal program development, and programming environments". (ibid: 63).

Todos estos investigadores forman parte simultáneamente de la área de "Programming Systems" y de "Theory" en la School of Computer Science. Y vemos objetivos diferenciados en ellos. Mientras algunos les interesa primeramente el diseño de sistemas informáticos como a Reynolds, Harper o Lee, a otros les interesa en primer lugar, la comprensión de dichos lenguajes de programación (Dana Scott, Leivant, Pfenning).

Ello se refleja en los objetivos del proyecto "Types in Programming".

El propósito principal del mismo es el siguiente: **"We envision a unifying theory of formal systems that will help us to comprehend, experiment with, and unify our present understanding of programming"**(1989:1-1). Esta teoría se centrará en la "type structure of languages". Basado en lo cual, el proyecto diseñará varios lenguajes de programación: FORSYTHE, LEAP, y LF.

La motivación de la investigación según se afirma en el proyecto es :

"The Types in Programming Project (TIP) seeks to understand languages for programming and program specification by exploring the unifying concept of type structure. The TIP Project is a community effort in which the results of basic research on the foundations of programming languages are utilized as an enabling technology for more immediately practical systems efforts."(1989:7-1).

El esfuerzo de comprensión en la ciencia tradicional empieza por las preguntas clásicas formuladas por Thomas S. Kuhn en "La estructura de las revoluciones científicas"(1987): ¿qué existe, y qué es?. **"El descubrimiento de un nuevo tipo de fenómenos es necesariamente un suceso complejo, que involucra el reconocimiento, tanto de que algo existe, como de qué es"**. (1987:97).

Pero ¿cuáles son las preguntas de este proyecto de "basic research"?: **"The**

problem in programming systems is how things fit together. For example:

-Can lazy evaluation be applied to languages with assignment statements?

-Can such languages be constrained to permit aliasing to be detected systactically?

-Can constructive proof methods be extended to programs that produce unbounded output, have side effects, or use non-local transfers of control?

-Can domain theory and mathematical logic in a way that will encompass the definition of programming and specification languages and the validation of proofs methods?." (ib:7-1).

Pero la pregunta por el "can" es una pregunta propia de la ingeniería entendida como diseño. Estos "computer scientists" quieren algo más que aplicar simplemente la estructura matemática de "types" a la actividad de programación o "programming".

Pretenden desarrollar una teoría de los "types" ligada a esta actividad.

La noción de "types" designa un conjunto de valores agrupados en un conjunto o clase. Son conocidos en el lenguaje PASCAL, los cuatro "primitive types":

INTEGER, REAL, CHAR Y BOOLEAN. Los dos primeros representan información numérica, el tercero caracteres, y el cuarto, información lógica. Cada "type"

incluye un conjunto de valores ("values") o información simbólica y operadores ("operators"), esto es, operaciones a realizar con la información contenida en el programa.

Como afirman los autores, en el apartado " The Central Role of Types" de su propuesta: **"It is probable the case that the notion of types first found its way into programming languages mainly as a convenience to compiler writers...Later, however--for instance in the design of the language Pascal--types began to be viewed as software engineering tools in their own right. Thus, the use of types in program design, particularly for data abstraction (e.g. in CLU and Alphard), modularity (e.g., in Modula and Ada), and inheritance (e.g. in Simula), now prevails in most software development methodologies.**

Despite the success of these and other typed languages, little is known about the mathematical foundations of types, and even less about the real value of types in software development". (1989:7-3).

Este proyecto nos conduce a analizar más detenidamente la relación entre Programming Systems y Software Engineering en CMU, y la relación de Programming Systems y la área de Theory.

En primer lugar, Programming Systems es una área dentro de Computer Science en CMU. Sus miembros se consideran "computer scientists". Con todo, podemos apreciar que su actividad de investigación se centra en el diseño de lenguajes y sistemas informáticos, según un enfoque que es propio de la ingeniería, no de la ciencia. Sin embargo, no se consideran oficialmente ingenieros.

Por otro lado, desde los años 60s se viene advirtiendo de la necesidad creciente de formar una ingeniería del software, dado que la demanda excede con mucho a la oferta de profesionales preparados. Un estudio de la Office of Technology Assesment estima en 1990 un deficit de entre 50.000 y 100.000 profesionales del software sólo en Estados Unidos.

Esta fue una de las razones principales para fundar el Software Engineering Institute en 1986, en CMU. Pero sus relaciones con la Computer Science de CMU se basan en un modelo que parte de la antigua división : Ciencia-Ingeniería, que no parece encajar fácilmente en una "computer culture".

Respecto a la relación "Programming Systems" y "Theory", apreciamos que hasta ahora la denominada "Theory se ha desarrollado por detrás de" Programming" , al menos en CMU. Como veremos, "Theory" en CMU se basa en áreas como Algorithms, Complexity y Logic, que son ramas de la matemática que tienen relación con

Computer Science pero no son aún una teoría de qué es "Computer Science". Son este sentido instrumentales, no finales a "Computer Science".

Empecemos con la relación del Computer Science de CMU con el Software Engineering Institute.

3.4. EL SOFTWARE ENGINEERING INSTITUTE DE CMU.

3.4.1. EL SOFTWARE ENGINEERING INSTITUTE Y LA SCHOOL OF COMPUTER SCIENCE DE CMU.

Cercano al campus de CMU, pero separado del mismo, se encuentra el Software Engineering Institute (SEI), fundado en 1984, dirigido por Larry Druffel. Este es un "federally funded research and development center", un centro nacional en investigación y desarrollo, el primer instituto nacional, en ingeniería del software en Estados Unidos, administrado por Carnegie Mellon bajo un contrato del Departamento de Defensa.

Norman Gibbs, director del área de educación de la institución, define así sus fines: "The objective of the SEI is to provide leadership in software engineering and in the transition of new software engineering technology into practice". (CMU-CS-FRG 1990-91:31).

El SEI cuenta con una plantilla total de 273 personas en el curso 1990-91, casi de iguales proporciones que la área de Computer Science de la SCS. A su vez en investigación es el tercer centro que más dinero obtiene de todo CMU: 20,9 millones de dólares en 1989, sólo aventajado por la SCS (39,249 millones) y el Carnegie Institute of Technology (22,800). (Carnegie Mellon Facts 1990:64).

Una novedad que aporta el centro es la institucionalización de una nueva profesión: el "software engineer", distinta del computer scientist, y el establecimiento de un Master junto con la School of Computer Science dedicado a formar este tipo nuevo de profesional.

Esta nueva ingeniería ha vuelto a plantear la necesidad que el diseño esté en la base del curriculum de este profesional (Mary Shaw. 1990. "Prospects for an Engineering Discipline of Software.).

La formación de esta profesión responde, según Norman Gibbs, a una distinta misión del software engineer, que lo diferencia del computer scientist: "**Since software engineers work in a product-oriented field, they require a different kind of education than that typically provided by research-oriented computer**

science departments. To meet the needs of an emerging profession, the founders of the SEI envisaged a graduate professional degree analogous to an MBA (master of business administration). " (1989: The SEI Education Program: 595). De este Master in Software Engineering (MSE), de dos años de duración, se encarga la School of Computer Science.

Hasta ahora, los computer scientists tenían el monopolio de la educación superior en ordenadores compartida en menor medida por los ingenieros electrónicos. Ahora está naciendo una nueva ingeniería, denominada "software engineering", que empieza ya a considerarse titulación superior aunque no aún se le permite dar doctorados. A su vez ésta coincide en el tiempo con el surgimiento de otra ingeniería nacida dentro de la Artificial Intelligence, el "knowledge engineering"(Stefick, Conway. 1982).

Norman Gibbs diferencia claramente esta profesión del computer scientist: " There are a large number of people who equate software engineering with programming and argue that software engineering is simply that part of computer science concerned with programming-in-the-large...This view is held specially in the academic community which tends to build software systems to demonstrate proof-of-concept, not products...Professional software developers are interested in building products which generally are delivered in multiple releases and take on a lifetime of decades" (1989:596).

Esta orientación sería la misma que hemos visto en el MACH Project: "Products, not only ideas". La orientación del SEI sería distinta de la de la mayoría de la School of Computer Science, basada en la "concept demonstration".

Segun el director del SEI: "Software engineers build on and apply the body of knowledge known as computer science; they must be facile at programming-in-the-small, and they need additional education in many of the following: software development processes, software project management, requirements analysis, technical communication, computer engineering, systems engineering, embedded and real-time systems engineering, configuration management, quality assurance, formality, performance analysis, metrics standards, verification and validation, security, human factors and specialized applications domains". (1989:596).

Recordemos que ingenieros de Carnegie Institute of Technology, como el profesor Norman Chigier, sostienen que la ingeniería es una disciplina más compleja que la ciencia: "Engineering is really more complex than science. It can't obey all

the rules of science because it deals with systems that are less easy to analyze. Engineers have the challenge of working with practical systems--end-products"(5 feb.1983: Focus:).

De momento, el "software engineering" se estudia aún dentro de la School of Computer Science.

La creación de la figura del "software engineer" ha surgido como fruto de la presión no de la Academia sino del gobierno y la industria, demanda especialmente apoyada por el Departamento de Defensa. Su surgimiento vino provocado por la llamada, "crisis del software", "software crisis".

Estas instancias vienen solicitando que se den cursos de "software engineering" a los estudiantes de universidad, a lo que la Academia se ha resistido. Según Norman Gibbs, hablando esta vez como profesor de Computer Science: "We resist by arguing that the best education remains a solid major in computer science followed by graduate professional education such as an MSE program... We run the risk of prematurely fragmenting the still young and evolving field of computer science". (1989:596).

En este campo, los computer scientists estarían retrasando el surgimiento de una ingeniería informática, como ocurría en Estados Unidos en 1861 cuando las universidades tradicionales no admitían la ingeniería como carrera académica, lo que provocó el nacimiento del Massachusetts Institute of Technology, como institución independiente de Harvard y la universidad tradicional.

Hoy, los problemas para constituir el "software engineering" como profesión independiente son aún más profundos.

3.4.2. LA "SOFTWARE CRISIS" Y LA "SOFTWARE INITIATIVE" DEL DEPARTAMENTO DE DEFENSA.

En el estudio "Informatics for a New Century: Computing Education for the 1990s and Beyond" (1990), Mary Shaw, profesora de CMU y experta en el tema, apunta el dato antes citado: En Estados Unidos, faltan hoy entre 50.000 y 100.000 puestos de profesionales del software en estos momentos. Otros profesores como Philip Miller elevan la cifra hasta 1 millón. (1987:x).

La importancia de este déficit se agrava si se tiene en cuenta que el software tiende a dominar el precio de la informática. El principal consumidor de informática del mundo, el Departamento de Defensa de los Estados Unidos que gastaba a finales de los 80s, según Angel Jordan, una cifra anual de 12.000

millones de dólares, de los cuales el 80% representaban gastos en software.

La necesidad de expertos profesionales en software se detectó ya en 1968 en la primera Conference International on Software Engineering en Garmish, en la entonces Alemania Occidental, organizada por la OTAN. Fue allí donde se acuñó por primera vez el término "software engineering", más como una aspiración que como una realidad.

Síntomas de esta "software crisis" eran principalmente dos: la falta de calidad de los sistemas de software y su cada vez más elevado precio. Respecto al primer punto, el llamado "mantenimiento" ("maintenance") de un sistema de software a lo largo de su existencia puede consumir un 60% de su coste total. Ello incluye mejoras, rediseño, reparación de errores, adaptación a los cambios de ambiente, entre otras tareas. Por otra parte, ya hemos señalado su alto precio. El coste anual del software que compraba el Departamento de Defensa en 1980 era de 4.000 millones de dólares. En 1986 subió a 10.000 millones y en los 90s se preveía que fuera de 30.000 millones. (Lieblein 1986:734). La demanda excedía con mucho a la oferta, lo cual hacía aumentar aceleradamente su precio.

En 1973, del total de este coste la mayor parte, un 56%, se gastaba en lo que se denominan "embedded computer systems" o sistemas informáticos insertos en sistemas de defensa más amplios cuya finalidad no es informática, por ejemplo, un sistema de radar, un sistema de misiles, o una nave espacial.

Estos sistemas comparten una serie de características comunes: 1. Son programas informáticos grandes, con miles o millones de líneas de código. 2. Tienen una larga vida, de entre 10 y 15 años. 3. Cambian continuamente dado que cambian sus objetivos. 4. Están constreñidos en sistemas de hardware delimitados. y 5. Han de ser altamente seguros y tolerantes a los fallos. (Booch, G.,1987:15)

Por esas fechas, el Departamento de Defensa empleaba 450 distintos lenguajes de programación, lo cual hacía imposible cualquier adiestramiento, dado que cada proyecto podía crear su propio lenguaje. Al mismo tiempo, esta situación no permitía ninguna transferencia de tecnología entre proyectos ni de estos a la industria. Cada sistema dependía de un particular vendedor.

A su vez la proliferación de lenguajes hacía imposible la creación de herramientas de software, como los ambientes de programación.

Para hacer frente a esta "crisis", el "Department of Defense" (DoD) empezó a organizar una iniciativa centralizada, que se llamaría la "Software Initiative" (Lieblein, 1986). En 1975 formó el denominado "High Order Language Working

Group" (HOLWG), dependiendo de la "Office of Secretary of Defense", y dirigida por el teniente coronel William Whitaker, del Ejército del Aire (USAF). El objetivo de este grupo era establecer los requerimientos para el lenguaje de programación a usar por el "Department of Defense" en todos los llamados "embedded software systems". A este grupo se invitó a representantes de Gran Bretaña, Alemania y Francia, países pertenecientes a la OTAN.

Este grupo tras consultar durante varios años a más de 200 personas, representando a 85 organizaciones del "DoD", 26 contratistas industriales, 16 universidades y otras 7 organizaciones, acabó definiendo unos requerimientos en un documento llamado STEELMAN para el diseño de un único lenguaje de programación, el denominado "HOL" ("High Order Language"). Estos requerimientos exigían un lenguaje con considerable poder expresivo para cubrir una amplia variedad de aplicaciones.

Finalmente, después de un concurso internacional, el proyecto del francés Jean Ichbiah, de la empresa Honeywell/Honeywell Bull fue el elegido, ayudado por un equipo internacional formado por franceses, ingleses, norteamericanos y alemanes. Para cumplir los llamados requerimientos STEELMAN, el equipo diseñador del mismo se fijó como primer objetivo de su lenguaje, el garantizar la "reliability and maintainability" de los programas, que se pudiera confiar en ellos y que fueran mantenibles. En segundo lugar, se marcó como objetivo ayudar a la actividad del programador para tratar la complejidad de los sistemas a diseñar: "Concern for programming as a human activity". El tercer objetivo fue su eficacia: "Efficiency". (Booch:54).

El resultado fue "ADA", lenguaje denominado así en honor de la hija de Lord Byron, Lady Lovelace, matemática y colaboradora de Charles Babbage. Fue terminado en 1980.

ADA no es sólo un lenguaje informático más. Grady Booch lo denomina una cultura: "The ADA Culture" (1987). En febrero de 1983 fue inscrito como lenguaje standard: ANSI/MIL-STD-1815 A ((Department of Defense, 1983) del Departamento de Defensa de los Estados Unidos. Es el lenguaje de programación informática del principal consumidor de software del mundo. Fue adoptado por la OTAN como estandar para sus sistemas de información. Y en marzo de 1986, 10 países de la OTAN firmaban acuerdos para desarrollar ambientes basados en ADA.

La participación de CMU en el mismo fue significativa, aunque secundaria. Nico Habermann participó en las revisiones del mismo. Este lenguaje, por otra parte,

cultimaba la familia de lenguajes iniciada a finales de los 50s con ALGOL, diseñado entre otros por Alan Perlis. Familia más tarde continuada con el lenguaje PASCAL.

En 1983, Nico Habermann escribió un libro con Dewayne Perry titulado: "ADA for Experienced Programmers", como manual de instrucción en este lenguaje.

ADA sólo fue el primer paso de la Software Initiative. Otros dos siguieron y le complementaron: el programa STARS y, finalmente, el Software Engineering Institute de CMU.

El programa STARS (Software Technology for Adaptable, Reliable Systems), iniciado en 1983, es un programa que tiene ADA como pieza clave y se dirige a mejorar la capacidad tecnológica de software del Departamento de Defensa, en particular la de sus llamados "embedded software systems". El teniente coronel, y doctor en Computer Science, Larry Druffel, director del SEI, jugó un papel destacado en su diseño.

La novedad que aporta este programa es que se basó no sólo en fomentar la R&D para producir una nueva tecnología para uso del gobierno, como suelen hacer otros programas federales, sino que además planteó dos novedades : 1. la introducción de esta tecnología en el proceso de desarrollo o "system development process" de sistemas de defensa reales y 2. su transferencia a la industria.

Como afirma Edward Lieblein: " The STARS program now incorporates two important new dimensions: technology insertion and industry leadership."(1986:738). Es por lo tanto un programa que tiene tres áreas distintas: 1. "R&D". 2. "Technology Insertion" y 3. "Technology Transfer".

La primera es la tradicional que se cubría mediante los llamados "Technology Projects". Estos se dividían en dos categorías: los destinados a la denominada "reusability and adaptability", esto es a ser reutilizados y adaptados a otros sistemas, y aquellos dirigidos a la "software reliability", o la seguridad. El SEI estaba encargado de establecer los standards para el uso en la construcción de los componentes de este software.

La segunda función era nueva. ¿Qué significa "technology insertion"?

Principalmente, la introducción de la nueva tecnología en los actuales "embedded software systems" del DoD. Hasta ahora, la tecnología se pensaba que finalizaba su ciclo cuando generaba un producto nuevo. El problema es que el software que necesita el DoD ha de ir insertado, introducido en sistemas de defensa

determinados que pueden ser o no nuevos. Ello obligaba a introducir un nuevo concepto: "technology insertion".

Y para ello, el programa STARS introducía un nuevo tipo de proyectos: "shadow projects". Estos proyectos se mostrarían anualmente en público para fomentar su difusión.

El término "shadow" indica que era proyectos concebidos como la sombra ("shadow"), de los reales proyectos de construcción de esos sistemas. Se preveían en áreas como comando y control, armamentos, o sistemas de inteligencia que no estaban usando ADA. Algunos de estos proyectos se desarrollan actualmente en el SEI. De ahí las críticas del movimiento pacifista de Pittsburgh contra este instituto.

Por último, la "industry leadership", o transferencia tecnológica, se conseguía a través de "Product Development Projects" o proyectos contratados con empresas seleccionadas para convertir la tecnología generada en STARS en "prototype advanced software engineering environments", o ambientes para la ingeniería del software avanzada. Recordemos que GANDALF era uno de estos ambientes.

El objetivo era atraer inversiones del sector privado en herramientas de software basadas en ADA, y establecer este nuevo mercado, que una vez consolidado produciría avanzados productos, de alta calidad, a precios competitivos.

Recordemos que la visión tradicional afirma que el ciclo tradicional de innovación empieza en la ciencia que produce la universidad, esta transfiere estos conocimientos a la industria que los aplica y crea la tecnología, y finalmente se distribuye en el mercado donde el consumidor la compra. La denominada "innovation engine" (Dertouzos, 1988).

Este es un ciclo que podríamos denominar científico-económico, pues se basa en que el modelo comienza con el avance de los descubrimientos científicos del "orden natural", orden aceptado por la propia economía política clásica, la técnica aplica estos descubrimientos en forma de productos, y el consumidor compra estos productos en el mercado con el objeto de satisfacer sus necesidades. Esta visión fisiocrática del ciclo económico ha servido en economías de bajo nivel tecnológico o "low technology industries", basadas en el uso extensivo de mano de obra y una base tecnológica basada en la fuerza mecánica o electro-mecánica. Pero cuando la economía depende cada vez más de la "high technology", la riqueza de las naciones depende más que del "valor

trabajo" del valor trabajo de innovación, esto es, del trabajo inventor de tecnología, y de tecnología con contenido intelectual cada vez más elevado. El ciclo de la Software Initiative no obedece al clásico ciclo de innovación. Lo inició una institución pública como el Departamento de Defensa al decidir que quería unos determinados sistemas de defensa basados en un software informático fiable y económico, que le resolvieran un problema detectado. Para ello, determinó que necesitaba de un lenguaje de programación único y fijó los requisitos del mismo ("requirements") . A continuación seleccionó a los técnicos entre la comunidad internacional de ingenieros del software que lo podían construir , eligiendo entre ésta a los más capaces técnicamente para esa tarea. Una vez diseñada la tecnología, la citada institución pública inició el fomento de su generación, construyendo un mercado para ello, animando a las empresas para que trabajaran dentro de esa nueva tecnología .

Por último, decidió crear una nueva disciplina académica, el software engineering, y una institución donde se educara a generaciones en esa nueva profesión.

Finalmente, el ciclo terminó por influir en las propias universidades de dos formas: 1. Empezaron a cambiar sus curriculum educativos e introduciendo títulos en Software Engineering y 2. Empezaron a orientar su investigación básica en función del nuevo lenguaje creado, por lo cual recibieron una financiación proporcional a su esfuerzo. Este ciclo no es distinto al que dió lugar al final de la II Guerra Mundial a la invención del ENIAC, el primer ordenador digital.

En concreto, ¿cómo afectó a CMU?. La "ADA culture" en esta universidad se puede detectar en varios niveles: tanto de investigación como educativos.

Introducida por Nico Habermann, director del departamento de computer science desde 1979 y primer director del SEI en 1984, en el terreno de investigación ha influido en proyectos como GANDALF para la generación de ambientes de ingeniería del software, así como en el proyecto "OBJECT MANAGEMENT", que continua al anterior.

A su vez, proyectos como "TYPES IN PROGRAMMING", se basan en uno de los aspectos fundamentales del lenguaje ADA, los "types". Una de las características de esta estructura matemática es que permite un mayor nivel de abstracción del programador, y esta mayor abstracción es necesaria para la solución de un problema fundamental del software engineer: como tratar con la denominada

"complexity" propia de los "embedded computer systems", también llamados "complex software systems". Como afirma Mary Shaw: "**The major practical problem for software development and maintenance is managing the complexity of software systems....A dominant theme in this growth is the development of tools dealing with abstractions...During the late 1970s, most research activity in abstraction techniques was focussed on one particular topic, data types.**" (CMU-CS-FRG 1991-92:72).

Por otra parte, en el terreno educativo, Nico Habermann fue impulsor del proyecto "ADA-EDU", más tarde llamado "CS-EDU". Su objetivo general era producir un ambiente de programación basado en ADA en un Macintosh II por un precio asequible: 200 dólares. Una de las partes de este proyecto era el MacGnome Project, dirigido por Philip Miller, y orientado a construir ambientes de programación apropiados para la enseñanza. Philip Miller es, a su vez, el autor del libro "Programming by Design"(1987), libro básico del primer curso de introducción en Computer Science en CMU, basado en el lenguaje PASCAL, cuyo estudio permite una introducción al lenguaje ADA, al ser ambos lenguajes algorítmicos, o "procedural languages".

Esta influencia de los militares en la creación de la computer culture no es nueva. Ya la analizamos en la formación del programa de investigación en Information Processing de CMU. Como afirmaba Angel Jordan : "**En Estados Unidos, Defensa ha sufragado la investigación básica. Y todos los descubrimientos tecnológicos como, por ejemplo, la informática, han sido patrocinados por centros dependientes de la Defensa...Después de la II Guerra Mundial, los principios de acción, tanto en Estados Unidos como en la Europa aliada...la investigación básica y aplicada, en ciencia y en tecnología, tenía como objetivo el desarrollo de la defensa**". (EL PAIS,16-12-90: 34).

Con todo, se han producido cambios importantes en este proceso. En 1945, se llamó a los científicos físicos a construir la bomba atómica. Esta comunidad ha dominado y dominan la política de la Science and Technology de Estados Unidos durante cuarenta años.

Durante ese periodo, la informática, una disciplina secundaria, instrumental hasta hace pocos años, se ha ido asentando. En sus primeros 40 años, influida por el modelo de los físicos, los matemáticos y de los científicos sociales que se han asociado a ella (Von Neumann, H. Simon,...) , se ha definido como una ciencia, "computer science".

En los años 80s, la SOFTWARE INITIATIVE, empezó a favorecer la creación de una área expresa de "software engineering", distinta al "computer science", dado que había un problema de alcance internacional que esta última ciencia no resolvía. CMU se había destacado por una orientación hacia el software y tenía un estilo de trabajo ingeniero, orientado al "programming". Y CMU ganó el concurso convocado para construir un centro nacional dedicado a la ingeniería del software. No obstante, sus relaciones con el Computer Science no son todo lo estrechas que en un principio se esperaban, según Angel Jordán.

La crisis del software tiene más de 20 años en Estados Unidos, y no se ha resuelto. Por el contrario no ha hecho más que agravarse. Parte de un desajuste entre la demanda de software y la escasa oferta del mismo. Es una crisis en la que influye el modelo de Science and Technology (S&T) seguido hasta hoy. Este modelo podría estar ahora retrasando el desarrollo de la tecnología informática, y en concreto, ahora un aspecto clave de la misma: la tecnología del software. Podría contribuir a su solución el que se priorizara el diseño y la ingeniería como modo de investigación por delante de la ciencia del ordenador, diferenciando dos tipos de investigación en esta nueva ingeniería, la básica y la aplicada sobre diseño de software informático.

El SEI es un intento de afrontar estos problemas.

3.4.3. EL SEI Y SU RELACION CON PROGRAMMING SYSTEMS DE SCS.

La tercera pieza clave de la Software Initiative fue la fundación del SOFTWARE ENGINEERING INSTITUTE. (SEI)

El 28 de diciembre de 1984, después de una dura competición nacional, el gobierno contrató con Carnegie Mellon establecer en Pittsburgh el SEI como nuevo "Federally Funded Research and Development Center". Angel Jordan, "provost" en esa época de CMU, y Nico Habermann, decano de la SCS, jugaron un papel destacado en su consecución, junto con otros miembros de la escuela como Edmund Clarke , Duane Adams, Mario Barbacci, Mary Shaw, Allen Newell y Howard Wactlar. El SEI tiene tres tareas: "technology transition", "research and development" y "education". Su misión principal es actuar de puente entre la academia y la industria en la área del software. Por lo tanto, su actividad principal es la primera.

Para Mario Barbacci, Senior Research Computer Scientist del SEI, y a su vez de la SCS, afirma que: " **El SEI es un instituto de transferencia tecnológica del**

software. La educación es una forma de transferencia tecnológica importante".

(Entrevista 21 feb. 92).

El SEI es un instituto para la ingeniería del software. Uno de los problemas que tiene es que este campo aún está poco definido. Cuando aún existe una indefinición sobre el carácter de la computer science, se plantea ahora en la "computer culture" la necesidad de abrir un nuevo campo, denominado "software engineering".

Esta disciplina nace de las necesidades prácticas antes referidas, y detectadas por primera vez en 1968. Pero en su desarrollo, esta área plantea problemas que van más allá de los que había en su inicio.

Un primero es la "technology transition". El "DoD" pretende que la tecnología del software, construída en lenguaje ADA, ha de ser utilizada por la industria. La particularidad de este modelo de transferencia es que no parte de la comunidad académica. El lenguaje ADA no fue el resultado de una investigación básica académica. Este lenguaje nació como encargo de el DoD a un grupo de software engineers de la empresa francesa CII Honeywell Bull Internacional. Se construyó al margen de la universidad y ésta manifiesta, de momento, poco interés en el uso de este lenguaje. La Inteligencia Artificial programa en lenguaje LISP, y la de Programming Systems en CMU en lenguaje "C", que es el del sistema operativo UNIX.

La industria, por motivos comerciales, a su vez, mayoritariamente, no programa en ADA sino en lenguaje "C", y ahora se está volcando en el desarrollo del "object-programming", otro sistema de programación pensado para el gran público, aunque, no obstante, se basa en principios como el de la modularidad utilizados en ADA.

Un segundo problema que ha de resolver el SEI, ligado con el anterior, es el de la "education" de los nuevos profesionales en esta nueva ingeniería. El SEI no es una institución que pueda impartir títulos de software engineering, es un laboratorio federal. Los títulos académicos los ha de impartir la School of Computer Science.

Sin embargo, este laboratorio se da la paradoja que se llama "Software Engineering Institute", tal y como si fuera una escuela de ingeniería, no pudiendo serlo. La contradicción es que tiene la vocación de animar a la formación de dicha profesión, pero no está reconocido académicamente.

El SEI refleja una gran necesidad por parte del gobierno y de la industria de

programadores cualificados y en gran cantidad, y educados en la "ADA culture". Pero no son esas las exigencias de la academia. La comunidad nacional académica en Computer es la "Computer Science Community", y ésta no parece interesada en alentar ahora una ingeniería que la relegue a un plano meramente teórico. Los datos son éstos. En 1987, las universidades norteamericanas produjeron sólo 466 doctores en Computer Science, el 2% del total de doctores en ciencia e ingeniería producidos ese año. La mitad de los mismos fueron a parar a estudiantes extranjeros. Por contraste, 845 doctores se concedieron en Matemáticas. (Dertouzos1988: 36-37).

En 1984, la demanda de doctores en Computer Science por la industria se valoraba en 1300 anuales (Hamblen,1984). Pero esta demanda es sólo una mínima parte de la que se precisa en programadores y especialistas en computing. A principios de los 80s, el National Science Board (1989) valoraba ya el déficit en 140.000 analistas y programadores para ese año. Por último, pronosticaba que en el periodo 1988 y 2000 , la ocupación que se espera que crezca más es la de "computer specialist" con 197.000 nuevos empleos, siendo la que tendrá el más rápido crecimiento de todas los empleos en Ciencia y Tecnología. (un 62%). La segunda ocupación con mayor crecimiento será la de ingeniero eléctrico/electrónico con 162.000 nuevos empleos. (ibid: 77).

El hecho es que más de 100.000 plazas de profesionales del software en Estados Unidos están sin cubrir y los 126 departamentos de Computer Science del país no pueden resolver este problema.

Una situación de enorme carencia de técnicos e ingenieros se vivió en Estados Unidos, tras la Guerra Civil , al comienzo de su desarrollo industrial. A principios de siglo pasado, no pasaba de 2 el número de ingenieros por cada estado. No existían escuelas de ingeniería industrial. Esta situación exigió cambios. Las reticencias de Harvard a la presión de la industria para formar ingenieros al nivel de las necesidades del país llevó a la formación en 1861 del Massachussets Institute of Technology. En 1862 el Congreso pasó la Morrill Act que garantizaba ayudas a los estados para establecer escuelas de "mechanical arts". Se favoreció un nuevo tipo de escuela basada en la idea de la "technical education". Las universidades se adaptaron y crearon los primeros departamentos de ingeniería en su interior. En 1917 había ya 126 escuelas de ingeniería. Ochenta años más tarde, en 1986, Estados Unidos ha graduado 77.000 ingenieros. Sin embargo, por estas fechas, Japón, con la mitad de la población, acaba de

graduar una cifra similar: 73.000. (op. cit.1989:5). Un dato significativo es que en 1989, el Japón superó por primera vez a Estados Unidos en el número proporcional de investigadores en R&D por población activa, 67 por 10.000 personas, comparados con 66 por Estados Unidos.

Lo que el SEI parece indicar es que hace falta una nueva ingeniería, una "high tech engineering", que alimente una explosión de la industria del software, y el conocimiento básico de la misma se llama: "design".

El Computer Science and Technology Board (Dertouzos, 1988) afirmaba en su folleto titulado, "The National Challenge in Computer Science and Technology": "

Unlike other products, the essence of software is in its design, which is inherently an intellectual activity...Accordingly, the cost of producing software is dominated by the costs of designing it and making certain that it defines a correct procedure for performing all of its desired tasks...".

(1988:46). Resolver la "software crisis" está asociado a resolver el problema del "knowledge is design", lo cual es un problema cultural: legitimar el diseño como conocimiento valorado socialmente al mismo nivel o más que la propia ciencia. La "software crisis" no es sólo una crisis del software. Refleja la limitación del modelo de investigación basado en Science & Engineering

3.4.4. LA INVESTIGACION Y DESARROLLO DE "EMBEDDED SOFTWARE SYSTEMS": EL PROBLEMA DE LA "COMPLEXITY".

Por último, el SEI ha de abordar el problema de la investigación en "complex software systems". Sistemas complejísimo y de carácter estratégico, que sostienen el funcionamiento de los grandes sistemas de defensa, de exploración del espacio, de las telecomunicaciones y sistemas de transporte aéreo del país. Pero esta investigación requiere el trabajo en equipo, interdisciplinar, coordinado, con objetivos a largo plazo. Y ante todo, requiere que este tipo de investigación esté legitimada como una investigación valorada intelectual e institucionalmente. Y no lo está. Se sigue viendo al programador como un mero practicante de los dictados del computer scientist.

La SCS, a excepción de los programas acogidos a DARPA, sigue dedicándose a proyectos más propios de la academia. En 1991-92, en la SCS existían 99 proyectos distintos, para un total de 83 profesores. De éstos, sólo 14 reunían 3 o más de tres profesores: Image Understanding, i Warp, Knowledge Based Machine Translation, Learning Robot, MACH, Mercury (Electronic Library Project), NAVLAB,

Nectar, Planetary Rover, SOAR, Speech Understanding, Types in Programming, VLSI, y Warp. Y recordemos que esta escuela es uno de los centros caracterizados por tener grandes programas estratégicos, lo cual no es el caso de la mayoría de los 126 Departamentos de Computer Science del país.

Gran parte de los proyectos citados son realizados por programadores que no cuentan como "faculty", o profesor titular. En 1990 habían catalogados un total de 68 "programmers" en la SCS, que son la característica específica que diferencia esta escuela de un mero departamento universitario. Esta masa semianónima permite a esta escuela construir sistemas como MACH o NECTAR que funcionan en el denominado mundo real o "real world" o y no meros esquemas ideales al modo académico mayoritario.

Estos problemas se reflejan en la área de investigación.

La concepción dominante en la SCS acerca del software aún lo denomina "programming systems". Por "programming" se entiende normalmente la actividad de escribir las instrucciones que el ordenador ha de ejecutar. Hemos visto que varios proyectos de la SCS tratan de proporcionar herramientas o "tools" que faciliten esta tarea de programación. Pero la ingeniería del software es más compleja que el mero hecho del "programming". El concepto "software" se ha separado del de "programming", al considerar que éste último se ha quedado estrecho.

"Software engineering" se define habitualmente con el nombre "programming-in-the-large", para distinguirla de la actividad usual del programador o "programming-in-the-small".

Mario Barbacci, Senior Research Computer Scientist, trabaja ambos centros, el SCS y el SEI, y explica así la diferencia: **"Software no es programming. La mayoría de los problemas en software no vienen del hecho de programar o escribir las instrucciones o código sino de dos fuentes: por un lado, de las especificaciones, de la falta de documentación adecuada para determinar los requerimientos que ha de tener el software, y por otro, de la falta de verificación, de validación del sistema una vez construido"**. (Entrevista, 21 feb. 1992).

Para este investigador, la SCS se ha quedado en las mismas cuatro áreas de investigación que tenían cuando él entró en ella en 1974. En suma, se ha quedado atrasado.

La investigación en el SEI no se centra en el concepto de "programming systems"

sino en el de "embedded software systems". Estos sistemas son enormemente complejos, se espera que duren décadas, y están en continuo cambio. Se podrían comparar con una construcción artificial compleja como una ciudad o una red viaria.

Esta duración conlleva que el diseñador que los construyó será distinto del que años después corrija sus errores y mantenga el sistema en funcionamiento. Y ello sólo se podrá hacer si está explícito y claro el historial del sistema, sus especificaciones, las instrucciones de su código, y sus sucesivos tests de verificación.

Estos ingenieros del software no trabajan sólo con ordenadores, sino con sistemas complejos que incluyen el ordenador pero no se reducen a él. Según Barbacci, lo más cercano a esta actividad es la área de Robotics, pues como los robots, estos sistemas pueden tener un procesador incorporado, pero sin tener la forma de un ordenador.

La complejidad de estos sistemas es una de sus principales características. Para poder tratarla han tomado un principio básico de la ingeniería: la modularidad, la descomposición del proyecto a realizar en piezas, y la solución independiente de cada uno de los problemas. Ello permite construir sistemas complejos a partir de módulos simples.

La complejidad no es un problema sólo de la ingeniería. Ciencias como la biología o las ciencias sociales han de tratar con sistemas complejos, estructurados, jerárquicamente o no. Pero el enfoque seguido por la ingeniería es distinto al de la ciencia. Mientras ésta trata de comprender el funcionamiento del sistema mediante un análisis y síntesis explicativos, esto es, descomponiéndolo en partes y recomponiendo luego el conjunto en forma de síntesis, la ingeniería trata de diseñar dichos sistemas, utilizando la descomposición como técnica para realizar una síntesis constructiva del sistema. Se podría afirmar que la explicación científica también es una construcción, y así es, pero exclusivamente explicativa de una realidad dada. La síntesis ingeniera es una construcción en el sentido completo, teórico y práctico de un nuevo sistema.

Este "embedded system", por otra parte, ha de actuar en el mundo real y en tiempo real. Ha de funcionar y para ello el software engineer ha de verificar que su funcionamiento sea correcto. Para Barbacci: **"Los computer scientists de la SCS no verifican. No necesitan mantener un sistema durante años"**. (ibid:).

En suma, los proyectos del software engineer han de seguir una metodología de desarrollo propia de la ingeniería: "specification, design, implementation, testing". (Aquí el concepto "design" se toma en el sentido restringido del término. En un sentido amplio, veremos a continuación que el llamado "design" es el método que se considera propio de la ingeniería).

Junto a esta metodología, el software engineer tiene otros aspectos que los diferencian de otros proyectos de programming. Se han de preocupar de la gestión del proyecto("project management"), de su presupuesto, de sus fechas de entrega, de su adecuación de patrones standars,etc. (Gibbs 1989:602).

En suma, estos proyectos del SEI han de resolver los cuellos de botella que según la Software Initiative están impidiendo el desarrollo del software: **"The first involves the transition from concept(or trial use) to prototype...and the second involves the transition from prototype to product (or fully supported method). A third bottleneck impedes the transition of products and methods to actual practice"** (Ed. Lieblein. 1989:741). Este proceso puede llevar entre 15 y 25 años.

Este diferente carácter de la investigación podemos apreciar una diferencia clave con la que se realiza en la SCS.

Según la Faculty Research Guide de Computer Science de 1991-92, en la escuela citada existen siete profesores, Barbacci, Fahlman, Garlan, Habermann, Rollins, Shaw y Wing, que declaran que su área de interés es "Software Engineering". Dos de ellos, Barbacci y Shaw tienen posiciones simultáneamente en el SEI y la SCS. Tradicionalmente, los tipos de investigación en la academia y en el laboratorio industrial difieren sustancialmente, hasta el punto de constituir dos comunidades de espaldas una a la otra.

Mary Shaw hablaba de distintos "modes of research" en una reunión organizada por la Association for Computing Machinery en 1989: **"Many speakers observed that culture and tradition have inhibited interaction between researchers and practitioners. Complex systems expertise resides in places where it may be hard to find, in corporations or government labs where product delivery schedules, considerations of proprietary information, and differences in the immediate interests of the participants make it hard to get it."** (1989: 52).

Esta diferencia entre tipos de investigación permite entender mejor la separación entre la cultura del SCS y la del SEI. La primera está influida por patrones académicos , y la segunda en el "expertise" de ingenieros del software

empleados en la industria o en laboratorios federales. Al fin y al cabo, el lenguaje ADA no fue el producto de la academia, sino de un grupo de expertos. Pero esta dicotomía no es tan simple. CMU tiene una cultura que como vimos en el primer capítulo no respondía a los patrones de la academia tradicional, sino que estaban conformados según el patrón de un taller: "My heart is in the work". El Department of Computer Science se caracterizó por realizar una investigación basada en el "pragmatic approach". Con todo, es cierto que durante las primeras décadas, la Computer Science y en especial la Artificial Intelligence, ha estado buscando legitimarse como una "ciencia", y realizando sus experimentos como "concept demonstration".

Ello provocó dos movimientos hacia la ingeniería: primero, el de Feigenbaum en los años 70s, dando lugar a los "expert systems" y la "knowledge engineering" (Stefick, Conway, 1982). Y otro durante la primera mitad de los 80s, que ha dado lugar al Software Engineering. Ambos movimientos influyeron en CMU favoreciendo una mayor atención a una Computer Science con proyección útil para la industria (Robotics Institute, 1979), para el Departamento de Defensa (SEI, 1984), o para el resto del campus (Andrew, 1982). Pero al mismo tiempo, estos pasos han favorecido otras reacciones: la constitución de Cognitive Science, en los 80s; la de Robotics Science, en 1990; la mayor formalización de Computer Science en la área de Theory,...

Mary Shaw proponía para afrontar esta dicotomía que se realizaran cambios en la Academia: "**Providing incentives for collaborative work by legitimizing academic exploration of large software systems, and adapting the university reward system so that collaborative as well as individual efforts are rewarded**". (1989:52).

Pero el esfuerzo del SEI no esáa legitimado en la Academia, y no interesa al "computer scientist" como proyecto intelectual.

El modelo de investigación del software engineering que incluye según M. Shaw: "**building systems, measuring things, improving performance, devising abstract formal models, devising representations and notations, and making improvements on previous systems. ..We must better understand group efforts by bringing in participants not just from computing, but also from management and the behavioral disciplines**".(ib.:52), se considera en Computer Science como instrumento, aunque paradójicamente muchos de sus proyectos (GANDALF, TYPES,...) no conducen más que a construir herramientas para la programación. En realidad, la diferencia entre Computer Science y Software Engineering estriba en dos tipos

de diseño, uno realizado a niveles quizá mas abstractos y otros a niveles más concretos.

3.4.5. TRANSFERENCIA TECNOLOGICA A TRAVES DE LA EDUCACION.

Una actividad fundamental del centro es establecer las pautas para la educación de los software engineers. Según Mario Barbacci: **"La educación es una forma de transferencia tecnológica"**. 21-2-92).

Su "Education Program" incluye el desarrollo de curriculums en software engineering para estudiantes tanto pregraduados como postgraduados, la producción de materiales educativos de apoyo, la financiación de conferencias y seminarios, la entrega de cursos en videotapes y la exploración de tecnologías educativas avanzadas. Ahora según informo Barbacci, **"estamos empezando a estudiar la introducción del software engineering en las escuelas"** (21-2-92).

Esta misión de educación forma parte de una estrategia. Según , Norman Gibbs, responsable en el SEI de esta área, refiriéndose a la estrategia de educación del instituto: **"Its role is to serve as a focal point for developments in software engineering education and as a catalyst that influences computer science and software engineering curricula"**.

Esta labor se realiza en conjunto con el "ADA Software Engineering Education and Training Team" (ASEET). En 1980, el Office of the Secretary of Defense creó el "Ada Joint Program Office" (AJPO) encargado de una política para ADA, ("Ada policy"), así como del fomento de la aceptación, el reconocimiento y la estandarización del lenguaje. Esta oficina estableció a su vez el Ada Software Engineering Education and Training Team (ASEET). Según Edward Lieblein: **"Major objectives of the ASEET Team include the identification of education and training requirements within the DoD, the development (in conjunction with the SEI) of Ada-oriented software engineering course materials, completion of a study to determine whether the certification of Ada-qualified software engineers would be of value, the establishment of DoD Ada Study Centers, and coordination with industry on the needs for and approaches to education and training in Ada."** (1986:738).

En el contrato del SEI con el DoD se especificaba que la educación es una misión del instituto: **"It shall...influence software engineering curricula development throughout the education community"**. El SEI define y administra esta actividad, que la llevan a cargo educadores y expertos que colaboran con el SEI, aunque

éste no puede dar títulos académicos.

Esta labor de educación es central en el instituto. Una prueba de ello es que, Norman E. Gibbs, fue anteriormente el encargado del Software Engineering Education programme del SEI, y en el curso 1990-91 dirigía el Master en Software Engineer de CMU. Anteriormente, había dirigido un esfuerzo nacional patrocinado por la Sloan Foundation para desarrollar un curriculum en computer science adecuado para programas de "liberal arts".

Una actividad central del SEI Education Program segun Norman E. Gibbs, reside en su "graduate curriculum project" que se inició en 1985:

"The goal of the project is to identify and disseminate a body of knowledge that will help current and future software engineers be more productive. This requires documenting the body of knowledge called software engineering as the discipline evolves, and ongoing activity that includes identifying fundamental software engineering principles and the best current software engineering practice and methodology". (1989:597).

Una vez conseguida esta documentación, partes de la misma se seleccionan y se empaquetan adecuándola para diferentes curriculums e instituciones. Cada área se subdivide en unidades denominadas "curriculum modules", cada uno de los cuales tiene un núcleo coherente de conocimientos, según Gibbs. El contenido final del software engineering puede consistir en un total de entre 50 y 100 módulos.

Lo interesante a destacar aquí es que el SEI no sólo realiza una labor educativa. Lo que realiza, ante todo, es una labor de codificación del cuerpo de conocimientos llamado "software engineering". Está construyendo la disciplina. La importancia de esta labor se mide por su impacto. El SEI cuenta con una red nacional e internacional de centros académicos afiliados que están influenciados por esta labor de fijación de estándares educativos. En 1989 contaba con 39 universidades en Estados Unidos, entre ellas la Universidad de California, Irvine, la de Purdue y la Columbia University, y con centros afiliados en España, Escocia, Australia e Inglaterra.

Los módulos antes citados se distribuyen de distinta forma:

- a través de masters en software engineering (MSE),
- de especialidades dentro de los programas de graduación en computer science (MS CS).
- de cursos electivos dentro de los masters de Computer Science.
- de uno o dos semestres para estudiantes de grado ("undergraduates").

- de cursos de proyectos en software engineering.
- de cursos cortos intensivos en la industria.
- de seminarios para managers y políticos ("decision makers").

Una de estas aplicaciones, el MSE es la que la School of Computer Science de CMU comenzó en otoño de 1989.

En suma la filosofía educativa de este instituto la expone N. Gibbs:

"Encouraging and helping others to implement the curriculum is vital to the technology transition mission of the SEI. Our strategy is based on the idea that if many people have a stake in the curriculum, it has a better chance of being accepted and used. Thus, the SEI does not prescribe educational solutions but rather cooperates, coordinates, disseminates, and supports. Many outside educators and experts come to the SEI to learn and to contribute." (1989:598).

Pero el SEI no reduce sus actividades educativas a la formación de postgrado. Ha dado un paso más y pretende dirigirse también a la gran masa de estudiantes de grado, los llamados en Estados Unidos los "undergraduates".

Para Norman Gibbs: **" In terms of numbers of students affected, the SEI Education Program can make its greatest impact by influencing undergraduate computer science curricula. It has established an undergraduate software engineering education project to exert that influence in effective ways. The project has two immediate goals: 1) To increase the awareness of instructors and students about the differences between programming and software engineering and 2) to develop curriculum guidelines and support materials to facilitate teaching software engineering in undergraduate computer science programs."** (1989:602).

3.4.6. EL "BODY OF KNOWLEDGE" DEL "SOFTWARE ENGINEER": "SPECIFICATIONS, DESIGN, IMPLEMENTATION, TESTING".

Como hemos visto, la transferencia tecnológica del SEI considera la educación una pieza clave. Y a su vez, la labor educativa ha de desarrollar un cuerpo de conocimientos, "a body of knowledge", que constituye la profesión de "software engineer".

¿Qué cuerpo de conocimientos sustentan el "software engineering"? ¿Qué diferencias tiene con el computer scientist?.

Este cuerpo de conocimientos es el que forma la base del Master of Software Engineering de CMU, en el que James Tomayko es su director asociado.

Este master consta de dos semestres en el que se imparte dos tipos de cursos :

los dedicados al tema de la programación de sistemas complejos ("programming-in-the-large"), y los dedicados a la gestión de dichos proyectos ("software project management").

Los cursos dedicados a programación son cinco: "Software System Engineering". "Formal Methods in Software Engineering". "Advanced System Design Principles". "Software Creation and Maintenance". y "Software Analysis". Estos recorren las cuatro etapas clásicas de un proyecto de diseño ingeniero, 1."specification",2. "design", 3."implementation" y 4. "testing".

Los cursos dedicados a la gestión son dos: "Software Project Management" y "Product Management". Estos tratan de la planificación del proyecto, su seguimiento y control, la dirección del equipo que lo diseña, su ajuste a los plazos de entrega, al presupuesto, así como aspectos legales y comerciales del mismo.

Por último, este Master incluye como rasgo característico e innovador del mismo un "Software Development Studio", cuyo responsable es James Tomayko, donde los estudiantes, como en Arquitectura, han de desarrollar un proyecto práctico en equipo.

Centremonos en el primer bloque de conocimientos: el que estudia la construcción de proyectos de software, como átomo básico del saber de una cultura tecnológica, un elemento básico de la "computer culture".

Una de las diferencias básicas entre "programming systems" y software engineering es el aspecto de los "requirements specifications".

Según Barry W. Boehm el software engineer ha de producir una documentación escrita que explicita las distintas fases de desarrollo del sistema, en especial los requerimientos del mismo: **"Software requirements engineering is the discipline for developing a complete, consistent, unambiguous specification --which can serve as a basis for common agreement among all parties concerned--describing what the software product will do (but not how it will do it; this is to be done in the design specification)".** (1975:1227).

Esta necesidad de hacer explícitos los requerimientos es propia de la ingeniería tradicional, en la que como profesional el ingeniero ha de partir de atender la demanda de un cliente, normalmente una industria que le solicita sus servicios para solucionar un problema dado. Estos servicios se plantean en forma de "design specifications" o "requirements specifications".

Este procedimiento permite ver que la tecnología no es un mero "know-how". El

"saber-cómo" se necesita después del "saber-qué-producto-se-quiere". Este segundo conocimiento es el que da sentido al primero y sin el cual no se sabe qué diseñar, pues no se sabe el para qué. Estos requerimientos los puede plantear el cliente, o se pueden elaborar conjuntamente entre el cliente y el software engineer, pero son previos al diseño del artefacto, utilizando el término diseño aquí en un sentido reducido.

En este sentido, hay dos tipos de diseño: el diseño de los requerimientos, y el diseño del sistema concreto para resolver esos requerimientos. Y ambos dan por resultado el producto tecnológico. Hasta ahora por diseño se ha entendido la labor sólo de diseño del artefacto. Conforme la técnica se ha complejizado, e instituciones de todo tipo precisan de ella (Defensa, políticos, empresas,...), se advierte que la fase de plantear los requerimientos que ha de cumplir el sistema es clave. Y, sin embargo, es la menos estudiada.

Esta visión ampliada del diseño puede introducir en una institución una cultura de planificación, diseño, e invención de sus propios fines, metas y funciones, en suma, una tecnocultura, una práctica de invención de su propia organización, incluyendo su sistema informático. No es extraño que el management naciera como disciplina de la ingeniería. Y la ingeniería estuviera durante siglos bajo la dirección de los militares. Esta ha pasado tres etapas: ingeniería militar, ingeniería industrial, y ahora empieza a ser ingeniería del conocimiento y del software, nacida de la cultura del ordenador.

Esta necesidad de ver el diseño de una forma más amplia fue ya detectada por Barry W. Boehm, al referirse al software engineering. Este veía preciso **"the necessity of considering a broad enough interpretation of the word 'design' to cover the extremely important activity of software requirements engineering"**.(1975:1226).

Tradicionalmente, la ingeniería ha partido de los requerimientos que le planteaba un cliente. Pero al ser este un cliente poco cualificado e ignorante en el tema, el ingeniero unía las dos funciones en la misma persona, así los requerimientos y las especificaciones del sistema venían a ser lo mismo. Es más, el sistema es el que planteaba al cliente requerimientos que éste ni se había imaginado. Es el modelo llamado "technology push".

Pero en el caso del SEI, al ser ésta una institución nacida como resultado de una iniciativa, comenzada con el lenguaje (ADA), un sistema construido explícitamente para cumplir los requerimientos del Department of Defense, o

STEELMAN requirements. El modelo es el inverso, el "technology pull". Y éste se rige por la demanda de otro tipo de tecnólogos, los técnicos militares, en concreto el HOLWG o High Order Working Group del DoD.

Si consideramos el diseño del lenguaje ADA como un modelo de "software system development " (SSD) veremos la importancia que ha tomado la función del diseño de los requerimientos, esto es, de los objetivos del sistema. El análisis de las diferentes fases de desarrollo de ADA lo encontramos en Grady Booch, "Software Engineering with ADA" (1987). Estas fueron según este autor: **1. "Analysis Phase", 2. "Requirements Definition Phase", 3. "Design Phase". 4. "Testing Phase". y 5. "Operational and Maintenance Phase".** (ib:14-22).

En la comunidad del SEI es muy conocido el modelo de Frederik P. Brooks Jr. (The Mythical Man-Month. 1982) para la construcción de sistemas de software. Se basa en el llamado "surgical team" (el equipo del cirujano) o en la relación "architect" y "builder", el arquitecto y el constructor o maestro de obras. Según este modelo, el "architect" o "surgeon" es el encargado de diseñar el sistema, escribiendo el "programming manual". En el caso de ADA habría sido el diseñador Jean Ichiabh, quien escribió las especificaciones del lenguaje , su diseño.

A continuación, el arquitecto pasaría los planos del sistema a los aparejadores o "builders", encargados de su implementación.

Pero este modelo se olvida que ADA fue diseñado de acuerdo con unos requerimientos previos fijados por el DoD. Y estos requerimientos eran una decisión adoptada para resolver un problema , ésto es, el famoso "gap" entre unos objetivos que tenía el DoD, un software de calidad y económico y una realidad por la cual el precio del software se estaba disparando y su calidad dejaba que desear.

Ello indica que el modelo del "surgical team" sólo es válido para pequeños proyectos. A más complejidad del proyecto, más importancia cobra la figura de la institución que fija los requerimientos, que no es quien diseña el sistema sino la que decide qué sistema ha de ser diseñado. Esto es, la que diseña al diseñador.

En CMU , profesores como Angel Jordan, y otros como Allen Newell o Raj Reddy, han cumplido esa función durante años. Como declaraba a un periódico español:

"Yo no diseño robots, pero sé cuales son los robots que hay que diseñar. Podría diseñar pero hay gente que lo hace mejor. Me dedico a aglutinar a esas personas

y a marcar las pautas". (EL PAIS, 15 DIC. 90).

Frederik P. Brooks, Jr. llama a esta figura el "**producer**" y la aconseja como jefe del proyecto, por encima del "**technical director**" o "**architect**" para los proyectos grandes.

El DoD actuó en el caso del ADA como "producer" o "community engineer", planteando los requerimientos que esa comunidad, sociedad o institución hacía a ese nuevo sistema tecnológico, y teniendo en cuenta el conjunto de la organización donde ese sistema iba a funcionar. Si esta función no está clara, se puede diseñar un sistema que funcione técnicamente de forma correcta, pero inválido para atender los requerimientos del cliente.

Pero eso plantea un problema sobre la propia funcionalidad de la tecnología.

¿Puede existir una tecnología que funcione correctamente al margen de su uso por la comunidad que la ha solicitado?. La naturaleza precedió al surgimiento del sapiens. Pero la tecnología es su producto. No puede preceder a su diseñador. El problema es que el modelo tecnológico que hasta ahora se ha seguido es aquel que derivaba la tecnología de la ciencia, ante todo, la ciencia natural, cuyo objeto de estudio, la naturaleza, sí precedía a su investigador. De ahí que se pensara que la técnica imitaba a la naturaleza, y era anterior al propio ser humano. Era en este sentido natural, "objetiva".

Pero el caso de la computer science apreciamos en su origen hay un propósito político directo, la defensa nacional, no científico natural. El esfuerzo bélico de la II Guerra Mundial no sólo creó la tecnología informática sino que dio origen a un nuevo tipo de disciplina, la "computer science", y ahora al "software engineering". Y este modelo tecnológico es planificado y diseñado antes con fines políticos y luego tecnológicamente.

Finalizada la Guerra Fría, la función motora del Departamento de Defensa en el impulso tecnológico pierde justificación social. La pregunta es, ¿qué institución o combinación de éstas va a plantear los requerimientos del nuevo modelo tecnológico?.

Si nos atenemos a este modelo que ha hecho derivar saltos tecnológicos de opciones políticas, nuevos saltos tecnológico de la humanidad dependen cada vez más del diseño por primera vez de programas culturales complejos y sus sistemas socio-tecnológicos correspondientes.

Para el diseño de estos sistemas socio-tecnológicos complejos ya no es suficiente la sola iniciativa de una institución como el DoD, por muy potente

que sea. Una de los fallos de la iniciativa en solitario de ADA y atendiendo principalmente necesidades militares, es que no consiguió el suficiente apoyo de la academia y la industria . A más participación y consenso en el diseño de requerimientos para los nuevos sistemas tecnológicos más aceptación social tendrán éstos. El actual programa sobre "High Performance Computing and Communications" iniciado en 1992 reúne ya a 10 agencias federales, entre ellas, DARPA, la NSF, la NASA, la DoE y otras.

La "software crisis" analizada es un síntoma de una enfermedad más profunda: la crisis del sistema de conocimiento denominado "Science and Technology". Se puede resolver si se reconoce que la tecnología en el caso de la "computer culture" antecede a la ciencia en innovación, y no queda relegada por ésta a un papel secundario.

Pero esta inversión requiere que la tecnología, y la ingeniería, no limiten sus esferas de actividad al mundo de las máquinas, sino que se extiendan a otras esferas monopolizadas por la ciencia, como los organismos vivos, incluido, al propio ser humano y su mundo cultural. El diseño de los requerimientos de un sistema tecnológico no es tan sólo una tarea de diseño tecnológico sino de tipo cultural más compleja. La experiencia en CMU del proyecto ANDREW lo demuestra.

3.5. EL CASO DEL ANDREW PROJECT Y SU RELACION CON PROGRAMMING SYSTEMS.

3.5.1. ANDREW PROJECT: UN "COMPUTER-INTENSIVE ENVIRONMENT" AL SERVICIO DE LA "LIBERAL EDUCATION".

El Andrew Project puede considerarse un proyecto más de Computer Science y en particular de su área de Programming Systems. Su software fue desarrollado en los años 80s por un equipo de "computer scientists" de CMU, dirigido por Jim Morris, en colaboración con la compañía IBM y un equipo de técnicos suyos. Pero el ANDREW PROJECT fue algo más que un diseño tecnológico. Fue un proyecto basado en un nuevo modelo de universidad.

Andrew fue un el primer proyecto en una universidad norteamericana de construir un "computer-intensive campus". Este concepto nació en CMU en 1980 (Daniel Updegrave. 1986), y culminó toda una era en CMU, la de la llamada "revolucion cognitiva". La iniciativa para su construcción partió del entonces Provost, Richard Van Horn, y tuvo un especial apoyo del College of Humanities and Social Sciences, teniendo a Pat Crecine, su decano entonces, como promotor.

El entonces presidente Richard Cyert en una declaración fechada el 29 de octubre de 1981 se comprometía a la expansión del papel del "computing" en CMU. Para ello formó un denominado "Task Force for The Future of Computing", dirigido por Allen Newell (1982).

Este, junto con dicha comisión representativa del profesorado del campus, redactó un documento básico, donde se exponían las bases de lo que sería el Andrew Project.

Al tratar de analizar este proyecto advertimos que no se formuló como un problema a resolver, al clásico modo del "human problem solving". No se trazó ninguna diferencia entre un "desired state" y un "initial state". Por contra se partía .por un lado, de una visión determinada de un proceso tecnológico "objetivo" que estaba avanzando. Y por otro, se consideraba el computing como un mero instrumento, o "tool" al servicio de un modelo educativo tradicional, fundamentalmente el de la "liberal education".

En su discurso "Charge to the Task Force" el President Cyert, el 29 de Octubre de 1981, partía constatando una realidad dada, un "initial state": **"The importance of computing on our campus continues to increase dramatically. This is occurring pervasively and not just along a single dimension and for a special group...But the causes lie also in our own repeated decisions throughout the last quarter century to involve ourselves with this new element--using it in our research, understanding its nature, building it into the fabric of our scholarly lives"** (Newell,1982:15).

El propósito de formar un Task Force era para **"to attain a broadly-based view and a sense of perspective about where we want to go"**. (Ib:15). Había que desarrollar lo que aquí denominan la "visión", a dónde querían ir. Dado que se partía del proceso inexorable, el camino a seguir estaba determinado por dicho proceso.

3.5.2. "ANDREW": "AN ELECTRONIC NETWORK AS A TOOL".

La tarea de desarrollar la visión , el tipo de futuro que sería más beneficioso ("what kind of future would benefit us most") fue encomendada a un Task Force, liderado por Allen Newell, y con la participación de 16 profesores representativos de CMU: T. Bajzek, B. Berry, L. Cheng, R. Cutkosky, S. Director, Ch. Eastman, J. Hoburg, Ch. Kriebel, J. Larkin, T. Michalak, D.A. Miller, J. Pople, J.R. Scott, L. Sproull, H.D. Wactlar, y L. Zuti.

Tres fueron los supuestos básicos de este informe :1. **"The base-line is a substantially increasing use of computers"**. 2. **" The analysis should focus on designing an environment and predicting impacts"**. y 3 **" Analysis should be in terms of general types of computational facilities"**. (ibid.:7).

Se trataba de aumentar el uso del ordenador por el profesorado y estudiantado del campus, en especial, los de liberal-arts.

Se pensaba que el **"computing will continue to increase substantially, with or without comprehensive planning. Both growth of computation in the external world and the already deep involvement of the university drive CMU along this path"** (1982:7).

Aquí apreciamos dos lógicas distintas de afrontar el problema: la primera, intenta planificar, diseñar un futuro a donde se quiere ir; la otra, simplemente constata que ,se diseñe o no, dicho futuro es inevitable y por lo tanto no hace falta tal plan.

El único requerimiento concreto que se planteó fue la necesidad de construir una red local informática de calidad: " The campus must have a high-quality local-area computer network". Y ésta ha sido la más importante realización de ANDREW. El objetivo de esta red era servir de herramienta, o "tool" al servicio de las tareas educativas, y en especial, de la "liberal education", aquella que no es ingeniería o computer science.

Las redes electrónicas no nacieron con ANDREW. fueron iniciadas en 1969 por el Pentágono con la denominada ARPANET . Hasta entonces el modelo de computing estaba basado en lo que se denominaba "time-sharing": Escasos ordenadores centrales a los que se tenía acceso en tiempo compartido desde terminales . La experiencia de un sistema de ordenadores personales distribuidos y conectados a una red sólo se había iniciado en el laboratorio Xerox Palo Alto Research Center.

La novedad aportada por Andrew fue, como acaba de afirmar William Y. Arms, actual Vice President for Academic Services en CMU, reconocer que este estilo de computing era apropiado para una universidad entera y que la baja de los costes del hardware lo hacían asequible. (Arms, W. 1990: 33).

El 20 de Octubre de 1982 Lewis Branscomb, director de investigación y vicepresidente de IBM y Richard Cyert, presidente de CMU presentaban en una conferencia en el Wherret Room de CMU el acuerdo para desarrollar el "personal computing environment".(CMU/IBM, 1982).

Para Cyert este sistema se iba a convertir en "the prototype for every major university in the country".(ibid:6) El modelo que se tenía de referencia eran las universidades de investigación del país, las "research universities".

Para Van Horn, provost de CMU, para poder preveer como iba a resultar el experimento había que fijarse en lo que estaba ocurriendo en otro computer intensive environment, "the computer science department". (ibid:11): "The best kind of model you can get there is to look at the kind of behavior that goes on now in a computer intensive environment, and that's really the computer science department".

El mismo Van Horn ya preveía cuales iban a ser los usos de los ordenadores por los estudiantes: **"The functions that students will use computers for are really already well defined on this campus. We see a lot of use of electronic mail, students communicate with each other, they communicate with faculty member, ...Wide use of the device is for text processing. And again, we like to stress independent activity...We see access to library catalogues, and probably, eventually, to library materials...Probably the most extensive use will be the development of skills, problem solving skills...We see this wide range of uses with emphasis on independent learning by students..."** (1982: 9).

IBM preveía invertir 20 millones de dólares en dicho proyecto entre 1982 y 1987, construyendo el Information Technology Center en CMU donde 30 computer scientists construirían el sofisticado software de la red.

Pero ANDREW era algo más que un mero experimento técnico. Era la parte técnica de una estrategia más compleja.

3.5.3. "ANDREW" Y LA "EDUCATIONAL REVOLUTION": "SOFTWARE FOR HIGHER EDUCATION".

ANDREW fue una red electrónica pensado con una finalidad educativa expresa. Se trataba de llevar a la práctica, al terreno de la educación, los descubrimientos realizados por la "revolución cognitiva" años antes. Era la parte educativa de dicha revolución.

Se veía como el proyecto tecnológico que iba a revolucionar la educación. **"The computer systems we are developing will be a revolution in education"** (Cyert. Pittsburgh Press. Oct 21.1982). La conclusión general era que CMU **"should become the leader of educational computing in universities"** (ICT 1984).

Advirtamos que cuando se ha hablado de educación tradicionalmente en CMU se

decía "liberal-professional education", no sólo "education". Y ello era porque el modelo de CMU era distinto al mayoritario en Estados Unidos. Cuando en este país se habla de education y más de "higher education" se está hablando de la educación impartida en las grandes universidades nacionales del país, con Harvard a la cabeza. Este cambio, el pasar a hablar en general de "revolution in education" es coherente con la idea de la "knowledge revolution", que entiende por conocimiento lo que la ciencia cognitiva, ante todo, conocimiento científico, basado en dos enfoques o paradigmas, "search" o "reasoning". (Simon & Kaplan, 1988:20).

La campaña de ANDREW trató de poner a CMU a la cabeza de este tipo de instituciones universitarias, con un modelo, no obstante, específico. Se trataba de hacer una revolución educativa en la que la liberal education utilizara como herramienta los ordenadores para producir software educativo, ésto es, software de liberal education, principalmente.

Para Herb Simon: **"The task now is to further develop that understanding and to apply it to creating software that we on this campus and other campuses are going to want to use. That is a valuable goal"**. (1987:14).

Un principal impulsor de esta orientación fue Pat Crecine, Vice President for Academic Affairs entonces de CMU: **"The major point here is that technological and institutional preconditions for an electronic information revolution in higher education can be well in place by the start of the next decade...With a large, installed base of compatible workstations and machine-independent systems software or operating environments, institutional preconditions will be in place to force incentives for producing software for higher education to a zenith"**.(1985. EDUCOM:3).

"Higher education" en Estados Unidos significa 156 "universities", 1953 "four year colleges", 1.378 "two-year colleges" y "technical schools" con un total de 793.000 "faculty", 14 millones de estudiantes y un volumen de negocio de 100.000 millones de dolares. (Time, April 13 1992). CMU estaba catalogada en 1990 entre las 25 primeras universidades del país. Ese grupo de universidades de élite, liderado por Harvard, era el modelo de referencia de ANDREW.

Jean Jacques Servan-Schreiber, político, periodista y ahora profesor en CMU llamaba al proyecto: **"La revolución del conocimiento"** (1986). En su opinión, la universidad estaba pasando a convertirse en "fabricante de conocimiento", en forma de software. (1986)

Perdida la batalla con Japón en el terreno industrial, en su opinión la única vía que le quedaba a Estados Unidos era poner a las universidades a producir software, unos de los campos en que Estados Unidos supera aún a Japón. Las universidades computerizadas serían la clave de esta estrategia, según Servan-Schreiber.

En ese mismo libro, este autor elogiaba al Software Engineering Institute poniendo a este instituto como una pieza más de CMU.

Sin embargo, Servan-Schreiber no advertía que el SEI había surgido de la deficiencia de la propia universidad para resolver un problema central : la "software crisis". Y el modelo educativo del SEI era distinto al que se pretendía para CMU. Esta nueva institución debía formar "software engineers", capaces de diseñar sistemas informáticos complejos. ANDREW se pensó como herramienta ante todo para los estudiantes de artes, ciencias y humanidades, a fin de que pudieran transformar sus conocimientos con ayuda del ordenador.

El software que Pat Crecine proponía construir en ANDREW no tenía ninguna finalidad industrial, era software for "higher education": se pensaba que los profesores de historia, de cálculo, de física, transformarían sus clases en software educativo que sustituyera a los libros de texto. Era poner la informática como una herramienta al servicio de las disciplinas tradicionales de una universidad : **"The computer is a tool--a means, not an end--to be used by the instructor when judged valuable. Like a book, it is powerful and has particular usefulness in the instructional setting"** (Newell,1982:9).

Este había sido el modelo seguido desde los años 60s por Herbert Simon: utilizar el ordenador al servicio de la "liberal education". ANDREW fue su culminación.

Esta red se pensó, en su aspecto educativo, principalmente pensando en las "Humanities and Social Sciences", y para las Ciencias en general. Jill Larkin, psicóloga y física, fue la primera directora del Center for Design of Educational Computing y Preston Covey, profesor de filosofía, el segundo.

El Information Technology Center (ITC), por su parte, dirigido por James Morris cumplió con su misión. Creó un sistema distribuido para correo electrónico y para boletines electrónicos, y por otra parte creó la red y el sistema de archivos (Andrew File System).

Por su parte el profesorado se ha esforzado, durante estos años, en construir el software previsto. El CDEC , Center for Design of Educational Computing, dirigido por Jill Larking y ahora por Preston Covey, ha dinamizado este

esfuerzo.

Pero la "revolution in education" no se ha producido, al menos como se esperaba en un principio. Como afirmaba en 1990 John Howard, director del ITC en esas fechas: "**Andrew clearly has not lived up to those expectations. It clearly has not revolutionized education. On the other hand, we have done some very worthwhile and interesting things**". (CM Magazine, Fall 1990:40).

¿Por qué no se ha producido ninguna revolución en la educación?

Si una revolución es un cambio innovador, ¿qué ha cambiado y qué no con ANDREW?

3.5.4. LOS DISEÑADORES DE "ANDREW": EL "INFORMATION TECHNOLOGY CENTER" DE CMU-IBM.

Nunca antes se había construido una red informática distribuida abarcando todo un campus universitario.

Técnicamente, James Morris, profesor de la School of Computer Science, dirigió el equipo constructor de dicho software, descrito en el artículo "**Andrew: A distributed personal computing environment**". (Morris, et al.1986).

La tarea de diseñar la red Andrew recayó en el Information Technology Center, centro construido en CMU con el dinero de IBM. La misión del ITC fue diseñar y desarrollar "the central part of the Andrew system, the network file system, and workstation software" (IBM,"CMU" : 5). La génesis del proyecto se incubó en el Computer Science Department y la construcción de dicho software fue encomendada a un grupo de 30 personas, 20 empleados de CMU y 10 de IBM.

Según el artículo citado de J. Morris: "**The ITC development effort has focused on three of these components: network communications, the shared file system, and the user interface on an individual workstation**" (1986:187). Esto es, en la red de comunicaciones, en el sistema de archivos compartidos y en la interface con el usuario en cada estación de trabajo.

Respecto a la red de comunicaciones los dos problemas que resolvieron fue el cableado con fibra óptica de todo el campus, conseguido con una "token ring network" desarrollada por IBM, y la conexión de las diferentes redes preexistentes en campus (Ethernet y ProNets) gracias a las máquinas "router" PDP/11s programadas por el Computer Science Department y a la extensión de los protocolos TCP/IP desarrollados por Arpanet.

Andrew es una extensión de la red Arpanet, la red patrocinada por el Departamento de Defensa y a la que el Departamento de Computer Science está

integrado.

El trabajo más innovador del proyecto Andrew en lo que respecta a la parte técnica fue el diseño del sistema de archivos o Andrew File System. (AFS), la piedra central del sistema de comunicaciones de Andrew.

En un reciente balance de la experiencia de Andrew William Y.Arms afirmaba: "**The single most important lesson from Andrew is the value of the network. The facilities and services provided by and through the network are the foundation of the future of computing at Carnegie Mellon**" (Cursor. March 1990).

¿Qué se ha conseguido?. Esencialmente los objetivos marcados en el "Task Force for the Future of Computing". La red técnicamente funciona y funciona muy bien. El Andrew message system soporta 3850 boletines electrónicos con fecha febrero de 1992. El Andrew file system es usado por 5000 usuarios cada mes y el Library Information System provee de información de bases de datos bibliográficos a todos los usuarios de la red.

Se ha conseguido lo que se denominaba una "computer literacy" en la mayoría de profesores y alumnos. Una encuesta de 1989 demostró que el 90% del faculty usa el ordenador para preparar documentos, el 68% para mandar correo electrónico, y el 76% para consultar a la librería. Y también se ha conseguido avanzar en el segundo objetivo: la producción de software educativo, aunque en menor medida.

3.5.5. EL SOFTWARE DE LOS PROFESORES: "THE NEW LIBERAL ARTS".

El Center for Design of Educational Computing (CDEC) ha sido el centro encargado de ayudar a los profesores en la producción de software educativo. Como lo define Preston Covey su actual director: "**CDEC's mission is to provide leadership and support for the design and development of advanced computer applications for the liberal arts. CDEC aims to generate strategic planning, design wisdom, dedicated academic and technical expertise focused on the problems and promise of advancing technology for the liberal arts**". (1988:7)

Por "liberal arts" se entiende lo que no es ingeniería o management: "sciences, mathematics, fine arts, humanities and social sciences", en suma, las dos grandes áreas de la educación clásica, las artes y las ciencias. "**CDEC addresses two broad areas of education, the arts and the sciences, and two broad strata of education, secondary and postsecondary**" (Covey, 1991:2).

Preston Covey llegó a CMU en 1974 para enseñar un programa en Filosofía dentro del departamento de Historia. Poco más tarde, este programa sería la base del

nuevo departamento de filosofía. La misión de este programa era enseñar técnicas, "skills": "skill of language, skill of rethoric, skill of logic" que pudieran ser útiles a los estudiantes de ciencias y de letras (Covey, Schwartz, Focus1974:). Lo nuevo de este programa, a diferencia de lo que se denominaban "old liberal arts concept", es que estaba planteado sobre casos prácticos (aborto, eutanasia) lo que denominaban "moral dilemas".

Como afirmaba Tom Schwartz, cofundador con Preston Covey del programa citado: **"Particularly in the area of ethics, it's the professional who's concerned with decisions and actions, rather than simply books and ideas, who's most acutely aware of and puzzled about normative problems....And that's one of the reasons most of us are here.."**(Focus.74).

Preston Covey ha seguido interesado en esta orientación. Acaba de organizar en 1990 el Center for the Advancement of Applied Ethics. Esta orientación es la que se dirige a educar profesionales con ayuda de instrumentos que puede proporcionar la ética aplicada.

Pero como director del CDEC su orientación es la distinta, por no decir la inversa. La orientación del centro es utilizar el computing para formar "educated people", o personas educadas en la "liberal education".

Su investigación se apoya en la investigación básica en psicología, pedagogía y computer science. Y la misión de su centro es utilizar la tecnología informática para hacer avanzar la ciencia y las humanidades.

Gracias a una donación de la Sloan Foundation, se desarrollaron en CMU, entre 1984 y 1988, 40 proyectos educativos (que contenían el diseño de curriculum y software educativo) y alrededor de 50 aplicaciones de software. El CDEC apoyó durante esa época el 75% de las aplicaciones informáticas educativas de CMU. Dos tipos de aplicaciones destacaron: las dedicadas a las ciencias, en especial la física, y las dedicadas a las artes y humanidades, con especial interés en la ética.

Jill Larkin, profesora de física y psicología, tuvo un papel promotor en la creación de dicho centro en 1984. Su interés estaba centrado en desarrollar sistemas informáticos que ayudaran a mejorar el aprendizaje de la física y las ciencias naturales. Su investigación sobre el aprendizaje está orientada a este fin. Forman parte de este grupo Frederick Reif y Bruce Sherwood, autor del ambiente de programación denominado "cT", utilizado para fines educativos tanto en CMU como en otras universidades dentro y fuera del país.

El segundo grupo del CDEC, dirigido por Preston Covey y Robert Cavalier, tiene como finalidad principal el desarrollo de aplicaciones que mejoren la enseñanza de materias del grupo de las humanidades y las artes.

En 1988, Preston Covey recibió un premio nacional por el videodisco interactivo, "A Right to Die?. The Case of Dax Cowart", donde basándose en un caso real presentaba un dilema ético a decidir por el estudiante, acerca de la eutanasia o no de un accidentado

Otros programas educativos informáticos de amplia aceptación en CMU son The Great American History Machine (1987), de David Miller y Stephan Greene, un software para aprender historia. El programa Sketch (1987), de Jill Larkin, Carol Scheftic y David Trowbridge, un tutor inteligente para realizar gráficamente ecuaciones de álgebra, o el programa General Chemistry, de Ruth Chabay, premio al mejor programa de química en 1987

Sin restar valor a estos esfuerzos, lo que apreciamos en ellos es que se intenta vehiculizar contenidos de la "liberal education" tradicional con nuevas formas extraídas de la cultura del ordenador. Así se dan cursos de historia o ciencias naturales utilizando lenguajes de programación.

Pero el software educativo no cambia, en lo sustancial, el modelo de educación, que sigue siendo "liberal education", ahora con ordenadores.

Pero lo que sí cambia es el significado del software al tener que adaptarse a la "liberal education". Concibido en este marco, y fuera de la cultura de Computer Science, el software no tiene como función el conseguir que una máquina realice unas determinadas instrucciones. Su objetivo no es tecnológico. El software educativo es instrumental, una vía para aprender algo que no es informática. Se compara con el libro. Lo educativo no es el proceso de diseñar ese software sino una vez diseñado, no importa como sea, el utilizarlo para aprender lo que antes se aprendía mediante libros. Es un software que podríamos llamar representacional, pues su objetivo es representar un conocimiento dado. Por eso utiliza sistemáticamente la simulación como principio básico. Es "como si" fuera la realidad. Es una metáfora, un símil, de la realidad, a la que trata de conocer.

Por contra, para el computer scientist y sobre todo el software engineer, el software es un fin en sí mismo, es la realidad principal, una realidad artificial nueva que funciona, que opera, que transforma el mundo real.

Podríamos llamar a este software operativo, real.

Si comparamos el software educativo de "liberal arts" con el que producen los software engineers del SEI veremos que este último es un fin en sí mismo, mientras que el de liberal arts es un medio para realizar sus funciones tradicionales artísticas, humanísticas o científicas.

Confundir los dos tipos de software puede llevarnos a equívocos graves. Una gran producción de software de liberal arts education es semejante a un gran biblioteca, puede contener mucho saber acumulado para el mero análisis de la realidad. Puede ser una nueva biblioteca de Alejandría informática. El siguiente proyecto a ANDREW ha sido el proyecto MERCURY, dirigido por William Arms, Vice President for Academic Services, un primer paso hacia la biblioteca informatizada universal. (CMU-Mercury, 1992, Kibbey & Evans, 1989). Es su opinión, el ordenador es sólo una herramienta del profesor: **"My idea is to consider the computer as a tool to the teacher, to support the education. I'm proud of the first electronic library in an american university. My project now is to connect personal libraries is the network"** (Entrevista, 11-2-92). Esta idea es interesante. Apunta hacia la biblioteca universal electronica, al modo de la biblioteca de Alejandría. Pero al mismo tiempo, borra las distinciones entre "liberal y professional education" propias de la cultura de CMU. La biblioteca de Alejandría fue la expresión final de la cultura clásica sobre la que se levanta el modelo de la "liberal education".

3.5.6. EL MODELO EDUCATIVO DE ANDREW: LA "NEW LIBERAL ARTS EDUCATION".

Carnegie Institute of Technology se fundó sobre un modelo denominado primero "technical education", luego "professional education", más tarde "liberal-professional education". Finalmente, ANDREW en los 80s fue pensado principalmente como una herramienta técnica al servicio de la "liberal education", de una "new liberal arts". Este cambio progresivo del modelo educativo fue acompañado por el intento de construir un "common core" curriculum, o un curriculum basado en un conjunto de asignaturas unificadas para toda la Universidad.

Este cambio se empezó a gestar a finales de los años 70s.

El 9 de marzo de 1977, Herbert Simon daba un discurso en CMU a una amplia audiencia de profesores y estudiantes titulado "Liberal Education in a Technological Society". (Recordemos el discurso de Elliot D. Smith en febrero de 1946: "Professional Education in a Free Society". Este discurso nueve años más

tarde se transformaba en otro distinto: "The Role of Humanistic-Social Education in Making Professional Education Liberal at Carnegie Institute of Technology".)

De esta segunda línea partía Herbert Simon en su discurso.

Partiendo de que "liberal education" significaba "education for a free man, a free person", el propósito de la misma era **"how to bring scientific knowledge about human behavior to bear on the design of a system of liberal education"**.(Focus, 1977:)

Se partía de la base que las universidades americanas, incluyendo Carnegie Mellon, no estaban educadas lo bastante liberalmente. Que el Carnegie Plan había apuntado en ese sentido al reforzar la educación "professional" con la "liberal": **" The idea behind the Carnegie Plan is that sound professional education is necessarily liberal education"** (ibid: 1977). **" If you want to capture part of the informal life of the campus for liberal education, you must create a larger common core as part of its social system"**. Continuaba afirmando que la educación que impartían las universidades americanas, incluida CMU, era provinciana y estrecha ("narrow"). Y por último, acababa proponiendo un conjunto de medidas para mejorar la liberal education en CMU. Una de estas medidas era el establecimiento de un **"program of liberal education for the faculty as a basis for the program that we'll subsequently offer to our students and that it require every CMU faculty member to pass the comprehensive examinations on the common core of that program within the next four years"**. (1977).

La creación de este "common core" en base a la "liberal education" fue una estrategia que se perseguirá durante los años subsiguientes paralelamente a la creación de ANDREW.

Este "core" se empezó estableciendo en el College of Humanities and Social Sciences en 1977 por Pat Crecine. Apoyándose en una tendencia iniciada por el Carnegie Plan, proponía, sin embargo, un cambio inverso en la definición de la "liberal-professional education" de CMU. Mientras que el Carnegie Plan de los años 30s introdujo una formación "liberal" al servicio de formar mejor profesionales, el "core curriculum" de Crecine estaba pensado para proporcionar técnicas o "skills", finalmente ordenadores, a la educación "liberal": **"While there are alternate strategies for achieving both liberal and professional educations, there are substantial overlaps, too. A sensible and defensible strategy, then, for achieving traditional, liberal educational objectives is to use professional objectives and problems to define the appropriate set of**

intellectual skills to be developed in a person's education: to use "professional" problems, broadly defined to answer the question of "what are the fundamentals?". (cit. Achenbaum, 1986, :17).

Esta "defensible strategy" significaba la inversión definitiva de la relación entre los dos tipos de educación coexistentes en CMU. Hasta entonces los ingenieros habían utilizado las humanidades y las artes como complemento de su educación profesional. Lo que se proponía era lo contrario, utilizar problemas ingenieros al servicio de la tradicional educación de liberal arts.

El core curriculum quedó establecido en cinco áreas: "Fundamental Methods & Skills: Writing, Computing, Reasoning", "Humanistic & Social Values", "Social, Political & Economic Systems", "Science and Technology" y "Language, Culture & the Arts".

En esta batalla del "common core" coincidió con un esfuerzo algo anterior iniciado por Harvard. Según, Preston Covey, para algunos se trataba de superar a Harvard en el diseño de ese core. (Entrevista personal, 27 nov. 90).

En 1975, Henry Rosovsky, decano de Arts and Sciences de Harvard University, inició una búsqueda de un conjunto de cursos básicos para los "undergraduates" que sirvieran de "core knowledge". Rosovsky había sido profesor en Berkeley durante los años de los "student troubles" y según Smith (1986:311) veía preciso continuar el esfuerzo iniciado hacia 30 años por James Conant, presidente de Harvard, en pos de un núcleo de cultura nacional coherente, base de una "general education".

Finalmente, en 1978 este core curriculum quedaba aprobado sintetizándose en cinco áreas: "Social Analysis and Moral Reasoning", "Science and Mathematics", "History", "Literature and the Arts", y "Foreign Languages and Cultures". (ibid. 316).

La orientación del entonces presidente de Harvard, Derek Bok, era clara. En 1979 afirmaba: **"Universities are designed to achieve particular purposes. Their special mission is the discovery and transmission of knowledge...Unlike other organizations, such as political parties, environment associations, or civil rights groups, their institutional goal is not to reform society in specific ways. Universities have neither the mandate nor the competence to administer foreign policy, set our social and economic priorities, enforce standards of conduct in the society, or carry out other social functions apart from learning and discovery".** (cit. por Smith,1985. "The Harvard Century":303).

La orientación del College of Humanities and Social Sciences de CMU era algo distinta. Lo que trataba es de extender el tipo de modelo utilizado por departamentos como el de Psicología, desde hacía años: basar la investigación científica tradicional, basada en el descubrimiento en métodos tecnológicamente más avanzados como el ordenador. Utilizar el ordenador como herramienta para la labor científica tradicional.

Este esfuerzo en pos del citado "core" no se quedó en Humanities and Social Sciences, sino que se intentó extender al conjunto de CMU. Y eso se vio reflejado en la orientación del ANDREW Project y en la batalla por un **"university-wide core curriculum"**, llevada simultáneamente. Pero al extenderse al conjunto de la universidad, sufrió interesantes modificaciones, poniendo el tema del **"design"** en su centro, lo que no había ocurrido en el College of Humanities.

3.5.7. "ANDREW" Y EL "UNIVERSITY -WIDE CORE CURRICULUM": EL "DESIGN" COMO "CORE KNOWLEDGE".

El 26 de abril de 1983 se publicó en todo el país un informe titulado "Nation at Risk" elaborado por la National Commission on Excellence in Education que tuvo el mérito de convertir el tema de la educación en un problema nacional. En este influyente informe se diagnosticaba que Estados Unidos tenía unos niveles educativos inferiores a sus competidores japoneses y alemanes, ofreciendo como modelo construir "The Learning Society": **"At the heart of this society is the commitment to a set of values and to a system of education...Such a society has as a basic foundation the idea that education is important not only because of what it contributes to one's career but also because of the value it adds to the general quality of one's life"** (1983:471).

Respecto al curriculum, las recomendaciones se dirigían a poner fin al "cafeteria-style curriculum", o estilo por el cual cada estudiante elegía el plato-curso preferido, y proponían que todos los institutos de enseñanza media públicos requirieran cinco cursos básicos, **"Five New Basics": "a) 4 years of English; b)3 years of mathematics; c)3 years of science; d) years of social studies, and e) one-half year of computer science."** (ib:475).

Este curriculum se planteaba como el adecuado para la "Information Age" en la que el país estaba entrando.

Esta necesidad de un "core curriculum" basado en la "liberal education" es la

misma que se recogió en CMU a principio de los 80s.

Una vez establecido el core curriculum en el College of Humanities and Social Sciences, el siguiente paso fue extenderlo al conjunto de la universidad. En 1980, el presidente Cyert estableció un comité para la reforma del curriculum, dirigido por Crecine, que en 1983 fue nombrado senior vice-president for academic affairs, encargado de la política educativa del campus, incluida la dirección del proyecto ANDREW.

En la primavera de 1984, representantes del College of Fine Arts, el Carnegie Institute of Technology, el College of Humanities and Social Sciences y el Mellon College of Science se reunían para poner en marcha un nuevo conjunto de requerimientos para la educación general. (Achenbaum, W.A.). El resto de escuelas no asistían al ser sólo de estudiantes graduados.

Para Achenbaum, se trataba de encontrar "a common intellectual skills ant the heart of both" de la "liberal" y de la "professional". (ibid:). Achenbaum, entonces Associate Vice President for Academic Affairs, trazaba una hipótesis interesante: **"is it not possible that the concept of design is an appropriate basis for beginning to think about some interconnections among the liberal arts, science and technology?...If this hypothesis is tenable, then CMU is an ideal environment for demonstrating its salience. "** (Focus, 1985).

Este "core knowledge" cambiaba algo su sentido original, respecto a la idea planteada por Simon en 1977. Se basaba en idea de "design", tratándola de extenderla de la ingeniería a otras profesiones y áreas, tal y como había planteado también Simon con anterioridad: **" Engineers are not the only professional designers. Everyone designs who devises courses of action aimed at changing existing situations into preferred ones. The intellectual activity that produces material artifacts is no different fundamentally from the one that prescribes remedies for a sick patient or the one that devises a new sales plan for a company or a social welfare policy for the state. Design, so construed, is the core of professional training"** (cit. Achenbaum, 1985).

Pero el problema que no detecta Achenbaum es que Simon está hablando aquí de "professional training" no de "liberal/profesional education". Y la "liberal education" no ha tenido el diseño como su centro, sino el descubrimiento, base de la ciencia.

La batalla de fondo del curriculum planteada por Simon en 1977, era la inversa, la necesidad de que la educación "liberal" pusiera por primera vez en CMU a la

"professional" bajo su dirección. No que esta extendiera su "core knowledge", el "design" a la ciencia y las humanidades. La hipótesis de Achenbaum no se pudo llevar a término.

Por otra parte, el diseño se consideraba de distinta manera. Para H.Simon era una ciencia, acorde con su modelo de "liberal education". Para otros, era un arte, lo cual era coherente también con el modelo "liberal-arts". Así Preston Covey, profesor de filosofía, sostenía que **"when the artifact in question, the object of design, is a social institution or culture, then the model of design as an art, rather than the model of design science, seems more apt"**.

(Notes:1991).

En cualquier caso, el conflicto central no era entre ciencia o arte, sino entre ciencia y arte de un lado ("liberal-arts education") e ingeniería de otro (professional education). Este conflicto se unió a otro: "education" versus "research", relacionado con el anterior y que se reflejó en una división en la cúpula de CMU.

En 1983, el presidente Cyert estableció una "two-provost structure" en Carnegie Mellon. Por un lado, Cyert nombró a Pat Crecine, "Senior Vice-President for Academic Affairs". Su tarea era extender la experiencia del "core" de "liberal education" al conjunto de la universidad, y establecer un "computing environment" como herramienta del mismo, es decir, ANDREW.

Al mismo tiempo, Cyert nombró a Angel Jordan, Provost de CMU, encargándole la dirección de las áreas profesionales y de investigación: la Graduate School of Industrial Administration, la School of Urban and Public Affairs, el Carnegie Institute of Technology y el Mellon College of Science, donde se ubicaba por entonces aún el Department of Computer Science. De este cargo dependía la responsabilidad de la "research policy" de la universidad.

El mayor éxito de Angel Jordan, como Provost fue traer, junto con otros profesores, el Software Engineering Institute a Carnegie Mellon en 1984. Este logro reforzó la área de Computer Science y le permitió autonomía suficiente para conseguir en 1985 su independencia del Mellon College of Science. Por su parte, la dirección de la "education" quedaba bajo la supervisión de Pat Crecine, del otro provost. Pero "education" sin las áreas que dirigía Angel Jordan sólo podía entenderse como "liberal education". La hipótesis de Achenbaum era inviable. La "liberal education" no estaba interesada en el "design", y la "research" y las áreas profesionales no les interesaba demasiado la "education".

Pat Crecine dejó CMU a finales de los 80s. Cyert abandono el cargo en 1990. Y en 1991, se eligiría en CMU un nuevo presidente, el Dr. Mehrabian, con el apoyo de la antigua Carnegie Institute of Technology. Pero la reforma de la educación con los principios del diseño ("design") ha quedado pendiente.

Este proceso permite comprender los avances y limitaciones de ANDREW .

ANDREW era varias cosas, un intento de IBM por introducirse en las Universidades, un proyecto de Computer Science por dirigir el campus desde la tecnología, pero sobre todo era el instrumento tecnológico de un cambio cultural más amplio que se planteó en CMU desde finales de los 70s: introducir un modelo de "liberal education" como core curriculum que consideraba la "professional education" como su complemento, revertiendo así toda la tradición de esta institución desde el Carnegie Plan.

ANDREW fue el proyecto técnico al servicio de esta estrategia. Pero este cambio no podía ser una "revolucion en la educación", pues en realidad significaba una vuelta al modelo educativo más tradicional, la "liberal education", y un desplazamiento del más reciente, la "technical education", base de la "professional education". La gran paradoja es que lo que aparecía ante toda la prensa como la "informatización de una universidad", un paso en la transformación de una universidad en una "industria de software", en realidad, era lo contrario, la conversión de un instituto tecnológico, en su origen, en una universidad tradicional, con un rasgo único, que era la más informatizada del país, y que sus estudiantes y profesores utilizaban en ella el ordenador como herramienta al servicio de los fines principales de toda universidad: "learning and discovery".

Esta batalla cogió a la comunidad de "computer scientists" de CMU buscando institucionalizarse como escuela independiente del College of Science. Pero al mismo tiempo, esta iniciativa estaba impulsada por una nueva "profession" que era más joven que el propio "computer science" el software engineering, y que obedecía a problemas candentes del país. Cuando se organiza el Computer Science como escuela independiente, después de muchos debates en CMU, esta decisión era ya demasiado tardía. Un año después, el informe Made in America del MIT lo que advertía es que Estados Unidos tenía que preocuparse de revitalizar su ingeniería hasta entonces demasiado absorbida por la ciencia. Lo que se precisaba era un impulso a las ingenierías nacidas del computer, más que a su consideración como una ciencia.

La etapa del Computer como Science en CMU se apoya crecientemente para avanzar en desarrollos que no son "science" sino cada vez más "engineering": el Software Engineering Institute de un lado, la influencia del Knowledge Engineering en AI y una creciente influencia de la área de Computer Systems en CMU, los diseñadores de nuevos ordenadores en paralelo, área dependiente de avances en la ingeniería electrónica como los VLSI.

4. "COMPUTER SYSTEMS" EN LA SCS.

4.1. QUIEN FORMA "COMPUTER SYSTEMS" EN LA SCS.

4.1.1. EL SECTOR MAS PEQUEÑO DE LA "COMPUTER SCIENCE COMMUNITY".

Hasta ahora hemos analizado la School of Computer Science sin estudiar a los que inventan los ordenadores en tanto que máquinas, que eso son, al fin y al cabo, los "computers", máquinas. Hasta aquí, aún no hemos advertido que tanto la Artificial Intelligence como Programming Systems, son subconjuntos de un sistema informático, de una máquina, que se la denomina ordenador.

Toda la institución se basa al fin y al cabo en el ordenador y su estudio. Y como afirmaba A. Perlis: **"Computer Science studies the phenomena that arise around, and because of, the computer. Without the one, the other would not exist.** ("The Computer Age, 1979:422).

Sin embargo, "Computer Systems" ha sido hasta ahora y sigue siendo la área más pequeña de la School. Inicialmente, en la propuesta de 1964, como veremos a continuación, no se la consideraba si quiera como área de investigación.

"Information Processing" nacía en CMU como un programa de investigación que prescindía de estudiar el ordenador como máquina.

Como ya indicamos en el inicio del Computer Science Department hay dos corrientes con dos proyectos de investigación distintos: "Information Processing" y "Computer Science". Para Simon y Newell, "Computer Science" era una parte de una disciplina más amplia denominada "Complex Information Processing", que incluía también y sobre todo, psicología cognitiva y A.I., disciplinas que dirigían el conjunto del programa. Para la segunda versión, dirigida por Perlis, Computer Science era la nueva disciplina que incluía en su interior a la A.I.. Ambas visiones, con todo, estaban de acuerdo en un punto: no hacer investigación en "hardware", en "Computer Systems", una área más propia de

la ingeniería como veremos a continuación, mejor dicho , de una nueva ingeniería denominada Computer Engineering.

Toda la historia del departamento, no obstante, ha sido ir reconociendo progresivamente el papel creciente de la ingeniería. Los pasos han sido progresivos. Primero reconocer que la Artificial Intelligence debe integrarse "in the world of classics informatics" (Carbonell, 1990) y segundo, que este mundo ha de basarse en la construcción de ordenadores o "computer systems" como iWARP o Nectar.

Ya en los 70s, Raj Reddy, en colaboración con A. Jordan impulsaron el Robotics Institute para encontrar aplicaciones a la Inteligencia Artificial. En los 80s, Nico Habermann, A. Jordan, D. Adams, H.Wactlar y otros profesores impulsaron el "software engineering", a fin de afrontar la llamada "crisis del software" . Y en los 90s, un área que muestra gran dinamismo es la de computer systems, con proyectos tecnológicos como NECTAR de H.T. Kung.

Este proceso ha sido forzado desde las agencias que financian la investigación, como DARPA, dada la progresiva pérdida de liderazgo de la industria norteamericana frente a la japonesa y la alemana. Ello ha provocado en 1989 lo que podríamos llamar una segunda "rebelión de los ingenieros" , el informe Made in America del MIT señalando la necesidad de volver al "manufacturing", y a la ingeniería, si Estados Unidos no quería perder definitivamente su lugar de potencia económica. La "information society" norteamericana está en crisis frente a los planes de convertir a Japón en un "technological country" (Karatsu, 1986: 111).

El análisis de esta área nos puede ayudar a entender la crisis del modelo actual de investigación en CMU, y más allá de esta universidad, del modelo de R&D seguido mayoritariamente en las universidades del país.

La área dedicada a diseñar ordenadores se denomina "Computer Systems" y cuenta en la School of Computer Science sólo con 17 profesores.

Los profesores que el curso 1991-92 forman parte de ella eran : Bianchini, Blelloch, Bryant, Cooper, Director, Fisher, Gibson, Gross, Kung, Maxion, Melzilcioglu, O'Hallaron, Rutenbar, Siewiorek, Steenkiste, Subhlok, y Tokuda. (CMU-CS-FRG, 1991-92)

Siete de ellos, Bianchini, Bryant, Fisher, Kung, Rutenbar, Siewiorek, Y Director, tienen doble adscripción a Computer Science y al Department of Electrical and Computer Engineering del Carnegie Institute of Technology,

incluido Stephen W. Director, actual decano de CIT. Esta relación estrecha con la ingeniería es significativa. Ese departamento del Carnegie Institute of Technology (CIT) tiene 47 profesores, siendo el mayor de esa escuela, seguido de cerca por el de Engineering and Public Policy, con 42 profesores.

Este departamento, catalogado entre los 5 primeros del país, tiene dos programas para estudiantes de licenciatura, uno en "Electrical Engineering" y otro en "Computer Engineering". Además cuenta con varios centros de investigación de excelencia, tres de ellos reconocidos internacionalmente: el "National Science Foundation's Engineering Research Center for Data Storage System", para el desarrollo de discos magnéticos de alta densidad, el "SRC-CMU Research Center for Computer-Aided Design", un centro federal que está desarrollando una metodología de diseño mediante ordenador de circuitos integrados a muy gran escala, y el "Center for Excellence in Optical Data Processing" con aplicaciones en áreas como inspección de productos, automatización, robótica, inteligencia artificial y otras. Cuenta, a su vez, con el "Pennsylvania SEMATECH Center of Excellence", que está desarrollando métodos algorítmicos y software para el control de líneas de manufacturas de circuitos integrados, con el "Center for Dependable Systems" y el "Laboratory for Automated Systems and Information Processing".

Volviendo a la área de Computer Systems es la que en Computer Science se encarga de diseñar y construir los ordenadores como sistemas, no sólo el hardware sino también el software, e incluso las aplicaciones, en suma, el sistema informático o "computer system".

Esta área es donde han nacido los ordenadores iWARP y NECTAR, de los que se enorgullece la escuela.

4.1.2. LOS PROYECTOS DE INVESTIGACION EN "COMPUTER SYSTEMS".

Los principales proyectos de investigación presentados a DARPA de esta área se dividen en dos : "BASIC" y "STRATEGIC".

El proyecto en investigación básica se denomina "VLSI Logic Validation and Parallel Computation"(1990-93) (CMU-SCS-Basic Research, 1989), y está dirigido por Randal E. Bryant, "associate professor" en Computer Science and Electrical and Computer Engineering.

A su vez, el equipo está compuesto por otros 4 profesores: Guy Blelloch, Stephen D. Brookes, Edmund M. Clarke y Allan L. Fisher.

Este proyecto tiene dos partes, la primera se centra en el perfeccionamiento del simulador de circuitos integrados denominado COSMOS. La segunda, en los lenguajes de programación para ordenadores en paralelo.

El segundo proyecto es la área llamada "STRATEGIC", y se denomina: "Research on Parallel Computing: Network-base Multicomputers". (1990-93). (CMU-SCS- Research on Parallel Computing, 1989) Este es el proyecto central de ordenadores en paralelo de CMU. Tiene como objetivo el desarrollo de un sistema completo denominado NECTAR, consistente en una red de ordenadores, con un sistema de programación propio para ella y distintas aplicaciones.

Este proyecto está dirigido por H.T.Kung, y cuenta con la participación más nutrida de profesores de CMU. En la propuesta a DARPA, colaboran Thomas Gross, Jon Webb, Michael Browne, Bernd Bruegge, Guy Jacobson y Onat Menzilcioglu. Este proyecto utiliza el sistema operativo MACH como el sistema base, así como el simulador COSMOS, desarrollado por Randal Bryant.

A su vez Nectar es la base de un proyecto piloto denominado GIGABIT NECTAR TESTBED, que es una red local de alta velocidad de ordenadores heterogéneos que sirve de prototipo de las redes que se construirán en los años 90s.

Un tercer proyecto que analizaremos es el denominado NEW CONCEPTS FOR ADVANCED HIGH DEFINITIONS SYSTEMS,(1991-93) (CMU-SCS-Robotics Institute, 1991) dirigido por Angel Jordan. Una particularidad de este proyecto aun no perteneciendo al área de Computer Systems, está produciendo un nuevo sistema informático.

Su centro, dentro de la School of Computer Science, no está en Computer Science sino en el Robotics Institute, donde está adscrito el profesor Angel Jordan , que a su vez tiene "joint appointment" con el Electrical and Computing Engineering Department de la Carnegie Institute of Technology (CIT).

Los ocho investigadores de este proyecto están repartidos entre ambos centros.

Este proyecto plantea la fusión de la tecnología del video con la de los ordenadores. La particularidad del mismo es que no parte de los actuales temas de computer systems, como "paralelismo" vs. "arquitecturas Von Newmann", sino la tecnología de la imagen y la utilización por éstas de determinados avances en computer science como el sistema operativo MACH, a fin de construir una televisión informática, un denominado "advanced high definition system".

4.1.3. 1964. "NO REAL EFFORT IN HARDWARE".

Antes de entrar en el análisis de estos proyectos y su lugar en Computer Science en CMU, es preciso conocer cuál ha sido la evolución de esta área.

Como ya hemos indicado anteriormente, la área de "hardware" en Computer Science no se consideró una área de investigación en la propuesta inicial elevada por Allen Newell, Alan J. Perlis y Edward R. Schatz a la Advanced Research Projects Agency el 27 de abril de 1964, y que fue la base de la investigación en este departamento.

Su objetivo era "understanding the nature of information processing", y aunque se admitía que hasta ese momento esta área era una tecnología, el propósito era hacer de ella una ciencia: **"The attempt to produce a science in an area that not long ago was purely a technology"**.(Newell, Peris, Schatz,1964:1).

Se afirmaba que: **"the field is concerned with information, the systems that processes and transform it, and the way it is used to control, integrate and coordinate other systems"**. (ib:2). Se partía de la información, y en segundo lugar de los sistemas que la procesaban. Ello invertía el orden que hasta entonces había seguido la tecnología informática, que había sido primero una máquina, y sólo después una máquina programada. Ahora, la información se ponía en primer plano, y el estudio científico de la misma como objetivo central.

En consecuencia, se afirmaba: **"The center as organized contains no real effort on hardware research...It stands on its own feet as a software oriented group"**. (ib.:41). El centro sólo contenía un pequeño grupo de ingenieros, adscritos conjuntamente a la School of Engineering. No obstante, se dejaba la puerta abierta a trabajar en aquél tipo de hardware que estuviera relacionado con la actividad de programming del centro.

El problema era que la denominada "naturaleza" del information processing había partido, en gran medida, de un invento tecnológico, y para entender esta "naturaleza" había que seguir desarrollando este invento. El software no podía funcionar si no era software de un ordenador. Esta separación entre hardware y software ha sido especialmente perjudicial para la área de Inteligencia Artificial, pues ha contribuido a dificultar la construcción de máquinas aptas para procesar símbolos y a extender el uso de la A.I. por la comunidad de computer scientists.

Por otra parte, el patrocinador principal de este centro era el Department of Defense, y no la National Science Foundation. Los científicos eran financiados por los militares, y una investigación en Inteligencia Artificial sin máquinas que la validaran, era inviable. El enfoque sistémico, o "total system effort", de la ingeniería se superponía al mero analítico de la ciencia.

Allen Newell admitió la investigación en la área del hardware. En 1966 empezó a colaborar con Gordon Bell, profesor de Electrical Engineering y Computer Science entre 1966 y 1972, en el diseño de multiprocesadores.

El inicio de la investigación en CMU sobre hardware fue favorecida por un estudio patrocinado por ARPA a inicios de los 70s sobre cuál sería la más apropiada futura generación de ordenadores para la Artificial Intelligence.

Según la proposal remitida a ARPA por CMU en 1976: "**Although CMU's involvement with multiple computer systems dates from the G-21 (a two-processor multiprocessor) in the early 1960's, the relevant history dates from two events in the early 70's. The first of these was a study sponsored by ARPA into the nature of the appropriate next generation computer for AI**"

(CMU-CS-Proposal.1976:26).

El estudio concluía con la recomendación de diseñar el sistema multiprocesador denominado "C.ai".

Por otra parte, la tesis doctoral de un estudiante de CMU, Strecker, que construyó un modelo analítico sobre la permanencia de memoria de un multiprocesador, permitió entrever que esta memoria no se degradaría al ser compartida por varios procesadores. Ambos hechos, junto con el la invención de microprocesador en 1971 que rebajaba el precio del computing, llevaron a la construcción en CMU de la máquina C.mmp., un multi-mini processor diseñado por Gordon Bell y W. A. Wulf en CMU.

La Inteligencia Artificial no podía desarrollarse sin un avance del hardware. En 1971, Gordon Bell y Allen Newell escribían el libro "Computer Structures", abriendo la línea de investigación en "Computer Systems" en el departamento. En 1972, Daniel Siewiorek, un electrical engineering de Stanford, se incorporaba a CMU como uno de los líderes en este campo. (Siewioreck, Bell, Newell, 1981).

4.1.4. LOS MULTIPROCESADORES DE CMU Y EL COMPUTER ENGINEERING: GORDON BELL.

La apuesta por los multiprocesadores ha sido una batalla que ha llevado Gordon Bell durante 30 años, a contracorriente de la mayoría de la industria. Este "electrical engineer" del MIT, empezó trabajando en Digital Equipment Corporation de 1961 a 1966 antes de venir a CMU. Aquí pasó 12 años como profesor de Computer Science y Electrical Engineering, donde desarrolló su diseño de multiprocesadores.

En el 25 aniversario de Computer Science de CMU, Bell ha reafirmado su apuesta en favor de este tipo de ordenador que en su opinión facilitará la quinta generación de ordenadores.

Los multiprocesadores son un tipo de ordenadores en paralelo. El tipo de ordenador mayoritario durante estos primeros 40 años de cultura informática han sido los ordenadores basados en las denominadas arquitecturas Von Newmann, el matemático que diseñó su estructura lógica. Estos ordenadores son máquinas secuenciales que realizan una operación (computación) al tiempo. Ello ha tenido dos resultados, un uso ineficiente de los avances en circuitos integrados, y dos, un tipo de programación secuencial, "a word-at-a-time thinking", que no permite al programador pensar en términos de grandes unidades conceptuales de tareas a resolver.

La Inteligencia Artificial ha sido la área más perjudicada por este tipo de arquitecturas.

Este tipo de arquitectura está en crisis. Y está en crisis todo el modelo empresarial de esta primera etapa de la cultura informática. El Proyecto japonés de Quinta Generación ha apostado desde principios de los 80s por un tipo de arquitectura distinta.

La alternativa empieza a denominarse: "**parallelism revolution**", el diseño de un nuevo tipo de ordenadores que está permitiendo un nuevo salto en la capacidad de procesamiento de los ordenadores. (Pelaez, E. 1990:69).

Las arquitecturas en paralelo permiten a varios procesadores trabajar en varias tareas al tiempo. Si lo comparamos con la construcción de una casa, el ordenador Von Newmann se basa en que un obrero construye la casa de principio a fin, paso a paso. Por contra, los ordenadores en paralelo permiten que varios obreros juntos construyan la casa coordinadamente. El principal problema en este tipo de sistemas es planificar, organizar y coordinar el trabajo de los procesadores.

Los ordenadores en paralelo se pueden clasificar atendiendo a varios problemas. El primero es la relación procesador/instrucciones. Si todos los procesadores reciben la misma instrucción, se denominan SIMD o "single-instruction-stream". Así funciona el ordenador Connection Machine desarrollado en el M.I.T. y construido por Dan Hillis de Thinking Machines Co. (Hillis, W.D.1985). Tiene 64.000 procesadores que trabajan al unísono según una misma corriente de instrucciones y con una memoria distribuida. Por contra, si cada procesador recibe su propio conjunto de instrucciones, se denominan MIMD o

"multiple-instruction-stream". Máquinas como WARP de H.T.Kung funcionan según este modelo.

Un segundo criterio de clasificación es la relación memoria/procesador. En las arquitecturas Von Neumann el procesador ("Central Processing Unit") está separado de la memoria. En las arquitecturas en paralelo memoria y procesador se diseñan conjuntamente, habiendo dos alternativas: la memoria compartida o ("share memory") en la cual los procesadores comparte la misma memoria. O la memoria distribuída ("distributed memory"), por la que cada procesador tiene su propia memoria, en cuyo caso el principal problema es el de la comunicación entre procesadores o "communication problem". Máquinas como Cray X-MP siguen el primer modelo, mientras que Warp o Connection Machine el segundo.

La alternativa de Bell es que varios procesadores puedan trabajar simultáneamente con una memoria compartida ("shared memory").

Según él, la idea de los multiprocesadores se puede entender con el símil del carro de caballos. Un ordenador uniprocador tiene un caballo atado al carro y su poder lo obtiene de acelerar la velocidad del caballo. El multiprocesador obtiene su poder de computación con decenas y potencialmente cientos o miles de caballos arrastrando juntos el carruaje, esto es, trabajando juntos en la misma aplicación.

Para el Computer Science and Technology Board : **"Perhaps the greatest promise lies in the evolution of multiprocessors...Development of this technology is motivated by the fact that current computing power is wholly inadequate, by orders of magnitude, to perform most the interesting applications of the future...That power should help multiprocessors achieve ambitious new applications of artificial intelligence, such as real-time speech understanding, machine vision, learning, natural language understanding, and better machine reasoning..."** (Dertouzos,1988:10).

Este tipo de máquinas, como la C.mmp, se empezó a diseñar en CMU a principios de los 70s. Y su resultado inmediato fue doble: uno, permitir el diseño de máquinas más aptas para el uso en Inteligencia Artificial, y dos, institucionalizar la investigación en Computer Systems. Una nueva disciplina nacía a caballo entre electrical engineering y computer science: el "computer engineering", aunque no se reconocería hasta años más tarde. En los años 80s, el departamento de Electrical Engineering, pasaría a denominarse, bajo la dirección de S. Director, Electrical and Computer Engineering. Hoy dicho departamento es el más grande del

Carnegie Institute of Technology y tiene entre sus áreas de investigación las siguientes: "**digital electronics, computer architecture, computer-aided design and manufacturing, robotic and control systems, telecommunications and computer networking, communication systems, signal and information processing, solid state physics and devices, electromagnetic and electromechanical systems, magnetic information technology, digital signal processing, and optical data processing**". (CMU. Undergraduate Catalogue, 1990-92:87).

Por su parte, el Department of Computer Science en los 70s abrió una cuarta área: el Computer Systems, permitiendo que "computer engineers" como Gordon Bell pasaran a trabajar en Computer Science.

Este paso del Computer Science Department hacia la ingeniería fue un precedente de otros como la apertura de una área de Robotics en 1979 entre ingenieros y computer scientists, la apertura del Software Engineering Institute en 1984.

El diseño de ordenadores ha influido decisivamente en la evolución del Department of Computer Science, aunque el modelo de sistema informático ha variado con el tiempo. De los multiprocesadores de los 70s se pasó a los llamados sistemas distribuidos o "distributed systems" y los "multicomputers" en los 80s. En esta evolución influyó una nueva tecnología, la denominada "Very Large Scale Integration" (VLSI), que permitía integrar millones de transistores por microprocesador, y aumentar enormemente la velocidad de procesamiento. Ello, unido al fenómeno de las redes informáticas, que permiten a los ordenadores intercomunicarse entre sí, ha dado protagonismo a un modelo de paralelismo, el "heterogeneous multicomputer" distinto al de Bell.

Pero el efecto de esta tendencia del Computer Science hacia la ingeniería no se ha borrado. Jóvenes matemáticos como H.T.Kung se han puesto a construir "computer architectures" pasando de la área de "Theory", la más abstracta y menos ingeniera de la escuela, a estar adscritos al Electrical and Computer Engineering Department.

La investigación en Computer Science de CMU quedará así unida inseparablemente una área que al principio no se pretendía investigar. Incluso hoy día líderes como Raj Reddy, hacen depender el futuro de la A.I., su propia rama, que aún es la mayoritaria en la escuela, de la aparición de los llamados "3G supercomputers" previstos para finales de la década de los 90s. (1991:465)

El Programa japonés de la Quinta Generación ha reforzado esta urgencia de un nuevo tipo de hardware. A diferencia del enfoque norteamericano en Artificial

Intelligence, que ha separado sistemáticamente el nivel del software, de el del hardware, al menos en CMU , la Quinta Generación se ha cuidado de construir una máquina razonadora, una Parallel Inference Machine. Y ha sido en torno a esta nueva máquina que se han desarrollado lenguajes lógicos y las aplicaciones. Este proyecto ha finalizado sin demasiado éxito. No obstante, han iniciado un segundo , el ordenador neuronal, a desarrollar entre 1992 y el año 2000, basado en la idea de redes de miles de procesadores imitando el funcionamiento del cerebro humano. (Fuchi, K. 1991)

La respuesta norteamericana fue el programa Strategic Computing de DARPA de 1983. La estrategia seguida ha sido apoyar diversos sistemas de computación en paralelo, entre ellos los que se han construido en CMU por H.T. Kung, el iWARP y NECTAR.

A su vez, ha apoyado avances en la tecnología de VLSI que permiten construir las anteriores máquinas.

Por su parte, Gordon Bell, inició en 1983, el proyecto ALFA-OMEGA en computadores avanzados para la Microelectronics & Computer Technology Corporation, un consorcio de 17 empresas norteamericanas construido en 1983 como respuesta empresarial al Proyecto japonés. (Feigenbaum, McCorduck, 1983:289).

El tipo de investigación se realiza hoy en Computer Systems en la School of Computer Science es el resultado de este proceso, marcado por dos grandes ejes: la computación en paralelo y la tecnología del VLSI.

4.2.LA LLAMADA "BASIC RESEARCH IN COMPUTER SYSTEMS".

4.2.1. LA INVENCION DE LA TECNOLOGIA DE VLSI.

En el 25 Anniversary de Computer Science, Raj Reddy dedicaba su discurso a unas nuevas máquinas, denominadas "3G Supercomputers". Estos ordenadores disponen de una gran potencia de procesamiento de información que se calcula que a finales de siglo podrán estar en el mercado al precio de los actuales ordenadores personales.

Según Reddy, estas máquinas podrán realizar algo que ha sido imposible hasta ahora: poner al alcance del gran público la Inteligencia Artificial. Para ello se necesitan máquinas con las que interactuar mediante la palabra y la imagen, y así ser utilizadas para tareas inteligentes como consultar diagnósticos o realizar traducciones simultáneas.

La informática representada por el ordenador personal no ha llegado al conjunto de la población como lo ha hecho el televisor. Sólo la mitad aproximadamente 100 millones de hogares norteamericanos tienen ordenador personal y su uso se reduce a tareas elementales como procesamiento de textos o videojuegos. El computing es una herramienta cuya difusión aún no es comparable con tecnologías como el teléfono o la televisión, y esto, en países industrializados avanzados.

A diferencia de otras tecnologías como el automóvil, cuanto menor es la preparación del usuario en informática, más sofisticado, potente, e inteligente, ha de ser el ordenador para poder interactuar con él. Ello conduce a la paradoja que el ordenador que puede interesarle tanto a un niño en edad escolar es aquél con el que pueda relacionarse con la palabra y la visión, y el que se pueda programar sin necesidad de conocer lenguajes formales de programación. Y ello exige un poder de procesamiento más alto que los de los actuales ordenadores personales e incluso que las estaciones de trabajo o "workstations".

Igualmente la Inteligencia Artificial, la área menos formalizada de Computer Science y la más cercana a las ciencias sociales, es la que necesita una mayor potencia informática para sus programas de procesamiento de símbolos. Los multiprocesadores, ordenadores con la mayor potencia de procesamiento, se empezaron a construir como el tipo de computadores que necesitaba esta área. Esta relación máquina-complejidad de la tarea es constante y podría formularse como sigue: A mayor complejidad de tarea a realizar por el ordenador, mayor potencia y complejidad precisa el sistema informático. Las tareas más complejas son las relativas al diseño de sistemas culturales complejos, uno de ellos, la simulación del funcionamiento de la mente humana.

Por lo tanto, el diseño en áreas de conocimiento complejas como las que estudia la antropología u otras ciencias sociales exigirá el máximo poder de computación conseguible por la máquina. El diseño de tecnoculturas, aún a niveles micro, precisará de un sistema informático con las máximas y más complejas capacidades de procesamiento. Incluso como más adelante veremos, es posible que ya se precise de un tipo de máquina distinta al ordenador para ello.

Esta relación indica simplemente una proporción directa entre complejidad cultural humana y complejidad tecnológica, y confirma el hecho que la tecnología no es más que un "man-made world", y que este mundo se complejiza utilizando más y más complejos y más simples artefactos técnicos. Pero una vez llegados a este

punto, como vimos con los "embedded computer systems", se precisa construir sistemas modulares, apropiados para tareas concretas, no "universales", como aún pretende el ordenador.

La informática tal como la conocemos ahora ha llegado a un límite. La crisis actual de IBM expresa este límite. Así explicaba Raj Reddy esta crisis con motivo del 25 aniversario de la institución: "What happened?. We discovered that creating a "plug and play" home computer system requiring little or no additional training by family members required a degree of sophistication in software and user interface technology that did not exist. The graphics were inadequate, the computational power was inadequate, the sound quality was inadequate, and the applications were nonexistent". (1990:467).

Según Reddy, las tres condiciones básicas para un uso masivo del ordenador son:

1. que las máquinas 3G estén disponibles a un bajo costo.
2. Que dispongan de una interface fácil de usar por el ser humano, y
3. Que se creen nuevas aplicaciones que mejoren la productividad de los "knowledge workers", la mayor parte de la población activa en U.S.

La condición inicial de todo el ciclo es una máquina potente y barata. Y la condición para lograrla: que siga cayendo el precio del computing al tiempo que sigue aumentando su potencia, según Reddy.

Es significativo de esta etapa de CMU que un líder destacado de la Inteligencia Artificial admita que el avance del uso de la A.I. dependa como primera condición de un avance de la tecnología de microprocesadores. Recordemos que en un principio, la Computer Science en CMU no contemplaba la investigación en "hardware", sino sólo en "software".

Reddy ve este proceso como inexorable: "**The cost will follow the inexorable exponential improvement of silicon chips and computer systems**". (ibid:467).

Pero los "3G supercomputers" a precio asequible son una nueva posibilidad, una nueva "oportunidad", no una necesidad inexorable. En 1964, el Computer Science de CMU decidió no investigar en hardware y el escaso peso de esta área en CMU ha tenido consecuencias, como por ejemplo el retrasar las aplicaciones de la A.I., al no disponer de un hardware adecuado.

La visión determinista de los procesos tecnológicos resulta ya difícil de mantener. No todo depende del descenso del precio del microprocesador sino también de cambios en la arquitectura de ordenador, así como de pensar en términos de producto para el mercado, no sólo de avance tecnológico en sí. Es un

conjunto de aspectos culturales relacionados entre sí lo que cuenta.

El problema con la predicción científica, al modo de "technology forecasting", que pronostica qué tendencia van a seguir los avances tecnológicos es que por definición va siempre por detrás de la tecnología. Se afirma, por ejemplo, que, en base a la tecnología VLSI, el precio del computing sera x al final de siglo. Pero esta visión presupone que ninguna nueva tecnología dejará obsoleta la de VLSI. Siguiendo este método, un pronosticador tecnológico habría previsto en 1945, que según el avance de la tecnología de los tubos de vacío, que los ordenadores sólo serían una sofisticada máquina para un reducido número de instituciones , debido al alto precio de dicho tubos. O que no surgirá un nuevo tipo de máquina que deje obsoleto al ordenador.

CMU, por suerte, no sólo hace "technology forecasting" sino que investiga en VLSI. La escuela de ingeniería, el Carnegie Institute of Technology, tiene uno de los dos centros nacionales de excelencia en el diseño de circuitos integrados a gran escala, el SRC-CMU Research Center for Computer-Aided Design. El proyecto de VLSI Logic Validation de Randal Bryan permite la verificación de dichos circuitos.

¿Qué es la tecnología VLSI , quien y para qué se inventó?.

El componente básico de todo "computer system" es el chip o circuito integrado. Estas pastillas de silicio pueden contener millones de transistores, diodos, y otros dispositivos que son los que permiten incrementar la capacidad de procesamiento de información de los ordenadores. Cuanto más pequeños son los transistores más pueden integrarse en un chip, y más puede aumentarse la potencia de los ordenadores. . Hoy la pastilla normal de silicio de 6 pulgadas contiene 5 millones de transistores. En 1999, la capacidad potencial será de 3.500 millones de transistores. (Rappaport& Halevi, 1991:73).

Los semiconductores son dispositivos electrónicos cuyo funcionamiento está basado en las propiedades físicas de algunos materiales como el silicio o el arseniuro de galio. Estos materiales en estado puro son aislantes o no conductores , pero la presencia de determinadas impurezas en su estructura cristalina los dota de determinadas propiedades eléctricas que aprovechadas permiten construir dispositivos como los transistores o diodos. El diseño de estos dispositivos es el objetivo de la microelectrónica.

Los avances que han permitido que el ordenador se haya convertido en una industria puntera han tenido origen en un avance en la tecnología de los

semiconductores.

Estos han sido fundamentalmente cuatro. Y las cuatro han sido norteamericanas.

Por orden histórico han sido:

1. La invención del transistor en 1947.
2. La invención del circuito integrado en 1959.
3. La invención del microprocesador en 1969 y
4. La invención de la tecnología de VLSI a finales de los 70s.

En 1947, W. Shockley, W. Brattain y y Bardeen inventaron en los Bells Labs el transistor, contracción de "transfer resistance", un dispositivo que permitía amplificar las señales eléctricas, igual que las válvulas de vacío, pero con mucha menos corriente y calor, y de dimensiones mucho más pequeñas. Así explicaban J.Barden y W.H.Brattain su invento: **"A three-element electronic device which utilizes a newly discovered principle involving a semiconductor as the basic element is described. It may be employed as an amplifier, oscillator, and for other purposes for which vacuum tubes are ordinarily used."** (1948:230).

En 1956, Schockley, Brattain y Bardeen recibirían el premio Nobel de física por esta invención. El primero relata así la misma: **"Durante unos pocos meses a finales de 1947 y principios de 1948...Se realizaron las primeras invenciones sobre el transistor...Los detalles son...el fracaso de mis proposiciones sobre el efecto-campo en los transistores en 1939 y de nuevo, independientemente, en 1945,...Dejamos a un lado nuestros esfuerzos para construir un transistor. Pero la situación cambió el 17 de noviembre de 1947. Entonces Walter Brattain...realizó un experimento sugerido ese mismo día por otro miembro del equipo. El resultado fue la evidencia de que el efecto-campo podía funcionar...El transistor punta de contacto de Bardeen y Brattain nació el 16 de diciembre de 1947 ...Yo inventé el "transistor de junta" un mes más tarde"**. (cit. E Rogers, 1986:257).

La invención del transistor fue el hecho que transformó un descubrimiento científico en una pieza útil de un nuevo sistema tecnológico: el transistor.

Como Shockley afirma, como científico trabajaba en un laboratorio industrial realizando descubrimientos para ser aplicados en forma de transistores, el cual fue utilizado años más tarde por Bardeen y Brattain para inventar el transistor.

Este modelo Diseño 1-Descubrimiento-Diseño2, ya lo volvimos a ver en el programa Types in Programming.

El mismo Shockley inventó nuevas variantes de este mismo dispositivo, y

finalmente decidió crear una empresa propia en Palo Alto, el Shockley Semiconductor Laboratory, que reuniría al núcleo promotor de lo que sería la industria de los semiconductores: los llamados "Shockley Eight".

Esta participación de un Premio Nobel de Física en el inicio de la high tech industry, así como la cercanía de la Stanford University, se ha interpretado como que dicha industria nace de la ciencia básica y de la universidad. En realidad, la investigación que dio lugar a la invención del transistor nació en un laboratorio industrial, el Bell Lab, que es la división de I+D de la American Telephone and Telegraph Company, cuyo objetivo era producir inventos tecnológicos, sirviéndose de la ciencia como medio. Durante más de 50 años este laboratorio ha producido 10.000 patentes . Produce 1 patente diaria. Igualmente la participación de otros premios Nobels como H.Simon en la cultura informática, está erróneamente interpretado como que sus aportaciones nacen de los descubrimientos científicos. Ya vimos que el caso de la Logic Theory Machine fue al revés.

Por su parte, el papel de la Stanford University en la generación de la "high tech culture" del valle no ha sido homogéneo. Shockley venía de un laboratorio industrial, no de una universidad, y en relación con la industria, el papel clave lo ha jugado su escuela de ingeniería, de donde han salido las promociones de ingenieros-promotores que han ido a formar empresas de semiconductores en el valle. Del Departamento de Ingeniería Eléctrica salieron Frederick Terman, David Packard y William R. Hewlett, todos ellos ingenieros eléctricos y figuras fundacionales del Stanford Industrial Park en 1951.

Lo nuevo de la high tech no fue ligarse a la Universidad en general sino a una escuela de ingeniería de rango universitario. Ello convertía a la technology en "high", en tecnología de rango universitario. Pero a cambio, la universidad se apropiaba como institución dirigida por la ciencia del patronazgo de la "high tech". Con todo como el estudio de Everett Rogers y Judith Larsen demostró el tipo humano del Silicon Valley clave era el llamado "ingeniero-promotor" (ibid: 41)

En CMU Angel Jordan fue de los pioneros a finales de los 50s en iniciar la investigación en el Electrical Engineering Department sobre los semiconductores, realizando su tesis doctoral como él mismo explica: "on the analysis of a very special semiconductor device I had invented and started to develop for Syntron Company". (1976, A Personal Experience. Focus.). Pero esta línea no se consideró

interesante cuando se formó el Computer Science en CMU al no contemplar la investigación en hardware.

La segunda gran invención fue la de los circuitos integrados, los famosos chips, en 1959 realizada por Jack Kilby de Texas Instrument y Robert Noyce, fundador de Intel y discípulo de W. Shockley.

Esta invención generó en la industria de semiconductores un efecto conocido por los ingenieros. Gordon Moore, cofundador de Intel, en 1964 lo describió como la "learning curve" o curva de aprendizaje. La "Ley de Moore" advertía que el precio de un chip decrecía rápidamente tras su introducción debido a que las firmas aprendían a fabricarlo con menos defectos y a un coste más bajo. Por esta ley, cuanto antes se situara una empresa sobre la curva de aprendizaje mayor será su ventaja sobre los competidores. Esta curva recompensa la innovación tecnológica continúa, pues es la única vía segura para la supervivencia. La evidencia de esta ley se manifestará con la invención del microprocesador. (Rogers & Larsen, 1986:86).

Marcian E. "Ted" Hoff, un "electrical engineering" de Stanford, que trabajaba en Intel en 1969, fue el inventor del microprocesador, realizando un paso más en esta integración. Ted Hoff reunió en un solo chip la parte central de un ordenador: la CPU (Central Processing Unit), asociándole dos chips con la memoria del ordenador con los datos y el programa. Con ello conseguía introducir un ordenador en un chip, y abrió el paso a los miniordenadores y al ordenador personal.

En 1973 el microprocesador 8080 de Intel que incorporaba la tecnología MOS (Metal-Oxidon-silicon) utilizada para los circuitos integrados, salía al mercado. Digital Equipment Corporation y Apple deben su auge a esta invención. El microprocesador es lo que ha permitido un computing a bajo precio, y lo que hace posible la máquina de Reddy.

La llamada "high tech industry" es un fenómeno tecnocultural complejo. De hecho el término "high tech" se originó en la industria no es la universidad. El tipo de invenciones en que se basa las ha protagonizado como hemos señalado el "ingeniero-promotor" (Rogers y Larsen: 41). Pero esta figura ha tenido un límite al no estar acompañada de un mayor nivel de integración social. Después de más de una década, y de muchos intentos, el fenómeno del Silicon Valley no se ha repetido con esa fuerza en todo el país, quedando junto al de la Carretera 128 de Boston, reducido a un guetto tecnológico. (Florida & Kenney, 1990)

Por otra parte, la industria norteamericana de los circuitos integrados no inventó un nuevo modelo de organización que siguiera la pauta de integración de su propia tecnología. Permaneció internamente fragmentada y dividida. La tecnología de VLSI ha sido aprovechada ante todo por Japón, y sus grandes grupos industriales o "keiretsu". De hecho de las cinco principales empresas de semiconductores: NEC, Hitachi, Toshiba, Mitsubishi y Matsushita, las cuatro primeras pertenecen a los seis principales keiretsus: Sumimoto, Fuyo, DKB, Sanwa, Mitsui y Mitsubishi.

En 1975, la industria norteamericana copaba el 60% del mercado mundial de semiconductores. En 1987, sólo disponía ya del 40 % del mercado mundial, y la japonesa alcanzaba el 50%. La razón dada por el MIT para la pérdida de este liderazgo es que la empresa norteamericana, innovadora pero fragmentada no pudo seguir el cambio de modelo industrial cuando a finales de los 70s se impuso la tecnología del VLSI, una tecnología inventada en Norteamérica pero que exigía la producción en masa de chips más elaborados (Dertouzos, Made in America:259). Los grandes grupos industriales japoneses, orientados por la "industrial policy" del Ministerio de Industria y Comercio Internacional, el M.I.T.I., consiguieron imponer un modelo de producción industrial. El burócrata innovador más el ingeniero promotor japoneses, están venciendo al solitario ingeniero promotor norteamericano.

Un circuito integrado a gran escala (VLSI) es aquél que contiene 100.000 o más transistores o dispositivos semiconductores. La integración de dichos dispositivos en un chip de 6 pulgadas es un problema de diseño.

La tecnología de diseño de circuitos integrados a gran escala o Very Large Scale Integration (VLSI), fue iniciada en varios laboratorios industriales y desarrollada por Carver Mead de California Institute of Technology y Lynn Conway de Xerox PARC entre 1976 y 1977.

Las técnicas que estos dos investigadores desarrollaron, el llamado "the'Mead-Conway'VLSI design", tuvieron un rasgo único: las concibieron explícitamente como la generación de "new cultural forms" (Conway 1981:1). Este equipo se apoyó para avanzar en sus técnicas de diseño en un experimento organizativo novedoso: para integrar semiconductores en un chip, Lyn Conway y Carver Mead integraron diseñadores en la red ARPANET. El objetivo de esta red era la invención de metodologías de diseño de dichos circuitos. Comunicaron a los diseñadores del país sus diseños preliminares y recibieron dibujos,

correcciones, cambios que fueron integrando, hasta producir un manual Introduction to VLSI (1980), un clásico en esta técnica. Dentro de una enorme computer network (Arpanet), formaron una subred de diseño.

Lo interesante es que este fue un experimento de diseño técnico intentando a su vez una integración cultural más amplia. Según Conway: **"The sort of question that really interest me is: How can we best organize to create, validate, and culturally integrate new design methods in new technologies?...How can you cause the cultural integration of the new methods, so that the average designer feels comfortable using the methods, considers such usage to be part of their normal duties, and works hard to correctly use the methods"** (1981:2)

Desde esta red, establecieron una nueva relación entre las universidades y la industria. Trabajando desde la Xerox, ofrecieron a los profesores poner a prueba estos conocimientos, realizando "project-oriented courses" basados en el diseño de circuitos integrados, transmitir éstos a través de ARPANET a la industria, y tener las pastillas completas y empaquetadas al cabo de un mes de finalizado el curso escolar, lo cual permitiría probar si estos diseños funcionaban o no en el real world. Este experimento se denominó **"multiuniversity, multiproject chip network"** conocido como "MPC79". (Ib:5). Una red de diseño. Una docena de universidades participaron en la red, las de vanguardia en la área :MIT, Caltech, Stanford, Berkeley, y CMU. Y un total de 100 diseñadores, entre ellos Bob Sproull de esta última universidad

Es importante destacar que un avance en las metodologías de diseño y la construcción de una red de diseñadores haya permitido avanzar en la tecnología clave de la industria informática.

Unos años más tarde, en 1982, Lynn Conway y Mark Stefick, escribían el artículo "Towards the Principled Engineering of Knowledge", (Stefik & Conway, 1982) donde proponían que también el conocimiento, no sólo los circuitos integrados, podía ser diseñado, ser tratado con métodos ingenieros, siguiendo la experiencia de los experts systems iniciados por E. Feigenbaum. Junto al slogan "Knowledge is power" de Feigenbaum, añadían el de: "Knowledge is an artifact, worthy of design" (1982:4). Nosotros añadimos un paso más: "Knowledge is design". Y el diseño es un sistema de conocimiento base de un tipo de cultura distinta, una tecnocultura

La experiencia del diseño de VLSI, así como la de expert systems, confirma este principio rector de la high tech. Este sistema de conocimiento avanza con la

contrucción de redes de diseñadores integradas a gran escala, donde el diseño técnico es inseparable del diseño cultural. Este fue el intento que realizamos, prematuramente, en 1990 en CMU iniciando la construcción de una red de diseño que denominamos "Designet".(Serra, Rojo, 1990).

4.2.2. "VLSI LOGIC VALIDATION" PROJECT: DISEÑANDO HERRAMIENTAS PARA LOS DISEÑADORES DE CIRCUITOS INTEGRADOS.

Randal E. Bryant recibió de Lynn Conway el citado curso. Siendo estudiante de doctorado en el M.I.T. diseño el primer simulador y verificador de circuitos integrados a gran escala, MOSSIM. El presente proyecto en CMU continúa esta línea de investigación.

La innovación principal que el proyecto reivindica es la siguiente : "A switch-level simulator, capable of simulating circuits of up to 1.000.000 transistors when run on a uniprocessor, will be developed". (CMU-SCS-Basic Research in Computer Science. 1989:1-1). Se trata de desarrollar un simulador de circuitos integrados.

Si nos atenemos a la definición tradicional de "investigación básica", este proyecto no debería estar clasificado ni siquiera como investigación aplicada sino como "development". Ciertamente tiene una segunda parte que se denomina "Parallel Computation" pero como veremos su objetivo es desarrollar lenguajes de programación para la computación en paralelo.

El simulador en cuestión se llama COSMOS y ya se utiliza en docenas de universidades, industrias y laboratorios, entre ellos Carnegie Mellon, University of Washington, Lincoln Laboratories y California Institute of Technology.

Las funciones de este artefacto son tres: 1. La simulación de circuitos integrados a gran escala. 2. La validación de los mismos, y 3. Su test.

El proyecto pretende, en primer lugar, mejorar la función de simulación de circuitos. Hoy día los electrical and computer engineering diseñan los circuitos con ayuda de las herramientas CAD (Computer Aided Design). COSMOS es una herramienta que les permite preprocesar el circuito produciendo una representación booleana (lógica) del mismo. Hoy el límite está en 200.000 transistores y lo pretende elevar en tres años a 1 millón.

La segunda función, es la validación de dichos circuitos. Para ello adopta un enfoque denominado "formal verification". Este enfoque consiste en tratar el diseño del circuito como un sistema lógico y verificar que los diseños se

derivan de las especificaciones, igual que las pruebas de un teorema se derivan de sus axiomas. Este método se empezó a utilizar en "programming systems" y ahora se extiende al hardware.

Por último, COSMOS ayuda a generar pruebas para comprobar que el circuito esta libre de defectos. Es en este nivel cuando se acaba el ciclo denominado:

"Circuit Validation". La validación del circuito no acaba con la verificación lógica de sus especificaciones sino con la prueba que sigue a este mecanismo lógico. El circuito no es un teorema lógico sino un ingenio mecánico, y su funcionamiento es lo que vale, no que sea verdadero o falso.

Este aspecto conviene resaltarlo, pues es clave para entender el sistema de conocimiento en esta cultura. La lógica es un instrumento al servicio de la construcción de circuitos integrados no su fundamento, como lo fue ya en el caso del Logic Theorist de Newell, Shaw y Simon.

Igualmente, el llamado "test" en ingeniería tiene un carácter distinto a lo que se denomina "test" en ciencia experimental. En el primer caso, como vemos, se trata de comprobar que el sistema diseñado funciona sin defectos. En el segundo, la prueba sirve para verificar que las hipótesis explicativas a cerca de un sistema estudiado son ciertas.

Pero todo esto está de momento confundido en un equívoco terreno denominado Computer Science, en el cual se denomina "basic research" al desarrollo de dispositivos electrónicos como COSMOS, hechos por ingenieros como Randal E. Bryant, que ostentan el cargo de "Associate Professor of Computer Science" en la SCS, y al mismo tiempo es profesor en el Electrical and Computer Engineering Department del Carnegie Institute of Technology.

4.3.3. "PARALLEL COMPUTATION" PROJECT: DISEÑANDO LENGUAJES PARA LA PROGRAMACION EN PARALELO.

La segunda parte de este programa de investigación en "basic research" se denomina "Parallel Computation". Pero como veremos su campo se reduce de hecho al tema del "parallel programming".

El proyecto parte del supuesto que el cuello de botella de la siguiente generación de ordenadores, los ordenadores en paralelo, ya no está en el diseño de sus arquitecturas sino en la falta de lenguajes de programación para ellas.

El proyecto apuesta por lo que denomina "data parallel languages", un tipo de lenguajes que combine los lenguajes funcionales como el llamado "C*" con los

utilizados en máquinas vectoriales como Cray.

Los ordenadores en paralelo permiten y exigen un tipo de programación distinta a los ordenadores Von Neumann. En lugar de utilizar lenguajes de programación secuenciales permiten el uso de lenguajes funcionales basados en el álgebra y en la lógica. Estos permiten al software engineer concentrarse en el problema a resolver, más que en cómo ha de resolverlo la máquina. A cambio la programación en paralelo es más compleja que la secuencial con lo que amenaza a convertir la "software crisis" en una "super-software crisis" (Pelaez, E. 1990: 75).

Este proyecto de CMU pretende lo siguiente: "We propose a multidisciplinary program of research to develop language designs and implementation technology for portable, general purpose parallel programming. This work will be based on innovation in three key areas...:languages, intermediate representations and compilation techniques. In addition to the usual process of writing and testing code, a critical part of our work will be to apply techniques and insight from semantic analysis to all three areas"(CMU-SCS-Basic Research in Computer Science: VLSI Logic Validation and Parallel Computation,1989:7-9).

Es pues una investigación básica centrada en el diseño de lenguajes de programación: "Our goal...is to learn how to design data parallel languages that optimize both ease of programming and performance of compiled code across multiple architectures".(ibid.,7-9)

Se trata de un tipo de investigación básica centrada en saber como diseñar sistemas informáticos, centrada, pues, en el diseño.

¿Cuáles son los pasos de esta investigación? ¿Cuál es su plan de trabajo?.

1. El punto de partida de la misma, según los autores, son las invenciones técnicas diseñadas anteriormente: "The starting point for this research will be the more specialized techniques that we and others at Carnegie Mellon have developed in previous work with W2, Apply, AL, Slang...and in programming a variety of machines including Warp machines,...the TMC Connection Machine, the iPSC hypercube, and the Cray XMP and YMP."(Ibid. :7-16).

2. El siguiente paso es el diseño de los rasgos fundamentales del lenguaje, las representaciones intermedias y las técnicas de generación del código basadas en la experiencia anterior y las ideas del proyecto: "We will design initial language features, intermediate representations, and analysis and code generation techniques based on generalizations from this experience and the ideas described in this proposal" (ibid:7-16).

3. El siguiente paso será, para los autores, la construcción de las implementaciones iniciales: "The next step...will be to build simple, easy to change and upgrade, initial implementations. We will design a small but useful parallel programming language to serve as a testbed for language features, in isolation and in combination". (ib: 7-16).

4. Una vez construido el lenguaje, el proyecto empezará a escribir una serie de programas piloto, en diferentes máquinas: "With the initial implementations, we will begin to code a series of example programs, covering an increasingly wider set of algorithms and programming styles." (ib:7-17).

El proyecto seguirá, según sus autores, las siguientes etapas: 1. Problema inicial, ideas para solucionarlo y experiencias adquiridas. 2. Diseño de los lenguajes. 3. Construcción de los mismos. 4. Prueba de ellos, usándolos para escribir determinados programas en varias máquinas distintas.

Esta es la sucesión de etapas clásica de un proyecto ingeniero: 1. Analysis of the problem. 2. Design. 3. Building. 4. Test

Una vez desarrollados estos lenguajes, estos profesores los enseñarán en sus asignaturas de parallel programming y los ofrecerán a otros proyectos de CMU para su uso.

A su vez, utilizarán las semánticas formales desarrolladas por Dana Scott para la implementación de un correcto lenguaje: "We will also use semantics as a touchstone for correct language implementation: A precise statement of the abstract semantics of a language construct can be used as a specification for its implementation". (ib:7-18).

Finalmente, se incorporarán las experiencias ganadas en esta investigación para elaborar nuevos modelos teóricos de computación en paralelo: "We will also work on incorporating the insights gained in the practical aspects of the research into new theoretical models of parallel computation". (ib:7-18). Modelos basados en los sistemas diseñados y construidos previamente.

Este es un proyecto de "basic research" en la área de Computer Systems de CMU. Sus autores, Guy E. Blelloch, Stephen D. Brookes, Edmund M. Clarke y Allan L. Fisher son profesores de Computer Science en la escuela.

Por último, los autores plantean que su proyecto avanzará sinérgicamente con otros proyectos como iWarp y Nectar, proyectos estos que estos "computer scientists" consideran "applied work". Mientras que este proyecto de "basic research" cuesta 2.575.000 dólares, para el conjunto de los tres años, los de

"applied research" costarán ocho veces más.

4.3. "THE NECTAR PROJECT".

4.3.1. EL "NECTAR PROJECT" Y LA "SCHOOL OF COMPUTER SCIENCE".

Nectar Project es en el curso 1991-92 el proyecto con el mayor número de profesores de la School of Computer Science (8), y con un mayor presupuesto: 19 millones de dólares para el periodo 1990-93.

Este hecho es tanto más significativo en un Departamento que no tenía en su inicio la intención de investigar en hardware, e indica una línea de lenta y costosa evolución de Computer Science en CMU hacia la ingeniería y el diseño, antes señalada.

El proyecto oficialmente se denomina: " RESEARCH ON PARALLEL COMPUTING: NETWORK-BASED MULTICOMPUTERS". (CMU-SCS- Basic Research on Parallel Computing, 1989). La dirección es del profesor H.T. Kung, y fue presentado a la Defense Advanced Research Project Agency, el 28 de diciembre de 1989, respondiendo al anuncio oficial 89-05 de dicha agencia. Su periodo de duración es de tres años inicialmente.

Los resultados o "deliverables" que este proyecto piensa entregar a DARPA al final de este contrato son:

1. "Nectar, a network backplane for multicomputers", ésto es, una red conectando ordenadores de distinto tipo (estaciones de trabajo, ordenadores vectoriales, ...), y permitiéndoles trabajar coordinadamente en una misma aplicación.

Esta red se compondrá de dos sistemas: "Nectar I system", basado en un prototipo ya construído, y que puede incorporar 128 nodos o conexiones distintas. Y "Nectar II System", que pretende servir de modelo como red capaz de conectar los ordenadores ultrarrápidos o "high-performance computers" de mediados de los 90s.

2. Un sistema de programación en computación en paralelo adecuado a la red: "Parallel and network-level programming support".

Y 3. Aplicaciones ("applications"). Entre ellas, un sistema de sonar multicanal, y el uso del simulador COSMOS funcionando en la red.

Pero este proyecto no se queda a un nivel meramente experimental. Se plantea el desarrollo de productos, "products", realizados por industrias que han empezado a trabajar colaborando con el proyecto.

Uno de los productos a entregar es el ordenador iWARP, desarrollado conjuntamente con Intel, empresa que se encarga de su venta comercial, en primer lugar al propio Department of Defense.

Nectar I System está siendo también desarrollado por una empresa subcontratista del proyecto: la Network System Corporation.

Por su parte, Nectar II Systems ha sido adoptado por la Corporation for National Research Initiatives, como proyecto piloto para el desarrollo de una red local de alta velocidad (gigabits, mil millones de operaciones por segundo).

Por último, las primeras aplicaciones han sido con el Naval Air Development Center (NADC) y con diversos departamentos de CMU, en primer lugar con el Robotics Institute, cuyo proyecto NAVLAB, un vehículo terrestre experimental totalmente automático que funciona sin conductor, utiliza el ordenador iWARP. Nectar es un sistema informático, un "computer system": hardware, software y aplicaciones. Y este sistema está basado en la construcción del ordenador, de la máquina.

A nivel teórico, este sistema se basa en un paradigma denominado "complex systems". Según H.T. Kung, se trata de una nueva vía en computer science: "Building complex systems" (Entrevista 15-3-91). Este enfoque prioriza el pensamiento de conjunto, "the whole thing".

Es de destacar que hasta ahora el paradigma dominante en la SCS ha sido el de la Inteligencia Artificial, liderado por Simon y Newell. Según este modelo, el Programming Systems y el Computer Systems son instrumentos de la A.I. para desarrollarse. Y así sigue viendo Raj Reddy las máquinas 3G Supercomputer, instrumentos para que el reconocimiento de voz y la A.I. llegue al gran público. Pero el paradigma de Kung es distinto. En este enfoque, las aplicaciones de A.I. u otras son un test para probar que su sistema funciona. Lo primero es el sistema no sus partes. De hecho, Nectar no es sólo una área de "Computer Systems", sino un modelo de organización que trata de abarcar el conjunto de áreas de la investigación en la SCS.

El modelo que se deriva de este paradigma de la "complexity" es que KUNG define como "heterogeneous multicomputer", que es uno de los posibles sistemas de computadores en paralelo.

H.T. Kung defendía así su modelo en un discurso con motivo del 25 aniversario de Computer Science Department: "A heterogeneous multicomputer is a network-based computing environment allowing two or more different kinds of computing resources cooperating on a single application...Heterogeneous multicomputers represent a new direction in high-performance architectures. ...allows simultaneous use of the best suited architectures to carry out computations of

different characteristics". (Kung,1990:237-238).

El sistema operativo MACH introdujo esta idea de sistema-plataforma, o sistema unificador que es el que permite que los diversos sistemas operativos funcionen conjuntamente. NECTAR extiende este modelo a la red informática. Es un sistema comodín, que permite que ordenadores de distinto tipo funcionen simultáneamente. Tiene la misma relación con las restantes máquinas que la Administración guarda con los empresarios privados, estableciendo un terreno de juego común. Tanto MACH como Nectar crecieron como proyectos del Strategic Computing Program de DARPA.

El Nectar System está en funcionamiento en CMU desde noviembre de 1988. Es una red informática para ordenadores de alto rendimiento, cada uno de los cuales se considera lo que se denomina un "host" . Los "host" van unidos a potentes coprocesadores ("CAB: Communication Accelerator Board") encargados de acelerar sus protocolos de comunicación. A su vez, estos CABs están unidos mediante fibra óptica con distintos "HUBs" o conmutadores que realizan la función de controlar la distribución de los mensajes en el sistema. Estos "hubs", por último, conectados entre sí forman el "HUB cluster".

H.T. Kung se ha concentrado desde finales de los 70s en este problema, "the communication problem". A diferencia del enfoque tradicional que ha priorizado el hacer chips más veloces a nivel de conmutadores ("switch level"), Kung se ha especializado en acelerar la comunicación entre ellos, a fin de tener los datos y el control "in the right place at the right time" en el chip.

4.3.2. H.T.KUNG Y CH. LEISERSON: "THE SYSTOLIC ARCHITECTURAL CONCEPT".

Hsiang-Tsung Kung ha sido hasta 1991 Shell Professor de Computer Science en CMU, de donde ha pasado a la Harvard University. Nació en Shangai, China en 1945. En 1973 obtuvo el doctorado en CMU en Mathematics bajo la dirección del Dr. Joseph Traub, matemático de origen y director por entonces del Department of Computer Science. Su tesis se tituló : "Topics in Analytic Computational Complexity". Su interés por el problema de la "complexity" en la área de Theory, le condujo a interesarse por la complejidad de los VLSI, tecnología que a mediados de los 70s se iniciaba.

Como explica Charles E. Leiserson, profesor en el MIT, alumno de H.T. Kung y coinventor con él de este algoritmo: " In 1978, H.T. Kung led a group of CMU graduate students, which included myself, to investigate the power of VLSI circuitry. At the time, we were amazed by this computational medium. Every one

to two years, the number of devices that fit on a VLSI chip doubled...Many computer scientists who became involved in VLSI naturally focused on the problems of computer-aided design...H.T.Kung and his students took a different path, however. Rather than concentrating on design issues, we investigated computational issues: algorithms, architectures, and intrinsic capabilities of the medium". (Leiserson 1990:30).

Aquí Leiserson diferencia problemas de diseño de problemas computacionales. Ello permite apreciar que el computer scientist al desarrollar, al diseñar, un algoritmo para resolver un problema, piensa que no está haciendo diseño sino "computer science". A renglón seguido nos dice Leiserson: "At the time, parallel machine architectures, such as C.mmp and Cm*, were proliferating at CMU. Despite our attempts to predict the behavior of algorithms running on these machines from the abstract machine models they implement, we discovered that actual running times differed considerably. The abstract machine model covers overhead costs, such as communication, which must be considered explicitly to estimate running times." (ib: 30).

El factor tiempo de comunicación no se tenía en cuenta, quedaba encubierto en las máquinas abstractas del computer scientist. Pero en la tecnología VLSI los costes de comunicación eran explícitos: "In VLSI technology, however, we found that communication costs are explicit: the capital cost of communicating between two processors is the area of the wire connecting them. Thus, we embraced the VLSI revolution, because it could provide a model in which the abstract costs of algorithms and architectures correspond closely to real implementation costs. Kung and I developed an accurate, yet abstract, model of VLSI computation, called systolic computation." (1990:30).

El equipo de computer scientists liderado por H.T.Kung pasaban a abrazar un paradigma tecnológico, directamente ingeniero, "the VLSI revolution", porque proporcionaba un modelo en el que los algoritmos y arquitecturas abstractos podían implementarse con costes reales. El paso que dieron fue dedicarse a diseñar un modelo de computación que pudiera ser construido mediante la tecnología VLSI.

Desarrollaron una estructura denominada "systolic arrays" que podía implementarse en hardware VLSI. Los "arrays" son una forma de estructurar los datos en la memoria del ordenador. La invención de Kung y Leiserson fue diseñar una nueva forma de comunicación de los datos en estas estructuras.

En 1979, en su escrito "Systolic Arrays (for VLSI)" estos "computer scientists" explicaban sus intereses en la investigación: "We are interested in high-performance parallel structures that can be implemented directly as low-cost hardware devices. By performance, we are not referring to the traditional operation counts that characterize classical analyses of algorithms, but rather, the throughput obtainable when a special purpose peripheral device is attached to a general purpose host computer". (1979:257).

Estos computer scientists pretendían diseñar ordenadores que funcionaran en tiempo real, y la tecnología VLSI al hacer caer los precios del computing permitía que un equipo universitario abordara esta tarea: "We are interested in designing multiprocessor structures which have simple and regular communication paths...Systolic systems provide a realistic model of computation which captures the concepts of pipelining, parallelism and interconnectio structures". (ib.:257).

En este modelo de comunicación, los datos eran "bombeados" por cada procesador (como el movimiento sistólico del corazón) a su más inmediato vecino. Dado que la velocidad de comunicación está limitada por la velocidad de la luz, la ordenación para la comunicación con el procesador vecino ("locality") producía una más rápida comunicacion de mensajes , a cambio de ser éstos más simples. Este modelo de chip, daría lugar a un modelo de arquitectura de ordenador, también denominada sistólica.

El centro de este proceso es el diseño de modelo de computación adecuado para la construcción de un tipo de computadores en paralelo.

Con ésto estos inventores confirmaban lo que Perlis indicaba como la aportación fundamental del "Computer Science", no los algoritmos, sino su unión con la máquina: "What will be new is the attachment of algoritms and languages to a machine-the electronic digital computer-which is the agent commanded to perform (to do , to accomplish) algorithms expressed in language". (1977:2).

Es interesante saber que este sistema sistólico se patentó como invento.

Leiserson y Kung inscribieron la U.S. Patent Application " Systolic Array Apparatuses for Matrix Computations" en Diciembre de 1978.

H.T. Kung por su parte inscribiría en 1985 una nueva patente con otros tres estudiantes de la SCS, F.H.Hsu, N. Teiji, y A.Sussman, denominada "Programmable Interconnection Chip for Computer System Functional Modules." Finalmente en 1991 recibiría el premio al "Inventor of the Year", concedido por la Pittsburgh

Intellectual Property Law Association, "in recognition of outstanding technological achievement and commercial implementation in the field of parallel computing, specifically in architectures used to deliver extremely high computation rates at low cost".

Preguntado, en la entrevista mantenida en CMU, acerca de cómo se definía él mismo, si como científico, ingeniero o ambas cosas, el afirmó: "I'm a problem solver, not a scientist. there is a tradition of problem solvers in this campus, like Herbert Simon. But I'm a minority in this school. The majority want academic positions. I don't care. I want to solve problems. If you have a problem I will solve it! (risas de Kung)"(15-3-91).

En esta entrevista con el Dr. Kung, este manifestó una particular visión de CMU y de su comunidad de computer scientists . Según él: " The real problem is how to increase the use of the computers. The speed is one of the problems...In the CMU computer science community nobody produces software. There is no real use of the computers in this high tech university. The people work without worry about the social responsibilities, as an intellectual hobby. The 90% of the faculty has no concern with the society, but only intellectual concerns...I want to help the society. The intellectual issue only is not sufficient for me. My point of view is in the minority" (Entrevista, 15-3-91).

Un efecto interesante de este invento de Kung y Leiserson ha sido el nutrir un nuevo campo teórico en Computer Science, la llamada "VLSI Theory". El profesor Leiserson finalizó el doctorado en 1981 con una tesis dirigida por H.T. Kung denominada: "Area-Efficient VLSI Computation", que trataba del diseño de sistemas sistólicos y del problema de determinar la área VLSI de un gráfico. Hoy lidera el grupo de "VLSI and Parallel Systems" del Department of Electrical Engineering and Computer Science del MIT. Uno de las ramas de este campo es la denominada "layout theory" (teoría del plan o diseño), que estudia como pueden ser incrustados ("embedded") gráficos en un plano de acuerdo con las reglas del diseño de VLSI.

Lo interesante aquí es ver cómo progresa la teoría en Computer Science. La "teoría" del VLSI ha nacido de la tecnología VLSI. Computer scientists han aprovechando herramientas matemáticas como los algoritmos para incrustarlos en las reglas del diseño ingeniero de VLSI. La computer technology permite descubrir algoritmos que pueden caber en dicho sistema. Como afirmaba Leiserson en su tesis doctoral d: " The remarkable advance of very large scale integrated

(VLSI) circuitry has sparked research into the design of algorithms suitable for direct hardware implementation. To exploit the massive parallelism offered by two-dimensional VLSI circuit technologies, the physical and the logical structures of algorithms must be harmonized". (CMU-CS-1981. Doctoral Dissertation.)

4.3.3. "WARP PROJECT": CAMBIOS EN EL MODELO DE INVESTIGACION DE LA "COMPUTER SCIENCE" DE CMU.

Una vez diseñada esta nueva forma de comunicación, H.T. Kung se dedicó al desarrollo de algoritmos sistólicos a fin de conseguir construir un chip integrado a gran escala que funcionara según este modelo y que pudiera probar que, en efecto, se podían construir máquinas más rápidas y económicas siguiendo este concepto.

Entre 1978 y 1983 los títulos de los principales papeles publicados por H.T. Kung nos indican en qué se centró su actividad:

1978. "Systolic Arrays (for VLSI)" . Junto con Leiserson.

1979. "Let's Design Algorithms for VLSI Systems".

1979. " Design of Special -Purpose VLSI Chips: Example and Opinions". (con M.J. Foster).

1980. "Special-Purpose Devices for Signal and Image Processing: An Opportunity in VLSI".

1981 "A Systolic 2-D Convolution Chip" (con S.W. Song).

1981. " Use of VLSI in Algebraic Computation: Some suggestions".

1982. " Why Systolic Architectures?".

1983. " Design of the PSC: A Programmable Systolic Chip". Junto con Allan Fisher, L.Monier, H. Walker y Y. Dohi.

En la propuesta remitida a DARPA en 1981 dentro del contrato básico de CMU para el periodo trienal 1981-83, (CMU-CS- Proposal, 81-83) la investigación en "Systolic Architectures" se inscribía dentro del capítulo denominado "Research in Integrated VLSI Systems". Y se planteaba como uno de los objetivos a largo plazo: "Understanding applied algorithm design for VLSI. The theory of systolic algorithms , begun at CMU in the last three years, will be extended and applied both to the design of special-purpose chips and to the design of programmable systolic arrays. The view of integrating more than one high-performance systolic device into a complete system will be pursued and evaluated". Diez años después,

el equipo de Kung ha conseguido con Warp y Nectar cumplir estos objetivos. A destacar que en el texto se habla de la "theory of systolic algorithms" que inicialmente fue una tecnología patentada.

En esta propuesta se expresaba el interés de continuar la línea de diseño de otros algoritmos sistólicos como los "database algorithms" pero el interés principal del equipo se dirigiría a otro tipo de estudio: "Although we will continue to study the properties of systolic algorithms themselves, our research for the next three years will be directed increasingly toward understanding some of the problems of how systolic arrays can be attached together or to conventional von Neumann computers. In other words, how are systolic algorithms to be incorporated into systems?".(1981:4-10).

Este tipo de planteamiento lo hemos visto repetido en otros proyectos, como el de Newell, Carbonell y Mitchell.

Diferencia al computer scientist del ingeniero tradicional, interesado ante todo en la construcción de sistemas, "building systems". Esta investigación pretende aumentar el conocimiento, no sólo hacer cosas. Los sistemas se hacen para probar que las ideas o conceptos iniciales funcionan. Pero estas ideas no se refieren a explicaciones de lo que existe y cómo funciona (T.Kuhn), al modo de la ciencia, sino sobre la construcción de diseños posibles. Es un saber sobre que-puede-ser ("can be") y como-se-hace-para-que-sea. Si la ciencia trata de descubrimientos, este saber no es científico, pero no podemos confundirlo con un simple "know-how". Es una nueva ingeniería teórica, o una teoría de los inventos tecnológicos, y por ende, de los inventos culturales.

Hasta la invención de la tecnología VLSI, los diseños del computer scientist se quedaban muchas veces, y aún ocurre, al nivel de meras abstracciones formales dado que no se implementaban al no disponer de un hardware apropiado dado su elevado costo. Tenía que utilizar las máquinas que la industria le facilitaba.

Pero ahora la tecnología VLSI ha empezado a permitir que las universidades puedan construirse los ordenadores adecuados para implementar sus proyectos.

Esto puede significar un cambio radical en un futuro.

Entre 1981-82 la actividad de investigación de Kung y su equipo se centró en el desarrollo del PSC o "programable systolic chip", un microprocesador en un sólo chip, de propósito especial y de alto rendimiento ("high performance"), construido con la intención de ser usado en grupos de decenas o centenares para la construcción de ordenadores sistólicos. En junio de 1983, este chip con

25.000 transistores empezaba a funcionar. A diferencia del modelo de Conway, ahora un equipo universitario pedía a la industria que le fabricara un nuevo tipo de chip, que por ser económico, era también como veremos comercializable. En la siguiente propuesta a DARPA de 1984 para los tres años siguientes, ya se planteaba un paso más: "For the next phase of the PSC research, we propose to study practical and theoretical issues related to the use of very large numbers of building-block processors like the PSC in a total system, with special emphases on testing, reliability, fault-tolerance and system support"(CMU-CS, Proposal,1984, 8-1). Este sistema será WARP.

Un resultado inmediato de este invento fue la construcción de arquitecturas de ordenadores ultrarrápidos para fines específicos.

Hasta ahora, el ordenador se ha considerado ante todo como una "universal machine" según el modelo de Turing . La universal machine es un ordenador denominado "general-purpose", que puede ser programado para cualquier fin. Pero desde el punto de vista económico, la "special-purpose parallel machine" tiene una clara ventaja sobre la universal machine. Son máquinas más baratas, eficientes y rápidas, aunque no tienen vocación de servir para todo. Pero tampoco lo son el coche o la lavadora. Y si para comprar un coche tuvieramos que comprar una combinación de coche-lavadora-lavaplatos-televisor, quizá se habrían vendido muchos menos.

El equipo de Kung prevee una nueva ola de ordenadores para muy distintas aplicaciones: "A wave of new computers for literally every computer application area will be investigated. They can be machines for AI expert systems, signal processing or playing chess. This new style of research calls for a highly integrated effort that takes into account important issues such as application, architecture, algorithms, technology, custom VLSI and system integration all in one framework". (Proposal,1984, 8-3).

Los "systolic systems" de Kung y Leiserson han dado un avance a este tipo de ordenadores. Una de las ventajas de los ordenadores en paralelo es que pueden ser construidos de muy diversas formas, dependiendo de los propósitos que se les asignen. No hace falta la máquina universal. Es posible construir una máquina para cada aplicación, conectadas o no entre sí por redes electrónicas.

WARP fue construido en CMU en colaboración con General Electric Co. y Honeywell, empleó un equipo de 70 personas. Es un ordenador en paralelo con arquitectura sistólica usado para computaciones para imagen y visión, áreas de Inteligencia

Artificial, y para cálculos científicos.

El proyecto WARP comenzó en 1984 completando una primera fase en 1985. Para su ampliación se realizó un contrato con las empresas antes citadas, entregando General Electric el primer prototipo de máquina en 1986 y llegando la de Honeywell unos meses más tarde.

Una de las más importantes consecuencias de esta investigación, junto con el MACH Project, fue permitir a CMU conseguir un macroproyecto de investigación financiado por el Strategic Computing Program de DARPA en 1986.

En febrero de 1986, H. T. Kung y Rich Rashid presentaban a DARPA su "Proposal for Research on Parallel Computing", (CMU-CS- Proposal, 1986). donde se proponía continuar la investigación en "heterogeneous architectures" como WARP , en sistemas operativos como MACH y en aplicaciones de éstos para AI.

Este programa tendrá tanto peso que se considerará separado del contrato básico histórico de CMU con dicha agencia gubernamental. Se pasara a denominar "STRATEGIC" en la SCS.

"STRATEGIC" instituyó en Computer Science de CMU un programa de investigación con un orden de prioridades invertido al usual en el Departamento. Mientras que las "proposals" del contrato con DARPA denominado "BASIC", seguían este orden:

1. Artificial Intelligence. 2. Programming Systems. y 3. Computer Systems, el orden del modelo de "STRATEGIC" era el contrario: "The research will cover parallel computer architectures, associate software, and the application of parallel processing to a set of tasks in the area of artificial intelligence".

(1986, 1-1).

Las áreas de Inteligencia Artificial elegidas para el uso de estos sistemas eran las de Speech de Raj Reddy y Vision de Takeo Kanade, las más formalizadas e ingenieras de A.I. y cuyos directores, Reddy y Kanade, lo eran y lo son a su vez del Robotics Institute de CMU.

Debido a este proyecto, esta investigación denominada "strategic" en Computer Science se diferenciaba de la tradicional "basic research". En 1985, el programa "SPEECH", dirigido por Raj Reddy, se había separado a su vez del BASIC. Los dos líderes principales de este nuevo programa eran H.T.Kung y Rich Rashid. Y sus sistemas: WARP, era un computer system, y MACH, un sistema operativo: "The largest tasks that we propose continue our development of the Warp architecture and the Mach multiprocessor operating system, both of which we began under the current contract. These tasks, in turn, reflect broader effort driven by the

needs in application areas such as speech and vision....The long-range goal of our proposed architecture and operating system research is to support real-time systems for both speech and vision".(1986,1-3).

La principal aplicación de WARP fue en la área de Visión. Uno de los objetivos del Strategic Computing Program fue el desarrollo de un vehículo de tierra autónomo, "an autonomous land vehicle". El equipo de Takeo Kanade de CMU inició la construcción de este vehículo que ha acabado llamándose "Navlab". Este es una furgoneta capaz de desplazarse sin conductor gracias a un sistema de visión que le permite comprender lo que el sistema detecta mediante cámaras de video. Este vehículo tiene un ordenador WARP instalado en su interior que al disponer de una potencia de 100 millones de operaciones (megaflops) por segundo efectúa los cálculos que requiere ese sistema.

Una vez probado que el sistema funcionaba, la investigación en Parallel Computing en CMU dio un nuevo paso: el proyecto NECTAR. Se pasaba del ordenador en paralelo individual a la red de ordenadores. Y de la red de ordenadores con velocidades de millones de operaciones por segundo o megaflops, a la de miles de millones o gigaflops. De Nectar a pasaba a Gigabit Nectar.

4.3.4. DISEÑANDO "GIGABIT NECTAR TESTBED": UNA RED PARA RESOLVER PROBLEMAS INGENIEROS.

La red Nectar, operacional desde Noviembre de 1988, ha demostrado que el objetivo de H.T. Kung es viable. Parece posible conectar entre si ordenadores de muy distinta naturaleza y propósitos trabajando en una misma tarea.

A partir de la tecnología VLSI, Kung ha conseguido diseñar un nuevo tipo de ordenadores en paralelo. El siguiente paso, a finales de los 80s, ha sido ponerlos en red. Ha seguido el modelo de los llamados "sistemas distribuidos".

Los sistemas distribuidos o "distributed systems" son redes de ordenadores separados geográficamente. Están computestas por máquinas autónomas controladas por usuarios individuales para tareas particulares pero que pueden comunicarse entre sí a través de protocolos comunes.

En 1968, la Advanced Research Projects Agency del Department of Defense construyó la primera red informática: ARPANET. Su finalidad era "to permit persons and programs at one research center to access data and use interactively programs that exists and run in other computers of the network". (1981, Heart, Kahn:402). En suma, permitir a determinadas comunidades el acceso a recursos

informáticos dispersos por el país de los que no disponían en su centro.

Al principio conectó a 10 centros de investigación sólo en Estados Unidos. 20 años después, su heredera, la red Internet es la principal red informática internacional. Opera en 26 países, comprende más de 5.000 subredes, y conecta más de 300.000 ordenadores. (Cerf, Vinton, 1991). En Estados Unidos se calcula que la usan 2 millones de personas. (NYT, 13 Mayo 1990), en particular, la comunidad de investigadores del país. El uso principal no es el inicialmente pensado para ARPANET, la comunicación hombre-máquina, sino la de hombre-máquina-hombre, en forma de correo electrónico o de comunicación de información entre investigadores a través de la red.

El promotor de ARPANET fue Robert Kahn, un electrical engineer del M.I.T., que posteriormente pasó a dirigir la Information Processing Techniques Office de la agencia, donde en 1983 inició el Strategic Computing Program.

Hoy Robert E. Kahn preside una organización denominada Corporation for National Research Initiatives, dedicada a fomentar esta tecnología.

A partir de los años 80s, con el desarrollo de las redes electrónicas se han elaborado planes en Estados Unidos para construir una "national information infrastructure". En 1987, el Federal Coordinating Council on Science, Engineering and Technology (FCCSET) realizó un estudio sobre las redes que elevó a la Casa Blanca. En las recomendaciones finales proponía la necesidad de una "national research network", así como un amplio programa sobre aplicaciones de superordenadores. El establecimiento de esta red se realizaría en tres fases: 1. Fase 1 a cubrir en la década de los 80s, la interconexión de las redes existentes y situándolas a una velocidad de uno y medio megabits por segundo. 2. Fase 2. En los 90s, aumentar la velocidad a 45 megabits. y Fase 3. En torno al cambio de siglo, establecer redes de 1 gigabit por segundo en base a fibras ópticas. Ello equivale a enviar el contenido de una enciclopedia cada segundo. Este tipo de red es la que se está construyendo en CMU. (R.E. Kahn, 1988).

Para el desarrollo de proyectos pilotos de gigabit networks, la Corporation for National Research Initiatives (CNRI) recibió en 1990 15,8 millones de dólares de DARPA y de la National Science Foundation. Gigabit Nectar Testbed de H.T. Kung es uno de estos experimentos.

En 1989, H.T.Kung, Eric Cooper y Michael Levine presentaron una propuesta a la CNRI para realizar durante el periodo 1990-93 "a broad program of research in high-speed networks". El objetivo de este proyecto: "We will develop a one

gigabit per second (Gb/s) or higher speed experimental network, called the gigabit Nectar testbed, which will connect a variety of high-performance hosts including the PSC's CRAY Y-MP and the CMU's iWARP parallel machine." (Kung, Cooper, Levine .1989:1).

Este proyecto se apoya en la propia red Nectar que funciona a 100 Megabits por segundo. Y pretende ser un producto comercial.

Esta red pretende ser un sistema completo. Para el software están utilizando el sistema operativo MACH.

Pero es en las aplicaciones donde podemos encontrar el interés de los investigadores en esta red . Según su "project manager", Karl Love, dos aplicaciones fundamentales se estaban desarrollando en 1991: 1. una simulación del diseño , control y funcionamiento económico de una gran planta química, a cargo del profesor Gregory McRae, del Carnegie Institute of Technology, y 2. cálculos combinatorios para resolver el denominado "traveling salesman problem", a cargo del profesor Gerald Thompson, de la Graduate School of Industrial Administration.

Para Gregory McRae, se trata de aprovechar la posibilidad que permite la red Nectar para que varios ordenadores de distinto tipo colaboren en el control de distintos aspectos de dicha refinería. Según McRae ello permitiría una mejora del 10% de la productividad de la planta, con el ahorro de millones de dólares. Por su parte, el problema del "traveling salesman" consiste en determinar el camino más corto a recorrer entre varias ciudades a visitar. Las aplicaciones de este en apariencia sencillo problema pueden ser desde el diseño de conexiones eléctricas entre cientos de miles de transistores de un chip, hasta determinar el desplazamiento de miles de soldados en una operación militar como la Guerra del Golfo.

Pero para el equipo de H.T.Kung y su equipo , estas aplicaciones tienen un significado más profundo: "These problems represent an important class of optimization problems whose efficient solutions are fundamental to many engineering designs" (1989:20).

Se trata no de cualquier tipo de problemas sino de problemas ingenieros clásicos. " The effective synthesis and operation of engineering systems often involves making a number of discrete (combinatorial) decisions. A common feature of these decisions is the goal of choosing the best of a large number of alternatives." (Ibid.:20).

Un rasgo característico de los problemas ingenieros es que ofrecen una amplia gama de alternativas para la solución de un problema, de donde se ha de elegir aquella que permita cumplir mejor los requerimientos que ha de cumplir el sistema, son problemas de optimización. Igualmente, los problemas de management, derivados de la ingeniería, se plantean a su vez la toma de decisión ante una gama enorme de posibilidades para la solución de un problema.

Es en ese marco cultural de problemas ingenieros y de administración de empresas, es decir, de problemas operativos y gerenciales, donde nacería el concepto de "search" como búsqueda de alternativas. Dentro de ellas, la búsqueda algorítmica trata de identificar la mejor de ellas, "the best". Por contra, la búsqueda heurística, se limitaría a encontrar una lo suficientemente buena. Esta aplicación de la matemática a problemas ingenieros y gerenciales recibiría un impulso durante la II Guerra Mundial. Disciplinas como la "operations research", la investigación operativa, de R. L. Ackoff y E. L. Arnoff, la teoría de los juegos de Von Neumann, o la "heuristic search" de H. Simon, estuvieron iniciadas por matemáticos o científicos sociales con formación matemática, que se pusieron a trabajar en problemas ingenieros o de administración.

El enfoque del equipo de H. T. Kung profundiza este enfoque y trata de probar que con la ayuda de las redes de ordenadores heterogéneos ese tipo de problemas ingenieros pueden encontrar la mejor solución.

Una de las características de esos problemas ingenieros es que por su carácter de cálculo combinatorio plantean los más difíciles problemas a la computer science: entre ellos, los llamados "NP-complete problems" o problemas denominados intratables o no computables

Con este tipo de problemas nos introducimos en la área llamada "computational complexity", en la que Kung se empezó a formar dirigido por Joseph Traub. Esta área teórica pretende determinar qué tipo de problemas son computables y cuáles no, estudiando los límites de la computabilidad.

En relación con la aplicación del "traveling salesman problem" (TSP) hasta ahora la investigación realizada indica que este problema forma parte de los "NP-complete problems" y que no existe un único algoritmo para su solución, por lo que la búsqueda a realizar ha de ser heurística, no exhaustiva ni destinada a formular la mejor solución sino una tan sólo lo suficientemente buena ("good enough") según la teoría de racionalidad limitada de Herb Simon.

Sin embargo, la apuesta de este equipo pretende probar que si existe al menos un

algoritmo para resolver la mejor solución entre al menos 1 millón de puntos:

"This application effort exists on the belief that computationally effective exact algorithms may be developed for closely related instances of combinatorial problems arising in engineering applications...Exact algorithms are essential to guaranteeing performance in economically important applications...We believe that combinatorial problems may be solved through a combination of algorithmic sophistication and raw computer power. Our main goal is to develop and implement branch and bound paradigms for exploiting distributed heterogeneous computing resources in the exact solution of NP-complete combinatorial optimization problems". (ib:21).

Este proyecto pretende pues significar un avance en el diseño de algoritmos para resolver este problema y demostrando a su vez la utilidad de las redes de miles de millones bits por segundo.

Sus aplicaciones más evidentes son para la industria del transporte, que mueve miles de vehículos para distribuir mercancías a miles de puntos de venta. Pero otra aplicación menos evidente pero de más alcance es la industria de semiconductores y de electrónica de consumo, "computer and electronic design and manufacturing problems".(ib: 26).

La elección de 1 millón de puntos no es aleatoria. Ya hemos dicho que una de las aplicaciones de este problema es el diseño de circuitos integrados a muy gran escala. Y encontrar la conexión óptima de 1 millón de transistores en dicho circuitos es un objetivo con claras implicaciones tecnológicas. La siguiente generación de televisiones digitales de alta definición depende de un VLSI chip más económico y con más alta integración. Una de las formas de conseguirlo es optimizar las conexiones entre los mismos.

La segunda aplicación de la red Gigabit Nectar va dirigida a la industria química, "to the chemical engineering community". Al aumentar la capacidad de los ordenadores, es posible ahora simular dinámicamente los procesos como el refinado de petróleo. El flujo de estos productos químicos a través de una planta, se denomina "flowsheets" y su simulación implica cientos de miles de ecuaciones. Se trata de mejorar el funcionamiento de dicha planta, "improved chemical plant operation".

El procedimiento es el siguiente: "In practice the engineer would start with a particular design or optimization problem and then develop the symbolic form of the equations that describe the processes occurring inside the plant. The

equations, $f(x)=0$, would then be compiled into C code and matched with a solution procedure appropriate to the architectures available on the network. By exploiting the power of particular computers it should be possible to dramatically lower the elapsed time needed for the engineer to solve the original design or optimization problem. In short, if the constraints imposed by the time for a single flowsheet solution can be eliminated then it is feasible to consider optimizing the performance of realistic, large-scale, chemical plants." (Ib.:29).

La capacidad de la red Nectar se utiliza en este caso para ayudar al ingeniero a realizar los cálculos que requiere ese proceso de simulación.

La utilidad de esta aplicación es extensible al conjunto de plantas químicas del país, dado que pequeños aumentos en la eficacia de su funcionamiento pueden traducirse en grandes ahorros económicos.

En suma, la red Gigabit Nectar, a diferencia de otras redes, como Andrew, van dirigidas expresamente a la comunidad ingeniera.

No obstante, Karl Love, en una entrevista personal, se quejaba del escaso interés demostrado por la propia comunidad de computer scientists en esta red. Entre las razones arguidas planteaba la dificultad de programar los ordenadores en paralelo: "Programming is a real problem"(12-6-91). Su impresión era que el profesor de CMU busca utilizar la red para resolver sus problemas y colabora con Nectar siempre y cuando no le cree más problemas. Por otra parte, el profesor de computer science trabaja individualmente, es "individualist" trabajando en pequeños proyectos aislados más que en uno grande como Nectar. El resultado es que H.T.Kung , según Karl Love, no encuentra con quién interactuar. "He is alone". Un problema adicional es que los investigadores que trabajan con Kung, los "research scientists", como P. Steekiste, Brian Zill, Steve Schlick y otros no son reconocidos como faculty y se van a otra universidad, con lo que el equipo se debilita.

Por otra parte, Karl Love y H.T. Kung intentan convencer a las grandes empresas productoras de ordenadores, en particular, las de estaciones de trabajo ingenieras como IBM, DEC, Hewlett Packart y SUN Microsystem, de las ventajas de una red de estas características, no con demasiado éxito.

En cualquier caso, Gigabit Nectar Testbed está en marcha y en 1993 se podrán valorar sus resultados.

Uno de los indicios de la importancia del trabajo iniciado por H.T. Kung es su

reciente nombramiento en 1992 por la Harvard University como University Professor de "Electrical Engineering and Computer Science". Si tenemos en cuenta que Harvard ha sido el patrón cultural que ha inspirado líderes de CMU como Herbert Simon a la conversión de lo que era un instituto tecnológico en una research university, no deja de ser paradójico que ahora la research university tome a un matemático convertido a ingeniero como profesor de su institución, dando un sesgo ingeniero, no científico, a su investigación en este campo. Este nombramiento continúa una evolución seguida por las universidades norteamericanas desde que el primer instituto de tecnología, el MIT, se construyera independiente de Harvard en 1861: el seguir los pasos de la ingeniería, el asumir su cultura orientada al "real world", su mentalidad de "problem solving", y su tipo de investigación basada en el diseño.

4.3.5. LA COMUNIDAD CIENTIFICA Y LAS REDES ELECTRONICAS: EL "HIGH PERFORMANCE COMPUTING & COMMUNICATIONS PROGRAM"

En el tema de las redes electrónicas es una de las áreas donde más claramente se puede ver la relación entre ciencia y tecnología en Estados Unidos, y en concreto en CMU.

Las redes electrónicas forman la más avanzada expresión de un "artificial world", de un mundo artificial diseñado integralmente por el ser humano. Es una empresa tecnológica, como en los 60s, lo fueran los viajes espaciales, que encontraron en un principio fuertes resistencias entre la comunidad científica.

En 1969, la Advanced Research Project Agency del Defense Department inició la primera red, a iniciativa de un grupo de ingenieros dirigido por R. Kahn. No fue hasta 1986, 17 años después, que la National Science Foundation (NSF) decidieran establecer 6 centros de supercomputación conectados entre sí: la denominada NSF-NET. Su objetivo: poner un supercomputador virtual en la mesa de cada investigador. Un año después, esta red estaba saturada.

En 1987, la NSF entró en un acuerdo con la empresa Merit, Inc. para rediseñar la red, en colaboración con IBM y MCI Telecommunications Co. Una vez rediseñada, empezó a aumentar de nuevo su tráfico. En 1989, conectaba a más de 800 redes estatales, regionales e institucionales, y alrededor de 100.000 computadores en Estados Unidos. En comparación, a principio de los 90s, Internet, la heredera de ARPANET, conectaba ya a 26 países con 5000 redes y 300.000 ordenadores conectados. (Scientific AMe:80).

Ahora ambas organizaciones NSF y DARPA, junto con otras agencias federales, se han unido en 1992 para colaborar en un programa denominado "Grand Challenges: High Performance Computing and Communication" (HPCC) elaborado por tres organizaciones, la Office of Science and Technology Policy, el Federal Coordinating Council for Science, Engineering and Technology y el Committee on Physical, Mathematical and Engineering Sciences.

Este programa está compuesto por un destacado grupo de representantes de estas organizaciones, y presidido por Charles Herzfeld, del Departamento de Defensa. Los objetivos de este programa son un compromiso entre ambas culturas, la más ingeniera y la más científica. Por un lado se afirma que : "The goal of the Federal HPCC Program is to accelerate significantly the commercial availability and utilization of the next generation of high performance computers and networks" (Office of Science and Technology Policy, 1992.:2).

Por otro se dice que su objetivo es permitir la investigación en los grandes retos, "grands challenges" críticos para las necesidades del país. Como ejemplos se citan: " prediction of the weather, climate, and global change; determination of molecular, atomic, and nuclear structure; understanding turbulence, pollution dispersion, and combustion systems; mapping the human genome and understanding the structure of biological macromolecules; improving research and education communications; understanding the nature of new materials; and problems applicable to national security needs". (ib:2).

En suma, la agenda de problemas propia de la comunidad científica, empezando por los físicos y otros científicos tradicionales

Para ello se estructura un programa en cuatro partes: que trata de desarrollar "High Performance Computer Systems" (HPCS), "Advanced Software Technology and Algorithms" (ASTA), una "National Research and Education Network" (NREN) y por último un programa de investigación básica y de formación de recursos humanos, el "Basic Research and Human Resources".(BRHR).

Según Duane Adams, de CMU, éste es el mayor programa informático hoy en Estados Unidos e indica la estrategia que se sigue para el futuro por estas grandes agencias.

Si ésto es así dos lecciones podemos apuntar, extraídas de la experiencia de la red Andrew, la primera "research and educational network" en un campus americano.

1. Una red electrónica para resolver problemas clásicos de la ciencia o las

humanidades no significa ningún cambio substancial en el sistema de conocimiento. No implica una "revolucion del conocimiento". El sistema de conocimiento se mantiene incambiable por más que los artículos científicos se escriban ahora en un procesador de textos. La ciencia más un ordenador significa la puesta de un nuevo instrumento tecnológico al servicio de la clásica tarea de descubrimiento. Las redes pueden ser de varios tipos. Unas como la de Mead-Conway sirvió para diseñar VLSI, otras sirven para comunicarse descubrimientos. No son lo mismo redes de información que "redes de diseño", como demostró el experimento Carver-Conway.

Una informática al servicio de la "educación" en general no cambia tampoco, como hemos visto en CMU, la estructura básica de esta tarea. Se sigue enculturar al estudiante en el patrón clásico del sistema de conocimiento presidido por la ciencia, sólo que con la ayuda de una herramienta más sofisticada. Podrían diseñarse redes para otro sistema educativo basado en la ingeniería y el diseño, y que ayudaran a formar a los 100.000 profesionales de software que faltan en U.S.. Pero estas serían un tipo de redes distintas para un tipo de educación distinta. El problema no es pues la "education" sino el tipo de educación.

2. Este tipo de redes de investigación o educativas abarcan a una ínfima parte de la población no modificando la estructura social de un país. Su impacto comercial, por lo tanto, es mínimo. Como señala Stephen S.Wolff, de la NSF: "Nobody knows how to make the transition from this nice genteel scholarly thing to a general-purpose network. Commercial interests will not want to fund a network just for scholars. They make money on millions of subscribers" (NSFNET, 1990: 20).

3. Este tipo de redes no afecta a la calidad del trabajo investigador ni a la competitividad industrial de forma directa. Japón, un país más atrasado que Estados Unidos en redes informáticas, y con menos conexiones internacionales que las existentes entre U.S. y Europa en materia de redes, no ha tenido obstáculos para conseguir en la pasada década crear una comunidad de investigadores que en 1989 ha sobrepasado proporcionalmente a la de Estados Unidos . Igualmente, Japón desde 1985 ha empezado, junto con Alemania, a sobrepasar a Estados Unidos en el porcentaje de R&D respecto al producto nacional bruto. Y su productividad industrial ha aumentado en relación con la norteamericana. (National Science Board, 1989).

Esta no concordancia entre la extensión de las redes electrónicas con la

productividad industrial de Estados Unidos, es un síntoma de la escisión irreversible entre lo que se denomina "information society" y lo que Hajime Karatsu empieza a llamar "a technological country" (1986)..

El Gigabit Nectar Testbed es un proyecto de una estrategia aún basada en la idea de "information society" según la cual, el diseñar redes de miles de millones de bits por segundo, conectar a ella a toda la comunidad de investigadores del país y ponerlos a trabajar en grandes "challenges" va a mantener el liderazgo tecnológico de Estados Unidos.

Pero este modelo ha hecho crisis. Y el aumento de bits por segundo no va a ser suficiente para resolverla. Otros enfoques se están planteando en CMU para ello.

4.4. EL PROYECTO "NEW CONCEPTS FOR ADVANCED HIGH DEFINITION SYSTEMS".

4.4.1. CRISIS DE LA "INFORMATION SOCIETY" EN ESTADOS UNIDOS: EL INFORME DEL MASSACHUSETTS INSTITUTE OF TECHNOLOGY.

En 1989 la Comisión on Industrial Productivity del Massachusetts Institute of Technology publicó un informe denominado "Made in America: Regaining the Productive Edge" (Dertouzos et. al, 1989). En él no se partía del concepto de la "Information Society", sino del estado de la industria norteamericana. El motivo era claro: "the fears of economic decline...linked to larger doubts about the nation's ability to retain its influence and standing in the world at large". (1989:2).

Se analizaron ocho sectores industriales tradicionales y también alguno perteneciente a lo que se denominan "high tech industries": automóviles, química, aviación comercial, electrónica de consumo, máquinas herramientas, semiconductores, ordenadores y equipamiento de oficina, siderurgia y textiles. En cada uno de ellos, se analizaron parámetros como la eficiencia, la calidad del producto, la innovación, la adaptabilidad y otros aspectos. 8 equipos de profesores del M.I.T., de una decena de media cada uno, visitaron más de 200 empresas en Estados Unidos, Europa y Japón, entrevistaron a 550 expertos durante tres años.

Finalmente emitieron un veredicto, a modo de jueces : "The verdict is that American industry indeed shows worrisome signs of weakness. In many important sectors of the economy, U.S. firms are losing ground to their competitors abroad". (1989:8).

El sector donde se había producido un cambio más significativo era el de los semiconductores y ordenadores, el motivo inicial del estudio. En el primer

sector, Estados Unidos había bajado en menos de una década del 60% al 40% de la producción mundial. En el sector de ordenadores, la balanza comercial, aún siendo positiva, había bajado de 7 mil millones de dólares en 1981 a 3 mil millones en 1987.

Ralph E. Gomory, antiguo Senior Vicepresident for Science and Technology de IBM, y hoy presidente de la Sloan Foundation, ante estas pérdidas exclamaba: "How could this be happening to the greatest scientific power in the world -the home of the most Nobel laureates and innumerable scientific breakthroughs?" (Gomory, 1989:99). Concluyendo: "In fact, the United States is learning only now the hard lesson it taught the rest of the world earlier this century: Product leadership can be built without scientific leadership if companies excel at design and the management of production. The United States was the leading industrial power well before it became the leading scientific power. When, during the 1920s, the capitals of science were the European universities, the United States excelled in worker productivity and per capita income and had the biggest trade surplus...Now U.S. universities are the capitals of science, and Japan has the trade surplus". (Ib.:99).

En efecto, las ganancias en la industria de los semiconductores habían sido aprovechadas principalmente por Japón, país que en esta área había pasado en una década a controlar del 28% al 50% del mercado internacional .

La estrategia a largo plazo de este país es arrebatar el liderazgo mundial a Estados Unidos en el campo de las denominadas "information industries". Según un estudio del Ministerio de Industria y Comercio Internacional japonés denominado "Vision for the Year 2000", la "information industry" producirá el 21 % de todo el producto nacional bruto de ese país a final de siglo. Este sector incluirá los semiconductores, la electrónica de consumo, las bases de datos, el equipamiento de informática y telecomunicación y los servicios que implican. Según el citado informe del MIT: " The goal of achieving economic primacy in the worldwide information industry seems logical and strategically sound for a country like Japan, as it requires little in the way of raw materials and depends primarily on a good supply of intelligent, well educated people". (1989:269).

Pero el problema es más profundo. La estrategia citada está siendo respaldada por una elaboración teórica más de fondo de profesores japoneses sobre las relaciones entre ciencia y tecnología en la presente etapa de las sociedades

industriales avanzadas.

En 1986, Hajime Karatsu, director del International Technological Strategy Research Group del gobierno japonés, escribía un libro titulado "Tough Words for American Industry"(Karatsu, 1986). En él, planteaba una estrategia a largo plazo dirigida a establecer a Japón como un "Technological Country". (1986:111). Esta estrategia partía de dos premisas teóricas: 1. "Distinguishing Technology and Science" y 2. "Understanding Technology".

Para este experto en política tecnológica, ingeniero de Matsushita y profesor de la Tokai University : " Science and technology inhabit essentially different realms and have different goals. People often speak of technology as flowering on a bed prepared by science. This implies that the two are actually the same thing-and from there a number of misconceptions have arisen". (1986:116).

Para este ingeniero japonés, la visión occidental de la tecnología como mera idea conduce a no entender su sentido más profundo : "No matter how wonderful an idea might be, it cannot be classed as 'technology' until it has been converted into a useful product...Technology is a method used for the purpose of making things ; if you cannot make anything, you don't have it.". (ib: 117)

Tres años más tarde, el informe del M.I.T. se hacía eco de las advertencias que desde los años 80s líderes japoneses innovadores como el propio Karatsu o Akio Morita de SONY han venido realizando sobre el peligro del vaciamiento de la industria norteamericana ("the hollowing of American Industry, op.cit. :37). En lugar de partir de la ya tradicional visión de la "information society", liderada por Estados Unidos y su base económica "postindustrial" basada en los servicios, respaldada en la creación de conocimiento en sus universidades, se partía del atraso tecnológico y industrial norteamericano frente a sus competidores japoneses y alemanes.

Se constataba, por ejemplo, que la falta de "education" de los jóvenes norteamericanos era más aguda en el sector tecnológico. El informe señalaba la falta de "education for technological competence" (Ib:134). Solo el 6% de estudiantes de bachillerato cursa carreras de ingeniería frente al 20% en Japón y 37% en Alemania. "Today and in the future, effective use of new technology will require people to develop their capabilities for planning, judgment, collaboration, and the analysis of complex systems". (:135). La opinión pública norteamericana está muy sensibilizada por el tema de la "education", pero el enfoque que los expertos dan en el tema está asociado en primer lugar al

aprendizaje de matemáticas y ciencias. El enfoque del M.I.T. tenía un enfoque distintos, orientado hacia un educación tecnológica basada fundamentalmente en conocimientos como "planning, judgment, collaboration, and the analysis o complex systems".

El mensaje del M.I.T. se dirige en explícito a relativizar el papel preeminente que la ciencia ha jugado en el país desde la II Guerra Mundial: "U.S. policy for science and technology has traditionally focused on basic research and paid much less attention to the commercial development and application of new technologies...An American system of engineering education (that) has progressively deemphasized product realization and process and production engineering since World War II. The retreat from these activities has its origins partly in the wartime experience itself and especially in the dominant role played by physicists and other natural scientists in the development of nuclear weapons...". (ib:77-78).

"As technology became increasingly based on science, the engineering curriculum evolved in the same direction...Excelence in engineering science became the principal criterion for faculty tenure and promotion. At the same time the design of manufacturing processes and production operations acquired a reputation as lowbrow activities and largely disappered from the curriculum...Indeed, even product design has been deemphasized. Engineering students are taught to analyze systems but no really to design them"(:79).

Pero no sólo los ingenieros se han visto absorbidos por la cultura científica, lo mismo ha ocurrido con los managers, según el informe citado. Las empresas japonesas no las dirigen personas con "masters" en "business administration" sino ingenieros con experiencia en administración industrial, como las antiguas compañías norteamericanas. El management, una disciplina inventada por un ingeniero, Frederik Taylor, ha pasado , tras la II Guerra Mundial, a estar dirigida por científicos de la administración y economistas. Finalmente, el empresario norteamericano ha acabado preocupándose por los resultados financieros más que por el producto que vendía. El resultado final ha sido la adopción por la industria del modelo del mercado de los valores bursátiles, acabado en el crack de 1987. "American managers focus too heavily on meeting short-term financial objectives, ...they value too lightly knowledge of production and markets, and ...they pay too litle attention to foreign markets". (Ib:43).

Según las teorías de D. Bell sobre la "sociedad postindustrial", ésta se iba a basar en una economía de servicios, dado que la mayor parte de la población asalariada tendía a concentrarse en ese sector, y dentro de él, la ciencia iba a ocupar el papel de liderazgo intelectual.

Pero la "information society" no suprime la necesidad de una economía industrial potente sino que la hace más necesaria aún. A la sociedad industrial no parecer poderle superar una "postindustrial society" basada en la ciencia y los servicios como pronosticaba D. Bell en los años 70s, sino una sociedad supraindustrial con una tecnología más avanzada, con una manufactura más productiva y eficiente, y una mano de obra más cualificada tecnológicamente. El dilema cultural no parece estar, pues, entre "sociedad industrial" y "sociedad postindustrial", sino entre "sociedad industrial" y sociedad superindustrial, resultado de una nueva cultura tecnológica. La sociedad de la información actual se puede denominar, en efecto, una "sociedad del conocimiento", pero no es un nuevo tipo de sociedad o cultura sino una etapa terminal de un modo cultural basada en la ciencia.

Tras ella puede nacer otra "sociedad del conocimiento", pero de un tipo de saber distinto, de un "conocimiento tecnológico", basado en el diseño, principalmente. Existen, pues, dos tipos de "sociedades del conocimiento" distintos, el conocimiento y la cultura científica-industrial y el conocimiento y la cultura tecnológica-social, la denominamos así, porque como hemos ido viendo por los proyectos en CMU, ya no se dirigen sólo a la industria, sino a la industria, a la administración y al conjunto de sectores que forman una comunidad cultural compleja. Lo que permite, en hipótesis, el generar sectores industriales más integrados y productivos que los hasta ahora conocidos.

Ello replantea el problema de la educación de forma nueva. Hasta ahora el gran defecto de la educación básica en Norteamérica se consideraba que era el que los jóvenes no sabían bastante matemáticas y ciencias ("Nation at Risk". 1983). Pero según este modelo del M.I.T., la gran laguna de la que depende el futuro del país es la educación tecnológica. Igual ocurre en la mayor parte de países europeos, con excepción quizá de Alemania. La educación en ciencias es suficiente, pero la educación tecnológica está subdesarrollada, culturalmente infravalorada.

El informe del M.I.T. era el diagnóstico de un instituto tecnológico, con ayuda de economistas. Pero durante la década de los 80s, Estados Unidos se ha regido

por teorías económicas elaboradas por la denominada Escuela de Chicago, dirigida por Milton Friedman. Estas habían nacido a fin de salir de la hiperinflación de gastos estatales burocráticos, propios de Welfare State. Frente a los controles y gastos federales, estos nuevos economistas creían en la iniciativa de empresarios individuales y en la reducción de los impuestos por parte de la Administración. Sostenían que la economía podía avanzar confiando en el control monetario por parte de la Administración. Durante esta época, el país ha vivido tendencias contradictorias.

Por un lado, fue durante esa época y gracias a esta doctrina antiestatista y que sostiene la iniciativa individual que el "ingeniero promotor" del Silicon Valley pudo desarrollar el máximo de su capacidad innovadora, y que la industria de la "high tech" pudo expandirse en Estados Unidos, sirviendo de modelo al mundo. Y esta cultura de innovación ha inspirado a Japón a adoptar su modelo tecnológico actual. Sin embargo, por otra parte, ha sido la época del llamado "money game", o la creación de riqueza basada en compras y ventas de valores bursátiles, y en especulación financiera en el mercado de capitales. Esta tendencia ha favorecido aún más la idea de la "information society", llevada al extremo por los intercambios financieros al margen de avances en la productividad industrial. El hundimiento de Wall Street en octubre de 1987 puso en crisis la doctrina monetarista.

Esta época parece tocar fin.

4.4.2. LA CRISIS DE LA "INFORMATION SOCIETY" EN CMU: LA VUELTA DE LOS INGENIEROS Y EL CASO PARADOJICO DEL DR. JORDAN.

La crisis del modelo de la "information society" se empezó a apreciar claramente en CMU entre 1990 y 1991. Coincide con la retirada de la antigua administración Cyert, y la elección de una nueva dirección presidida por Robert Merahbian. A finales de los 80s, Carnegie Mellon llegaba al cenit como "research university". Conseguía colocarse entre la élite de las 25 primeras universidades nacionales. Había establecido la red Andrew. La School of Computer Science estaba recién estrenada. En toda su historia anterior, esta institución no había llegado tan alto en la jerarquía de las instituciones de "higher education" del país.

J.J. Servan Schreiber publicaba desde ella en 1986 un pequeño libro titulado la "revolución del conocimiento", en que ponía a CMU, y su experimento ANDREW, como

modelo alternativo de los Estados Unidos frente al desafío industrial japonés.

El Dr. Angel Jordan había colaborado desde su cargo de Provost en conseguir ese estatus. El paso lógico era presentarse a la presidencia de CMU, oportunidad que se le ofreció al dimitir R. Cyert en junio de 1990.

El Dr. Jordan había sido durante 7 años (1983-1990) provost de dicha universidad encargado de la investigación. Su carrera la había iniciado como licenciado en Físicas por la Universidad de Zaragoza (1952). En Estados Unidos se pasó a la ingeniería doctorándose en el Electrical Engineering Department de Carnegie Tech en 1959. La mayor parte de su carrera ha discurrido en el Carnegie Institute of Technology. En 1969 llegó a director del departamento que más tarde cambió su nombre por el de Electrical and Computing Engineering Department. Diez años más tarde en 1979, llegó a decano de la escuela de ingeniería y en 1983 fue nombrado provost de CMU.

Su trayectoria ha estado marcada por una preocupación central: CMU . A esta universidad ha dedicado su vida profesional, tratando dentro de ella de favorecer una línea de apertura a la industria, propia de la ingeniería . Para Angel Jordan no bastaba con convertir a CMU en una universidad de excelencia en computer science. Había que conectarla con la industria y con las tendencias del país. Su modelo de inspiración, no era la clásica universidad al estilo Harvard, sino claramente el M.I.T..

Durante décadas ha tratado de dar un carácter aplicado a las investigaciones que se hacían en algunas ramas de esta universidad, en especial al computer science e ingeniería. Potenció con Raj Reddy en 1979 la creación del Robotics Institute, que fue la primera vía de conseguir que la Inteligencia Artificial pasara en CMU a mostrar su utilidad para la industria. Luego fomentó con Nico Habermann y otros profesores, la creación del Software Engineering Institute para dar salida aplicada al software. Fue uno de los miembros fundadores del Pittsburgh High Technology Council, sociedad dedicada a potenciar la high tech industry en la ciudad. Es miembro de varios consejos de administración de empresas nacionales y está conectado con otro número de empresas nacionales e internacionales.

Tras la dimisión del Dr. Cyert, el profesorado de Carnegie Mellon eligió a un ingeniero para presidir CMU, pero éste no fue Angel Jordan , sino el Dr. Robert Mehrabian, hasta entonces decano de la escuela de ingeniería de la Universidad de California en Santa Bárbara, donde había jugado un papel destacado en su promoción.

A su vez, para el cargo de Provost se nombró también a otro ingeniero, el Dr. Paul Cristiano, antiguo decano adjunto del Carnegie Tech cuando Angel Jordan era decano.

Este cambio es culturalmente muy significativo y expresa varias tendencias de fondo: una de ellas, de especial interés, es la reacción de la cultura ingeniera de CMU contra la antigua administración demasiado comprometida con la "computer science community", y en parte con la "liberal arts".

Durante una generación, casi 30 años, la "computer science" ha tenido en esta universidad una posición de hegemonía, desplazando la tenida hasta la II Guerra Mundial por la ingeniería. Pero este modelo está en crisis ahora. Pese a tener la más avanzada investigación del mundo en esta área, y pese a ser la área clave de la "information society", Estados Unidos está perdiendo liderazgo industrial frente a otra potencia, Japón. Esto ha iniciado una reacción ingeniera en el país, como el informe del M.I.T. indica.

Pero ante la ausencia de un modelo tecnológico más complejo, se vuelve a la antigua "engineering culture" que hizo al país económicamente poderoso. Su traducción en CMU ha consistido en elegir una nueva dirección, de carácter ingeniero. A ello ha contribuido también el factor económico que hace que esta universidad deba preocuparse más por la educación de sus estudiantes de licenciatura, no sólo de los graduados, de donde obtiene buena parte de sus fondos económicos.

La paradoja es que no se ha elegido a una dirección de "computer engineers". Angel Jordan representa la paradoja de un ingeniero que ha trabajado aliado a los computer scientists y es visto de una forma contradictoria por ambas comunidades. Las visiones que a cerca de su anterior mandato tiene el campus depende de las escuelas. Para algunos profesores de Humanidades y Bellas Artes, esta figura de CMU no hizo lo bastante por la "liberal education" en el campus, sólo preocupándose del "research". Para un sector minoritario del profesorado, incluyendo parte de la ingeniería, Angel Jordan representa la mala conciencia del dinero obtenido del Departamento de Defensa.

Pero es más interesante analizar qué significa este profesor para las dos escuelas principales de CMU: el Carnegie Institute of Technology y la School of Computer Science. Para el sector ingeniero mayoritario, Angel Jordan ha estado asociado con una área, la computer science, que ha despreciado históricamente a la ingeniería como es propio de la ciencia. Por otra parte, ha sido la área que

ha recibido más dinero de investigación que ninguna, más que la propia ingeniería, a cambio de menos obligaciones en el terreno educativo. Sin embargo, después de 30 años de "computer revolution", la "information society" que ha ayudado a establecer no parece garantizar a Estados Unidos su papel de liderazgo económico en el mundo.

Japón tiene un nivel de computer science más atrasado que el americano pero en ese país el ingeniero o el ingeniero-empresario, como Akio Morita, presidente de SONY, tiene una más alta consideración social que el científico o un premio Nobel.

Por otra parte, Angel Jordan no acaba de encajar en la School of Computer Science. El no es un "computer scientist" sino un "electronic engineer". Y como es sabido la ingeniería es vista en la cultura universitaria como una disciplina de menor rango que la ciencia.

Ello hace la historia de este profesor realmente paradójica. Vino de España en 1956 con un título de científico bajo el brazo, pues su licenciatura fue en Físicas por la Universidad de Zaragoza. Aquí como otros muchos europeos se sintió atraído por lo que tradicionalmente atrajo a los europeos a America, la tecnología, las "oportunidades", el "every-thing-is-possible". Sacó su doctorado en "electronic engineering". Y vino a hacerlo en un momento en que el país iba a iniciar un largo periodo de sobrevaloración de la ciencia como el conocimiento fundamental en la comunidad académica. Este ha sido el periodo de la Computer Science en este campus. Y el Dr. Jordan siendo ingeniero ha favorecido esta dinámica en pro de la ciencia, y en detrimento de la ingeniería, aunque esta ciencia fuera la Computer Science. 30 años después, se empiezan a ver los límites de esta era.

El caso Jordan expresa dos cuestiones interesantes desde el punto de vista del antropólogo: 1. El agotamiento del modelo de "computer scientist" para liderar el campus de CMU, dada su persistente negativa a atender las tendencias hacia la ingeniería que el país exige de forma creciente. y 2. Los límites de la ingeniería sin la tecnología informática para superar al computer scientist en el diseño de una nueva cultura tecnológica. El ingeniero tradicional no puede dirigir al computer scientist.

Una forma de resolver este dilema es reconocer que están naciendo un nuevo tipo de ingenierías, que no son "computer science" ni tampoco las tradicionales ingenierías (civiles, mecánicas...). Estas nuevas ingenierías que podríamos

llamar "high engineering" como el software engineering , el knowledge engineering, el computer engineering, la bioengineering, y otras , forman una primera base de la "high tech culture", y cuyo conocimiento central es el diseño. Este primer estrato de nuevas ingenierías es ya insuficiente para abordar el diseño de sistemas tecnológicos complejos que ya plantean la necesidad de otro nivel de "high technologies" esta vez provenientes de áreas como el "industrial management", el "policy analysis", y otras áreas sociales y culturales, como la sociología de las nuevas tecnologías o la tecnoantropología. De momento, Angel Jordan , se ha situado como profesor al Robotics Institute, que es un centro de la School of Computer Science, pero que históricamente ha guardado excelentes relaciones con la escuela de ingeniería, la de administración industrial y la de administración pública. En este instituto, ha emprendido dos nuevos programas de investigación aplicados: uno sobre high definition systems, y otro sobre "computer industry", que expresan este cruce interdisciplinar de culturas académicas que él intenta representar.

4.4.3. EL PROYECTO "ADVANCED HIGH DEFINITION SYSTEMS" DE CMU.

El primer proyecto se denomina "New Advanced High Definition Systems".(CMU-SCS-Robotics Inst. 1991).

Según Angel Jordán, estamos asistiendo a una progresiva unificación de tres mundos tecnológicos que estaban separados hasta la fecha: el de la tecnología digital del ordenador, el de la tecnología de la imagen y el video, y el de la tecnología del teléfono y las telecomunicaciones. Esos tres mundos pueden confluír en un sistema tecnológico unificado, basado a su vez en una máquina unificada, que cumplirá estas tres funciones y otras nuevas.

Angel Jordan la denomina un "advanced high definition system", y piensa que ya no debe considerarse un ordenador, aunque lleva un sistema de procesamiento de información inteligente en su interior.

El nuevo enfoque adoptado es que este sistema no parte del ordenador sino de la televisión, una tecnología que no se estudiaba en CMU tradicionalmente. La idea es " 'software for Television' not "Television for Software' " (R. Thibadeau).

La estrategia del Dr. Jordan para su construcción ha combinado avances de ingeniería electrónica con otros de computer science, pero partiendo de la dirección de la ingeniería electrónica no de la ciencia. Este sistema se empezó concibiendo como un televisor de alta definición con pantalla plana basada en la tecnología de las matrices activas . A continuación Jordan pensó que ésta

tecnología podía mejorarse utilizando la tecnología digital del ordenador, en la cual CMU era fuerte. Se pensó en adoptar el sistema operativo MACH para acoplarlo en la nueva televisión. Esta tecnología se introdujo en un sistema digital basado en un dispositivo de control inteligente de las imágenes, idea aportada por R. Thibadeau. Finalmente estas imágenes se visualizan en tres dimensiones con la tecnología oportuna, en la que es experto Mel Siegel.

El nuevo ingenio en proceso de diseño se denomina: "Advanced Video Display System (AVDS)" y consiste en un sistema de tres componentes: una plataforma para el desarrollo de aplicaciones de video, o "Video Applications Development Platform", una tecnología de pantallas avanzadas o "Advanced Display Technologies", y unas tecnologías para la construcción de imágenes tridimensionales, o "Three-Dimensional Implementation Technologies". Existe a su vez un cuarto proyecto para el desarrollo de ambientes de aprendizaje para "software engineers", dirigido por Scott Stevens, que ha sido financiado a través del SEI.

Tres equipos distintos situados en dos institutos distintos han iniciado a finales de 1991 la investigación, financiada por DARPA.

El primero lo dirige Robert H. Thibadeau, a Senior Research Scientist y director del Imaging Systems Laboratory del Robotics Institute en la School of Computer Science.

El segundo lo dirige T.P. Brody, en el Electrical and Computer Engineering Department. Peter Brody fue el inventor en los laboratorios de la Westinghouse Electric Co. de Pittsburgh de la tecnología de las matrices activas, dispositivos que forman la base de las pantallas de cristal líquido que se utilizan ya en los ordenadores portátiles.

Junto con él, trabaja Richard D. Warner, director de investigación en el Graphic Arts Technical Foundation (GATF), inventor del Comunicador de Color GATF, un dispositivo de generación de color, así como Raymond J. Prince.

El tercer equipo está dirigido por Melvin W. Siegel, director del Measurement and Control Laboratory en el Robotics Institute.

Angel Jordan es el coordinador del conjunto del proyecto.

El primer equipo pretende construir una plataforma para la fusión de la tecnología del video y del ordenador, conteniendo Inteligencia Artificial: "A Video Applications Development Platform for the fusion of video technology and computers, allowing advanced video display (AVD) systems to access existing

computer software and imaging data and supporting the rapid prototyping of AVD applications".(CMU-SCS-Robotics, 1991: 1).

Esta plataforma incorporara dos dispositivos fundamentales: un sistema de control de imágenes o "Intelligent Frame Buffer" y el sistema operativo MACH, en que trabaja un veterano alumno de Computer Science en CMU, Joseph M . Newcomer.

El segundo equipo está desarrollando tecnologías de manufacturación de las pantallas planas de televisión. Aunque estas pantallas ya se empiezan a utilizar en los ordenadores portátiles y las televisiones de bolsillo los costes de su manufacturación son aún muy elevados. El problema es que las técnicas utilizadas para la manufactura de circuitos integrados no sirven para la de dispositivos de pantallas de cristal líquido, por lo que este equipo se plantea un nuevo enfoque del proceso de manufactura de estos dispositivos microelectrónicos.

El tercer equipo, se plantea explorar la televisión de alta definición tridimensional o "3-D TV". Las televisiones tridimensionales y los ordenadores con gráficos estereoscópicos están ya en el mercado, pero son caros, e incompatibles con las televisiones de imágenes bidimensionales. El propósito de este equipo es otro: "We imagine a future industry of economical, high quality, robust 3-D TV imaging and computer graphics" (ibid.: 24).

Hay rasgos fundamentales de este proyecto que ahora comienza y que lo diferencian de otros proyectos de la School of Computer Science y de Electrical and Computer Engineering.

1. Este no es un proyecto de Computer Systems clásico sino de "electronical engineering" que utiliza tecnología desarrollada en Computer Science para llevar a cabo un nuevo sistema tecnológico. Ha empezado utilizando el sistema operativo Mach y un dispositivo de control inteligente , el "IFB". De la misma forma, mañana puede utilizar los sistemas expertos o los programas de reconocimiento de voz. Pero no es un ordenador, sino un "advanced video display system".

2. Es un proyecto que introduce a computer scientists en la cultura ingeniera, muy ligada a la industria. Uno de los proyectos, entra en directo a desarrollar un nuevo proceso de manufacturación. El Dr. Brody, principal investigador del mismo es a su vez, fundador de la empresa Magnascreen Corporation, que fabrica la tecnología de las matrices activas para la construcción de grandes pantallas para la televisión del alta definición.

3. Como otros proyectos de la School, éste también está financiado por DARPA. Pero responde a una orientación algo distinta en la agencia. Este es un proyecto

dentro de un conjunto de ellos interrelacionados y patrocinados por esta agencia, conjunto inicialmente pensado con fines primeramente industriales y sólo, en segundo lugar, de la defensa.

4.4.4. LA INVESTIGACION EN HDTV Y LAS NUEVAS TENDENCIAS EN DARPA.

Este programa de investigación indica un cambio de largo alcance en las líneas de investigación de CMU y de la propia agencia que ha sido el sostén principal del Computer Science durante 30 años.

Inicialmente, la investigación que financiaba la antigua ARPA era de carácter científico, básico, y con finalidad exclusiva para la defensa nacional. Un grupo de científicos con ideas interesantes sobre la Inteligencia Artificial consiguieron de ella apoyo durante los primeros años. Estos fueron los años 60s y primeros de los 70s. A este dinero se le llamaba "blue-sky money" (David Pahnos), era un dinero para pensar cosas interesantes, "interesting things". El programa sobre "Information Processing" inicial de CMU se desarrolló en esta fase.

Tras la crisis de 1973, y sobre todo, tras el Programa Japonés de la Quinta Generación, DARPA inició un cambio de orientación. Empezó a exigir expresamente que los proyectos tuvieran una doble utilidad, la "national security" y la "national economy". Este fue, desde 1983, el sentido del Strategic Computing Program . La mayor parte de proyectos de la SCS todavía corresponden a esta segunda etapa. Con todo, el objetivo principal seguía siendo defensa, lo cual venía justificado por la escalada militar de los 80s.

Pero en los años 90s la situación ha cambiado radicalmente. El que inició el cambio en DARPA fue Craig Fields en 1989. Ese año, la agencia empezó a financiar programas que tenían prioritariamente una utilidad industrial: los primeros de ellos, diversos proyectos de High Definition Television, así como otros sobre chips de arseniuro de galio (Gazelle Microcircuits) y baterías de litio. Algunos de estos proyectos los llevaba a cabo SEMATECH, un consorcio de empresas de semiconductores norteamericanas. En suma, DARPA empezó a realizar, lo que se denomina "industrial policy" en directo, al estilo del M.I.T.I. japonés, pero solapadamente, pues esta actividad no entra dentro de sus competencias.

El gobierno Bush hizo dimitir a Craig Fields en mayo de 1990 como director de DARPA, presionado por el Chief of Staff de la Casa Blanca, John Sununu, defensor de la línea neoliberal clásica que deja la economía al libre juego del mercado, y no admite ninguna "industrial policy".

Sin embargo, unos meses más tarde, en septiembre, el nuevo Assistant to the President for Science and Technology, el físico Allan Bromley, anunciaba el comienzo de una "technology policy" por primera vez en Estados Unidos. John Sununu dimitía de su cargo. Y la crisis económica se convertía en el tema central de vida política del país, y el tema clave de las elecciones presidenciales de 1992.

El programa de investigación sobre HDTV de Angel Jordan responde a esta nueva situación del país.

Uno de los centros clave de la investigación en Televisión Avanzada en Estados Unidos está en el M.I.T.. En esta institución hay dos centros que realizan investigaciones en este sentido. El primero y más conocido, el Media Lab de Nicolas Negroponte. Están desarrollando la idea de "interactive television" y "ultra-personalized intimate technology", (Brand Steward, 1988). La filosofía de Negroponte es: " We write about what we do, but we don't write unless we've done it". (ibid.: 7).

El otro centro en el M.I.T. es el que dirige William F. Schreiber, el MIT's Advanced Television Research Program. En diciembre de 1988, DARPA concedió 30 millones de dólares a su laboratorio para realizar su investigación en lo que denomina una Open Architecture Receiver (OAR). Este sistema es un nuevo enfoque para el diseño de sistemas de alta definición, que permite que cualquier señal que llegue al receptor sea tratada por dispositivo de procesamiento de señales o "signal-processing module" y enviada a la pantalla. Dado que este módulo es un ordenador, la apuesta es similar a la del proyecto de A. Jordan, permitir a las computer industries norteamericanas competir con las japonesas y europeas en este nuevo mercado.

Pero la estrategia es distinta. Mientras que el proyecto del M.I.T. está pensado en función de las industrias de ordenadores norteamericanas para que se reconviertan a la nueva televisión, el de CMU está pensado para desarrollar una nueva industria de televisión en Norteamérica, aliada con la industria de televisión por cable, y sirviéndose de avances técnicos de la informática. La estrategia de este proyecto de Angel Jordán no pretende construir el ordenador multimedia sino en el televisor con capacidades informáticas,ésto es, interactivo, con memoria y capacidad de procesar información, pero económico y al alcance de todo el mundo, aprovechando que éste aparato ya está ya introducido en la cultura de masas del país.

¿Qué importancia tiene este proyecto?. Hasta hoy se calculan en 50 millones los ordenadores personales en Estados Unidos (Dertouzos, 1991:63), sin embargo, el número de televisores es tres veces más, 160 millones. Ello indica un fenómeno más profundo. El ordenador no ha penetrado, después de 30 años, en la vida diaria de las familias norteamericanas, como lo hizo la luz eléctrica de Edison o el automóvil de Henry Ford. El televisor, sí. Pero Estados Unidos no se ha preocupado de fabricar televisores sino ordenadores. Y la penetración japonesa en el sector llamado "electronica de consumo", iniciada por SONY en los 60s, no ha tenido competidores en este campo. Pero SONY no es una empresa de ordenadores sino de electrónica de consumo. Y es, en este mercado, no en el de ordenadores, donde según algunos expertos, se juega el futuro de las "information industries".

Las compañías más innovadoras del país como Microsoft apuntan en esta dirección. Como indica Rob Glaser de Microsoft: " Our competitors aren't companies like IBM that sell three millions computers a year. They are the Japanese consumer-electronics companies whose volumes are far higher than IBM's." (Moody, F. 1992, NYT:30). Se calcula en existen ya en el mundo 500 millones de televisores y 250 millones de videos domésticos.

Según W. Schreiber: "High-definition sets are going to use a lot of chips, some of them quite sophisticated-not only memory chips but microprocessors and digital signal-processing chips. That's a very big market, potentially bigger than the entire computer industry" (Donahue, 1989:)

Este campo es una oportunidad para Estados Unidos de revitalizar su su capacidad industrial.

En CMU este proyecto permite asociar la ingeniería electrónica con la computer science y con la nueva industria.

4.4.5. EL PROGRAMA SOBRE "COMPUTER INDUSTRY" DE CMU.

Este proyecto ha dejado atrasada una segunda iniciativa que trata de llevar adelante A. Jordan con otros científicos sociales: el Center for the Study of the Computer Industry.

Este fue una consecuencia del Informe del M.I.T. "Made in America".

La tecnología no basta para resolver la situación crítica de la economía del país. La tecnología precisa convertirse en producto industrial, y venderse en el mercado.

Computer Science ha generado una institución con 700 personas en CMU, una de las

unidades académicas en este campo más grandes del mundo, y sin embargo, no tiene una escuela empresarial que se dedique a formar a los creadores de las computer industries. CMU no ha generado un Silicon Valley.

Si comparamos el Computer Science con la Ingeniería tradicional veremos una diferencia sustancial. El científico no crea industrias. Ha sido históricamente el ingeniero el que se ha encargado de ello. Y el caso del Silicon Valley lejos de ser una excepción confirma la regla. El management, empezando por el del ingeniero Frederick Taylor, es un invento de la ingeniería.

Lo que no existe es un management generado por el computer scientist. Estos se interesan poco por la industria.

Un intento en este sentido fue el planteado por Angel Jordan de organizar un centro para interesar a los "computer scientists" en dicho problema. Junto con el profesor Richard Florida, de la School of Urban and Public Affairs, presentó esta propuesta a la Sloan Foundation, la misma que financió el citado informe del M.I.T. Pero esta propuesta no prosperó. La fundación dio la financiación al Center for the Study of the Computer Industry de Stanford, dirigido por Daniel I. Okimoto.

Es lógico que la universidad que ha jugado un papel clave en el desarrollo del Silicon Valley sea la encargada de estudiar la "computer industry". La pregunta es si un estudio sobre la dicho sector va a ayudar a resolver los problemas del mismo.

De hecho la idea para formar un centro para estudiar la computer industry en CMU fue anterior a la de construir un "advanced video display system". Y tiene una metodología distinta. La primera está pensada para analizar la crisis de la IBM y las grandes compañías informáticas, al estilo de los estudios de casos empresariales. La segunda está creada para desarrollar un nuevo sector industrial en el país, al estilo ingeniero.

Ponerlos en sintonía lo que implicaría sería más bien dedicarse a diseñar un sector industrial de "advanced high definition systems", para dar salida a la nueva tecnología que se está diseñando. Pero para ello haría falta contar con un tipo de científico social, de economistas y de administradores con los que aún no se cuenta en CMU.

Una ventaja de este enfoque es que permitiría coger pequeñas empresas o incluso formar nuevas, a modo de experimento cultural, como se hace con un sistema tecnológico como el Mach. Ello permitiría unir el diseño tecnológico con un

diseño de organizaciones más complejo, y no de cualquier tipo de organizaciones, sino de la organización industrial adecuada para competir en el mundo de los "advanced video display systems". Se podría diseñar su sistema de producción, su sistema de marketing, su sistema de información, etc. saliendo de los clásicos estudios descriptivos propios de las ciencias sociales.

Al fin y al cabo el propósito del Informe del M.I.T. y de la Fundación Sloan no es aumentar el conocimiento sobre la crisis de la industria norteamericana sino empezar a resolverla. Y el HDTV es un reto interesante.

Ralph Gomory, presidente de dicha Fundación, ve así el problema: "Americans excel at founding industries based on scientific ideas. We are the ones who translated the fundamentals of solid-state physics into semiconductors, the ones who are turning molecular biology into biotechnology. But our difficulties start when our industries move from the ladder stage, when engineers keep refining scientific advancements until they yield a new product, to the cycle stage, when those products have to go through continuous rounds of incremental improvements". (Deutsch, NYT 18 august. 1991:)

El problema es que las universidades gradúan gente, según Gomory, que no entienden de donde vienen los productos: "Americans have just never found manufacturing to be interesting, and you don't excel at something that doesn't interest you." (ib. 18-8-91) Pero esto hemos visto que en la historia de CMU no es cierto. Ha habido un etapa en la que el ingeniero ha tenido la ciencia como modelo, en la que se ha abandonado el interés por la industria. Ha sido la etapa donde el laboratorio industrial ha tenido la Universidad como ejemplo de institución y donde el conjunto del país tenía la "information society" como el modelo de cultura que sucedería como "postindustrial society" a la sociedad industrial. Cuanto antes se abandonara la industria y más empleo en servicios hubiera antes se llegaba a esa sociedad.

En CMU, esta etapa han sido los últimos 30 años, en los que por conocimiento se ha entendido "information processing", y donde se ha creído que hacer software para la liberal education iba a solucionar la desindustrialización del país.

Ahora se empieza a ver que la universidad, incluso la "computer university", precisa aún más de la industria que antes para desarrollarse, y que hace falta un tipo de estudios de campo que se realicen tanto en un sitio como en otro para entender cuál es el problema y cómo resolverlo.

Estos son los centros de industrias específicas a los que se refiere R. Gomory:

"The universities and businesses have to send people back and forth to each other's worlds. That's where the industry-specific centers come in. They will be a source of knowledge about what is happening in any given industry...They will interact with companies from around the world to give them insights into what matters. They can be like laboratories, where people study industry much as an anthropologist studies villages." (Ibid, 18-8-91).

Pero estos últimos 30 años no sólo han hecho a los ingenieros científicos, han hecho a los científicos ingenieros. Se ha producido un cambio fundamental en la moderna cultura: la parte más intelectual de la misma, la encargada de elaborar el cuerpo teórico de conocimientos, se ha interesado por el diseño de sistemas, empezando por el diseño del ordenador. La matemática se ha hecho diseñadora, la física de estado sólido también, la psicología también, la lingüística,...

Ello ha conducido a elaborar una disciplina con contenidos ingenieros, como la computer science, pero que a diferencia de la ingeniería tradicional, pretende desarrollar un cuerpo propio de teoría, una "theory".

La Computer Science de CMU tiene una cuarta área denominada "Theory", que es clave para entender el sistema de conocimiento de esta comunidad. Y al mismo tiempo, ello servirá para comprender los límites de la ingeniería tradicional, desprovista de esta área, para competir con la computer science.

5. "THEORY" EN LA SCHOOL OF COMPUTER SCIENCE.

5.1. ¿QUIEN FORMA PARTE DE LA "THEORY"?

La área denominada "Theory" en la School of Computer Science está formada en el curso 1991-92 por un grupo de 17 profesores, siendo junto con Computer Systems, la más pequeña de la School. Esta área es la más reciente de las cuatro en Computer Science, iniciándose de forma organizada a principios de los 80s.

Entre los profesores que la componen destacan Stephen Brooks, Merrick Furst, Robert Harper, Ravi Kannan, Daniel Leivant, Gary Miller, Frank Pfenning, John Reynolds, Steven Rudich, Daniel Sleator, Dana Scott y Doug Tygar.

La área de "Theory" está constituida por una actividad de investigación y una actividad docente. Tres son las subáreas en que se divide ésta: "Computability and Complexity", "Logic" y "Algorithms and Analysis".

Esta área no forma parte de la investigación bajo el contrato denominado "Information Processing Research" que tradicionalmente ha tenido CMU con DARPA. Aunque ésta agencia financia también proyectos en esta área, esta investigación

se ha financiado principalmente de fondos provenientes de la National Science Foundation. Proyectos como "Types in Programming", antes analizado, o ERGO, están clasificados por dicha agencia en la área de "Software".

Según Merrick Furst, uno de los fundadores de la área de "Theory" en CMU: "There is not research projects in Theory. One can work in a problem in this area and after 10 years, finally, there may be no results . The 'projects' in Theory are the articles the people write. The projects are people". (Entrevista Personal. 16-4-92).

Merrick Furst es, a su vez, Associate Dean for Graduate Education de la School. Según él, uno de los cambios introducidos en la escuela en la última década son los exámenes y asignaturas que los estudiantes de grado ahora han de pasar en la escuela. Antes, según Furst, se aprendía como los artesanos, mediante la participación en proyectos de investigación, al lado del profesor que lo dirigía. "I've collaborated to introduce the courses, as in the university structure". (ib.).

La área de Theory tiene sólo 10 años en CMU. Fue promovida por el antiguo director del entonces departamento, Nico Habermann, en colaboración con Furst. A pesar de su corta existencia, no obstante, su influencia va siendo creciente en la normalización del Computer Science en esta universidad con el resto de centros del país. Según Furst, CMU era un caso excepcional hasta los años 80s, pues a diferencia de otros departamentos importantes del país, no contaba con un grupo específico dedicado a "Theory".

Preguntado por las razones de esta diferencia, Furst afirma: "In every place, there was a theoretical area in Computer Science, but not in CMU. The reason might be because at the beginning this Department was built by psychologists and philosophers, like Allen Newell and Herbert Simon". (ib.16-4-92).

En efecto, "Theory" es una parte sustancial para la Computer Science. Y sin Computer Science, los ordenadores serían calculadoras electrónicas ultrarrápidas pero sin ninguna pretensión de ser "universal computing machines", como Turing preveyó. ENIAC, el primer ordenador digital inventado por Mauchly y Eckert era aún un ordenador para un propósito definido "special purpose", calcular trayectorias balísticas. La idea de convertir una calculadora en una "universal machine", no sólo se debe a Turing, sino a Von Neumann, y su denominada "general and logical theory of automata" (Von Neumann, 1961:vol V:288-328) según la cual las llamadas "large computing machines" eran comparables con "living organisms".

La "Theory" no es un apéndice en la escuela como podría pensarse desde una visión pragmática estrecha de la misma. Es la área que legitima ante la Academia la "computer science" como ciencia y de donde, se reconoce que salieron los padres fundadores de la misma: entre ellos, Alonzo Church, Alan Turing, y John Von Neumann. (M. Shaw. Great Ideas. abril 1991).

El ordenador, en efecto, es un ingenio como el automóvil o la máquina de vapor, pero a diferencia de éstas, es una máquina con una particularidad en su origen: la iniciativa para su invención provino principalmente de matemáticos y científicos, si bien los ingenieros contribuyeron decisivamente a su construcción. Y ello fue así desde la primera Máquina Analítica del matemático Charles Babbage, la calculadora de Blaise Pascal, hasta el ordenador ACE de Turing. Es cierto que el ENIAC, el primer ordenador digital fue desarrollado en la Moore School of Electrical Engineering de la Universidad de Pennsylvania, pero aún en este caso, la iniciativa del equipo, según la historiadora Nancy Stern, la tuvo el físico J. Mauchly, en colaboración con el ingeniero J. Eckert. (N. Stern. 1981).

Este ingenio comenzó siendo una calculadora mecánica, una herramienta inventada por matemáticos al servicio de sus cálculos aritméticos. Y ha conservado el nombre: "Computer", que en inglés es sinónimo de "calculadora", y "computation", cálculo. El historiador de la tecnología Paul Ceruzzi señala en el significado del término "computer" un cambio interesante. Antes de la II Guerra Mundial, según el diccionario, el término "computer" designaba la persona que efectuaba los cálculos en las empresas. Después de la misma, "computer" pasó a significar máquina calculadora.

Es cierto que en la computer science confluye otra corriente directamente ingeniera proveniente del M.I.T., las máquinas automáticas o automatas, y que culminará en la robótica. De hecho el Robotics Institute de CMU se formó en 1979 independientemente del Department of Computer Science, y sólo con la formación de la escuela pasó a ser integrado como una unidad dentro de la School of Computer Science.

Esta corriente proveniente más de la ingeniería que de las matemáticas, fue la encabezada por el matemático Norbert Wiener y su Cibernética, nacida en el M.I.T. y desarrollará una teoría de la comunicación y control en animales y máquinas, basada en el mecanismo del feed-back o retroalimentación nacido en la

ingeniería de control. El fin de estas máquinas no era el cómputo sino el control de procesos físicos para el diseño de artefactos eléctricos no lineales.

A su vez, Claude Shannon de los Bell Laboratories desarrolló su teoría estadística de la información, salida de la ingeniería de comunicaciones.

Pero esta corriente con base ingeniera no es la que construyó la Computer Science culture en el resto del país, y tampoco en CMU.

El ordenador que inicialmente se concebía como un autómatas, pasó a ser el autómatas hegemónico, y la "Logic Theory Machine", el ordenador más avanzado, consiguiendo el liderazgo en esta comunidad universitaria. Es interesante observar como finalmente , el "computer", la maquina razonadora, se ha convertido en la máquina por excelencia en la comunidad universitaria, no sólo en CMU, sino crecientemente en el conjunto del sistema denominado "Science and Technology".

La ingeniería de comunicaciones, en particular la obra del ingeniero C. Shannon, aportó el principio digital. Físicos e ingenieros electrónicos como Mauchly y Eckert aportaron su tecnología electrónica. Científicos sociales como Newell y Simon, la hicieron capaz de probar teoremas lógicos. Pero todas estas contribuciones no han cambiado su carácter de máquina esencialmente de cómputo, la máquina lógica.

Ciertamente el ordenador se ha transformado en estos 40 años. De simple herramienta al servicio de la ciencia, se ha pasado a considerar un objeto de estudio en sí mismo. Ha pasado de simple medio a fin. Ello ha producido una situación invertida. Ahora, una disciplina, la computer science, " a sort of science, engineering and mathematics" (Mary Shaw), utiliza las matemáticas así como la lógica para desarrollarse a sí misma, invirtiendo la relación inicial según la cual el ordenador era una herramienta del matemático para efectuar sus cálculos.

Ahora la matemática y la lógica son las que han de producir herramientas para el computer scientist. Ello se aprecia en especial en la área de "Theory", área aún a caballo entre la matemática y una teoría propia de la nueva disciplina.

Un tema clave a dilucidar es pues la relación entre matemáticos, lógicos y computer scientists en la SCS, y en particular en la área de la "Theory", pues es en esta área donde la Computer Science define su identidad como disciplina independiente. Dada la complejidad del tema, éste requerirá posteriores estudios.

Una parte clave para entender la "Computer Science", como sistema de conocimiento cultural, es la área de "Theory". Esta área en la School of Computer Science de CMU nos puede ayudar a explicar cuales son los rasgos de este tipo nuevo de ciencia, así como situar algunos de sus límites.

5.2. "COMPUTABILITY AND COMPLEXITY", "LOGIC", "ALGORITHMS AND ANALYSIS" : AREAS

DE "THEORY" EN LA SCS.

El grupo de Theory de la School está formado por dos subgrupos claramente diferenciados: la "Logic Community", dirigida por Dana Scott, "community" que dirige a su vez un Programa de Doctorado en "Pure and Applied Logic", entre los departamentos de Matemáticas, Filosofía y Computer Science, y por otra parte el denominado "Algorithms Group", dirigido por Ravi Kannan y Merrick Furst, y que tiene un programa de doctorado interdisciplinar en "Algorithms, Combinatorics and Optimization", en colaboración con el grupo de Operations Research de la GSIA, y con la área de Matemáticas Discreta del Departamento de Matemáticas. Esta unión de los dos grupos bajo la rúbrica de "Theory" es anómala en América, según Furst. Según este profesor, en Europa cuando se refieren a "Theory" se entiende normalmente Lógica. De hecho Dana Scott, así como S. Brooks, fue profesor entre 1972 y 1981 en la Oxford University. Sin embargo, en Estados Unidos, los grupos de Theory normalmente están más centrados en Algoritmos y Operation Research.

Para Kannan y Furst, la idea de "feasible" es clave para el estudio de los algoritmos en Computer Science, hasta el punto que esta idea está renovando las propias matemáticas. (Entrevista, 21-4-92). Sin embargo, según éstos, esta noción no es importante para la "Logic Community", cuya idea básica es la "correctness" de los programas informáticos.

En cualquier caso, estos forman dos comunidades separadas en la área de Theory en CMU, con reuniones, publicaciones y conexiones internacionales distintas.

La importancia de esta área de "Theory" en CMU, que tiene un 25% del profesorado del conjunto de Computer Science es reciente. A diferencia de otras

Universidades, como Stanford, la Computer Science en esta universidad no se ha regido por un enfoque principalmente matemático o formal. Incluso la matemática, en esta universidad de origen ingeniero, ha tenido un carácter aplicado desde el principio. El Mathematics Department de CMU esta dirigido por Williams O. Williams, y está catalogado entre los 10 primeros del país en matemática

aplicada. Entre sus áreas de investigación incluyen: "Continuum Mechanics", "Differential Equations", "Discrete Mathematics", "Logic", "Numerical Analysis", "Optimization and Operation Research" y "Stochastic Analysis". Es un departamento con 30 profesores, de los cuales cuatro tienen "joint appointment" con la SCS: P. Andrew, Ravindran Kannan, Dana Scott, y R. Statman

5.3. DEL "PRAGMATIC APPROACH" AL "EMPIRICAL APPROACH".

La evolución de la área de Theory es significativa del carácter cultural de la Computer Science de CMU. Igual que Computer Systems, Theory es la área más pequeña de la School. Inicialmente ninguna de las dos se consideraba en la propuesta inicial dirigida a ARPA en 1964. Pero al igual que ésta, con el tiempo ha ido cobrando creciente peso.

Volvamos a un tema que ya introducimos en la primera parte del estudio pero que ahora cobra relevancia desde otro enfoque.

En la propuesta redactada por Newell, Perlis y Schatz de 1964 no aparece siquiera el nombre de Computer Science sino el de "science of information processing".(1964:3). No se contemplaba construir máquinas (hardware). Se planteaba una orientación sólo hacia "software". Y el enfoque "teórico" estaba basado en una orientación que se denominaba "empirical", basada en la construcción de programas que hacían cosas.

La teoría era la teoría del "information processing". Y afirmaba: "We are theoretical, although not so much by a dependency on formal models (such as automata theory), as by trying to formulate the essential nature of information processing." (1964:4).

En esa propuesta se expone una oposición binaria que ha sido una constante a lo largo de estos 25 años: "formal theory" versus "pragmatic approach".

Según sus autores: "There are at least two ways to go about an integration. One can develop a formal theory of information processing systems, including the development of the associated mathematics...In 1937, before any of the technology existed that has made the study of information processing as imperative, Turing formulated the concept of a universal computer... Their virtues lie in in being able to establish extremely general properties of information processing systems within the model; their major difficulty is limited power on relevant problems...". (Newell et al. 1964:12-13).

Pero los fundadores de este centro en Carnegie Tech, apostaron por otro enfoque denominado: "pragmatic approach". Según ellos: "An alternative to developing

formal theories is to incorporate all the important features of the separate strands into a single unified programming structure. This is a much more pragmatic approach than the formal one. As in the processes of basic discovery, its success is measured by the clarity of structure achieved, validated by useful programming systems applicable to a diversity of new problems".(ib: 13).

Al mismo tiempo afirmaban: "We are oriented towards 'software' --towards programs written on digital computers. We are empirical, in that we believe in constructing programs that do things, and in learning about information processing from the difficulties of construction and the behavior of the resulting programs".(ib.:4).

Se puede advertir que a este enfoque se le denomina a la vez "empirical" y "pragmatic". Ya hemos visto que no son lo mismo. La ciencia empírica se basa en el descubrimiento. Es la tecnología la que tiene un enfoque pragmático, a la búsqueda del "building systems".

En realidad han existido, no dos, sino tres enfoques teóricos en computer science de CMU: los denominados "pragmatic", "empirical", y "formal". Estos han sido protagonizados por tipos intelectuales distintos: el ingeniero, el científico experimental y el matemático o lógico,. Estos tipos que podrían estar representados en las figuras de Allen Newell-Alan Perlis , Herbert Simon y Dana Scott . A menudo se identifica en exceso la obra de Newell con la de Simon no permitiendo distinguir que sus enfoques difieren en gran parte, al menos si atendemos a sus resultados. Herbert Simon, es, ante todo, un científico experimental que se sirve de programas informáticos para probar teorías acerca de la mente humana. Allen Newell, al menos en la mayor parte de su vida, así como Alan Perlis, han sido computer scientists que se han servido de la ciencia experimental como la psicología o de las matemáticas para hacer sistemas informáticos. Pero, finalmente, un problema de ciencia empírica ha terminado por concentrar su atención como veremos a continuación.

El enfoque pragmático es el que inicialmente promovió la Computer Science de CMU. Este enfoque no contemplaba la necesidad de una área de Theory separada del Programming o de la Artificial Intelligence. No obstante, los otros enfoques, el de la ciencia experimental ("Search") y el de la ciencia formal ("Logic") han influido esta área en periodos sucesivos, tratando de modificar su inicial impulso de carácter pragmático.

El enfoque pragmático se podría formular a grosso modo de esta forma: no hay

separación alguna entre el pensar y el actuar. La teoría de Computer Science son los programas que se construyen en dicha escuela y que funcionan. SOAR, el sistema construido por Allen Newell, es concebido por él como un sistema inteligente y al mismo tiempo como una teoría acerca de la cognición humana (1990).

Pero como veremos el enfoque pragmático orientado a la construcción de sistemas ahora se ve por Newell como una mera herramienta para un objetivo más importante, una teoría de la cognición humana. En este sentido es muy significativa su obra "Unified Theories of Cognition", que reúnen un conjunto de charlas, las William James Lectures, dadas por este autor en 1987 en Harvard University: "To 'convert' an existing A.I. system to a cognitive theory seems perhaps a rather strange operation. In the event, there turned out to be two activities. The first can only be described as contemplation --contemplate SOAR in the light of everything I knew about human cognition, to see what fits and what doesn't, how mechanisms fare, and what role they play in explaining psychological phenomena. Though SOAR presented itself as a fixed set of mechanisms, there were open to adaptation under the pressure of becoming a theory of human cognition"...The second activity was to attempt to use Soar as a theory of cognition to model and explain in detail specific psychological phenomena and data in a wide range of domains: in short, to exercise Soar." (1990:x)

Lo curioso aquí es que Newell no concibe la teoría de la cognición humana a partir de su propia obra de construcción de Soar, sino a partir de ese sistema informático una vez construido, contemplándolo como objeto empírico, y haciendo una teoría de él. El pragmatismo se ha hecho empirismo. La A.I. se ha puesto finalmente al servicio de la psicología cognitiva, y la "liberal university" dirige a la "professional approach".

Este enfoque que ha evolucionado del "pragmatic approach" al "empirical approach" ha tenido como resultado que finalmente Newell, tras 30 años de vida científica, se defina a sí mismo ahora no como interesado en A.I. principalmente sino en la psicología cognitiva, y defina su preocupación científica central, "my single ultimate scientific question is the nature of human mind" (1991. "Desires and Diversions". 4 dec. SCS).

¿Qué efectos ha tenido esta orientación en Computer Science, en especial, en la área de Programming?. Dejar sin enfoque teórico a la misma. El "information

processing" se mostraba más interesado en "understanding the nature of human mind" que en construir sistemas computacionales.

Este centro se definió en 1964 , ante todo, como un centro de software. Su objetivo era producir programas informáticos. La actividad de "programming" durante años se ha identificado con la actividad del computer scientist. Hasta el punto de confundir "computer science" con programación. Y esta actividad se la consideraba una actividad propia de la ingeniería.

En 1981, en un libro de texto publicado por W. Wulf, Mary Shaw, Paul Hilfinger de CMU y Lawrence Flon, de la University of Southern California, titulado "Fundamental Structure of Computer Science" se afirmaba: "Programming is an engineering activity. Hence, good programming, like other good engineering, is rooted in the careful application of science to practical problems...Programming is the act of writing a program; it is, in effect, the act of building a specialized tool. It is this emphasis on "building something" that leads to our premise". (1981:1). Así habla la ingeniería tradicional, "building systems".

Según este enfoque, programar es encontrar el algoritmo o procedimiento efectivo. Para Alan Perlis: "An algorithm is a finite sequence of elementary tasks, each performable by a computer in a finite amount of time". (1971:5). En suma, un conjunto de operaciones realizables por el ordenador para la solución de un problema.

Con todo en el libro de texto de Mary Shaw y otros profesores citado, se advertía que existía una diferencia fundamental entre la ingeniería tradicional y el "software engineering": "Most engineers must deal with physical limitations of the materials they use- weight and strength, for example. Achieving an acceptable design within these limitations is often the central problem faced by the engineer. Although there are a few limitations of computers such as the amount of memory and processor speed, these limitations do not matter for a large collection of programs. Instead, the limitations encountered in programming are most often related to our own, very human capabilities: The principal limitations on the programs we write are often imposed by our inability to comprehend the design- not by the physical capabilities of computers". (1981:1).

Pero esta comprensión de la actividad de diseño, nacida del "pragmatic approach", no se ha desarrollado teóricamente.

Diseñar, por otra parte, no es una mera actividad práctica. Se trata de

planificar un modelo de comportamiento que ha de seguir la máquina, no de ejecutar ese comportamiento. Una cosa es el "behavior" y otra el "pattern of behaviour". Para Edsger W. Dijkstra, la actividad del "programmer" se podía definir como: "designing a class of computations" (E. Dijkstra. 1971:8).

La antropología, en tanto que disciplina que se encarga de estudiar los "patterns of behavior" de los seres humanos, de sus normas de conducta, puede ayudar a entender esta diferencia. La cultura no es el mero conjunto de comportamientos humanos organizados. Sino, ante todo, las pautas que rigen esos comportamientos.

El programador informático diseña el modelo de comportamiento de un ordenador. Un programador cultural haría algo parecido con un sistema cultural. De hecho es lo que vienen haciendo consejos de ancianos, legisladores, políticos y educadores de todas las épocas y culturas. Alan Perlis lo entendió correctamente: "Educators, generals, dieticians, psychologists, and parents program. Armies, students, and some societies are programmed. An assault on large problems employs a succession of programs, most of which spring into existence on route..." (Foreword, in Abelson& Sussman, 1985:xi). "The Promethean enterprise", llamó este computer scientist a esta actividad de "programming".

Pero el enfoque pragmático, a falta de una teoría de diseño, se vio sin suficiente armas para afrontar los problemas generados por la propia dinámica de la disciplina. La crisis se inició por la "software crisis". Hacían falta programas correctos y complejos. Y para ello se consideró preciso elevar el nivel de abstracción del programador y su rigurosidad en la verificación de los mismos. Y la Computer Science de CMU llamó a los lógicos en su ayuda, y con ellos, un "logic approach".

Se fue poniendo en evidencia que programar no era escribir sólo las instrucciones o código de la máquina. No era un mero "learning by doing", sino una compleja tarea de planificar y diseñar qué instrucciones han de ser escritas. Por el contrario, una improvisada y acelerada actividad programadora, lo que hacía era retrasar y encarecer aún más el "programming", al producir una gran cantidad de errores ("bugs").

A su vez los programas tendían a ser cada vez más complejos. En los años 70s la Computer Science en CMU decidió introducir una área teórica denominada "Complexity", dirigida por el matemático Joseph Traub, a fin de afrontar el tema de la complejidad en el que se había empantanado el área de Programming. En los

80s, se inició la área de "Logic". Ambas áreas teóricas venían en ayuda de problemas prácticos que un enfoque meramente pragmático no había sido capaz de resolver, y que el enfoque "empirical" de la A.I. no entraban en el campo de sus intereses.

5.4."COMPUTABILITY " Y " COMPLEXITY" EN CMU: JOSEPH TRAUB.

Josep Traub fue el primer director del Departamento que se dedicó expresamente a la Theory. Su tema: la Complexity. Nacido en 1932 en Karlsruhe, Alemania, vino a Estados Unidos en 1939. Obtuvo el doctorado en Columbia University en 1959. Ocupó la dirección del Departamento de Computer Science de CMU entre 1971 y 1979, de donde pasó a fundar el departamento de Computer Science de la Columbia University en New York.

El problema de la complejidad venía impuesto por los propios límites de la mente humana para construir sistemas complejos como los programas informáticos. El problema de la "complexity" se suscitó ante la tarea práctica del software engineer y el ingente trabajo que significaba escribir programas con miles de líneas de código, y sin errores. Este problema es una de las razones por las E.W. Dijkstra, uno de los fundadores del denominado "structured programming", y profesor en la Universidad de Texas en Austin, denomina al "computer science" una "radical novelty in our history". Según Dijkstra: " The programmer is in the unique position that his is the only discipline and profession in which such a gigantic ratio, which totally baffles our imagination, has to be bridged by a single technology. He has to be able to think in terms of conceptual hierarchies that are much deeper than a single mind ever needed to face before. Compared to that number of semantic levels, the average mathematical theory is almost flat. By evoking the need of deep conceptual hierarchies the automatic computer confronts us with a radically new intellectual challenge that has no precedent in our history". (1989:1398)

El cómo hacer frente a la complejidad, el cómo poder manejarla, se convirtió en un problema nacional, parte central de la "software crisis", ya estudiada en el capítulo anterior. Para H.T. Kung, discípulo de J. Traub, el tema de la complejidad introduce un paradigma nuevo: " How to manage complexity, how to build complex systems. " (Entrev. 15-3-91).

La área de la complexity, iniciada por problemas prácticos en CMU, ha ido derivando hacia cuestiones más de fondo, de carácter a veces filosófico.

Joseph Traub, realizaba en 1990 un discurso con motivo del 25 aniversario del

Computer Science en CMU, titulado : "What Is Scientifically Knowable?". El tema del mismo era la relación entre "science and computation". Y la pregunta central: "What are the intrinsic impediments to the computational solution of scientific problems?". (1991:490). Se trataba de analizar los límites que imponía el ordenador para el cálculo de problemas científicos.

Esta problemática ya fue iniciada por Turing al reducir el conjunto de números definibles a aquellos computables por el ordenador. Turing introdujo un tipo de funciones no computables, a partir del denominado "halting problem", según el cual, la máquina una vez iniciado el algoritmo para la resolución del problema paso a paso no puede detenerse al dar como resultado otro nuevo programa que no se puede terminar a sí mismo, ni lo contrario.

Para J. Traub una de las consecuencias de este tipo de funciones es que impone límites a los problemas científicos, tratables por medios mecánicos, es decir, mediante el ordenador. Un problema se dice que es "intractable" si no pueden haber suficientes medios informáticos para su solución. En el capítulo de este artículo titulado "Why We Might Expect Difficulties with Supercomputing and with the Foundations of Physics", J. Traub sostiene que la intratabilidad podría ser un impedimento para lo que es cognoscible por la ciencia: "Indeed, I believe there are problems that are strongly intractable, i.e., intractable in all settings" (ib: 498).

Mas allá aún, Traub se aventura a afirmar que no sólo hay problemas matemáticos que son intratables por medios mecánicos, sino que también existen problemas científicos que producen formulaciones matemáticas que a su vez también lo son: "Do there exist scientific problems such that every mathematical formulation is strongly intractable? I believe that there are such scientific problems and that they are unknowable". (Ib: 501).

Las implicaciones de esta teoría son enormes. De un lado conducen a lo que Lyotard, y que Traub cita, llama la "postmodern science", la ciencia concerniente al caos, la teoría de las catástrofes. En suma, al cambio, del significado del término conocimiento ("knowledge") por el de no-conocimiento ("unknown"). (Lyotard, cit.Traub, 1990:489).

Pero estos límites que impone el ordenador a lo que es cognoscible por la ciencia, permite por contra abrir una concepción de lo que significa conocimiento distinta a la de la ciencia: el conocimiento como diseño. Conocer en este nuevo sentido sería diseñar qué nuevos sistemas mecánicos, biológicos o

sociales son posibles y cómo construirlos.

Esta visión del conocimiento como diseño no pretenden la comprensibilidad de todo el universo, sino sólo el diseño de aquella parte del universo tecnológicamente, esto es, culturalmente factible, diseñable por la única especie que de momento conocemos, que definimos como sapiens y que es capaz de inventar cultura, y máquinas, como parte de ellas. Esto puede parecer una monstruosa reducción del campo del conocimiento del ser humano. Pero no lo fue menos la visión de aquellos científicos que recortaron su campo de estudio al mundo de los fenómenos sólo cuantificables matemáticamente, y gracias a ella, se abrió un nuevo modo de conocimiento humano, un nuevo modo cultural, en el que estamos viviendo una gran parte de la humanidad.

Este tema de la Complexity, aún en evolución, parece indicar que la Computer Science no pretender abarcar como las Matemáticas todo el universo de objetos definibles matemáticamente. Es una disciplina más modesta en sus pretensiones. Se limita a la complejidad de los números y símbolos computables. No es una ciencia formal a la búsqueda de orden del universo expresado en números, según el ideal religioso pitagórico. Sus límites parecen estar impuestos por las máquinas, los artefactos, que los seres humanos podemos inventar teórica y prácticamente. Sólo Pitágoras y su secta religiosa, que al parecer no necesitó de máquinas, pudo aspirar a cubrir tal objetivo. Los computer scientists se limitan a aquellas matemáticas que las máquinas pueden computar.

Pero esta distinción entre matemáticos o lógicos y computer scientists aún es confusa.

En 1978, tuvo lugar en Snowbird una reunión organizada por Joseph Traub entonces en CMU, a cerca de la naturaleza del Computer Science. La posición más radical la introdujo Juris Hartmannis de Cornell. Su polémica se centraba en distinguir, la computer science de las ciencias físicas, sosteniendo que "in computer science (or the new species of the sciences of which there surely will be more) we are primarily concerned with that which is possible, with what can exist... I believe that in computer science, even more than in physical sciences, the theory, the conceptualization, the intellectual framework will determine what can be observed and what is worth observing. Most of the things we will study in computer science in the future have yet to be imagined." (Traub, 1981:354).

Pero la computer science no se interesa por todo lo que puede existir, sino por lo que puede ser hecho ("feasible", Furst) por un modelo determinado de máquina

inventada por el ser humano. El problema de la intratabilidad de determinados problemas científicos indica pues una primera separación entre las construcciones mentales matemáticas y la de aquellos problemas tratables por medios mecánicos. Sólo estos últimos son los que interesan a computer science. Más que la ciencia de las posibilidades, la problemática de la complejidad de Traub plantea una ciencia de los límites de dichas posibilidades.

No obstante, la "complexity theory" no ha conseguido proporcionar herramientas teóricas en positivo para afrontar el tema de la complejidad del programming.

Más preocupado en conocer qué es lo que no pueden hacer las máquinas, que lo que se puede hacer con ellas, ha dejado la otra parte del problema sin solucionar.

Una de las vías para intentar resolver esta parte fue recurrir en CMU a otra arma teórica de la matemática: la abstracción.

5.5. "LOGIC": "ABSTRACTION" Y "REASONING ABOUT PROGRAMMING"; DANA SCOTT.

En el libro "Fundamental Structures of Computer Science" (1981) su autores nos explican el papel de la abstracción en esta disciplina: "In order to deal with complex situations we use a powerful technique--abstraction. We choose to ignore the details of complex systems; instead we develop models which reflect only certain important, macroscopic behavioral properties. Newton's laws of motion, for example, are a model of physical reality." (Wulf et al.1981: 3).

Pero dichos autores diferenciaban, el papel de abstracción en la física de su lugar en computer science. En este campo, consideraban que este mecanismo intelectual jugaba un papel aún mayor: " Here the notions of models and abstraction also play a central role, but the situation is also somewhat different. For the chemist and physicist, reality is ; they propose models to describe and predict reality...The programmer, on the other hand, is building an artificial reality. He can also use abstractions to control complexity, but he has the ability , in principle, to make these models exact. That is, he can make the program perform exactly as predicted by the model under all circumstances. Thus in a sense, the notions of modeling and abstraction are even more powerful in programming than in the physical sciences" (ib: 3-4).

Esta valoración del papel de la abstracción en Computer Science, coincidió a principios de los años 80s con una campaña nacional iniciada en 1983, con el informe "Nation at Risk", que aún pervive, sobre la necesidad de poner la educación en matemáticas y ciencias al día en Estados Unidos.

En la área de la Computer Science, la orientación más pragmática fue progresivamente sustituida por el paradigma del "structured programming", con una creciente tendencia hacia la formalización.

En CMU, esta tendencia puso durante los años 80s la Lógica y la denominada "Logic Community" dirigida por Dana Scott, en el primer plano en el campo de la Theory en Computer Science. Merrick Furst y Nico Habermann trajeron a esta autoridad internacionalmente reconocida a CMU desde Oxford University donde entonces trabajaba afin de dar peso a esta área

Dana Scott ha sido y es el principal teórico de esta corriente. El Dr. Dana Scott es University Professor en Computer Science, Mathematical Logic and Philosophy. Nacido en 1932 en Berkeley, California, Dana Scott obtuvo el doctorado en Matemáticas en Princeton University en 1958 bajo la dirección de Alonzo Church. Fue profesor de filosofía en dicha universidad de 1969 a 1972 y de lógica matemática en Oxford entre 1972 y 1981, de donde pasó a CMU.

Fue el introductor del "inferential programming" con William Scherlis en 1983 (Scherlis & Scott, 1983), y es una autoridad internacional en el campo de las semánticas de programación. Según sus propias palabras: " My past work in logic has concerned mainly with model theory, automata, set theory, modal and intuitionistic logic, constructive mathematics, and connections between category theory and logic. Philosophical interests concern the foundations of logic, the philosophy of mathematics, and the semantical analysis of natural language.

...Work in computer science has been directed to the development of denotational semantics of programming languages". (CMU-CS-FRG. 1991-92:70).

Sus proyectos actuales se dirigen a unificar el enfoque semántico con formalismos de lógica constructiva para conseguir métodos de pruebas rigurosos y computables y el desarrollo de herramientas para la construcción inferencial de programas correctos. Así mismo, está utilizando métodos y programas informáticos, como el denominado Matemática, para la investigación y enseñanza de las matemáticas. Y al mismo tiempo, colabora en proyectos tecnológicos como el Mercury Project a fin de conseguir una "electronic library" en CMU, modelo de las bibliotecas electrónicas del futuro.

Si nos guiamos por sus áreas de interés dentro de la School of Computer Science, es el profesor que abarca un mayor número de ellas. En el curso 1991-92 eran 25 las áreas de interés de este profesor, seguido por Allen Newell con 20 y por J. Carbonell.

Estas 25 áreas eran: "Automated Theorem Proving", "Computation Theory", "Computational Linguistics", "Constructive Mathematics", "Foundations of Mathematics", "Functional Programming", "Inferential Programming", "Information Retrieval", "Interfaces, human-computer", "Logic", "Logic of Programs", "Music, Computer", "Program Transformation", "Programming Environments", "Programming Languages", "Programming Methodology", "Proof Theory", "Semantic of Natural Languages", "Specifications", "Symbolic Computation", "Text, Electronic", "Typesetting", "Type Theory", y "Verification".

La contribución fundamental de Dana Scott reconocida internacionalmente ha sido en el ámbito de las semánticas de lenguajes de programación, ya analizada en el capítulo sobre Programming Systems. Su aportación fue concebida como una forma de comprender los lenguajes de programación: "From the late 1960's my own work has concentrated on seeing whether the ideas of logic can be used to give a better conceptual understanding of programming languages". (1977: 634). Para ello, ha seguido un procedimiento de separación de la máquina del programa, y centrarse en las semánticas de éstos como sistemas formales. "I tried once in a certain way to connect automata and programming languages by suggesting a more systematic way of separating the machine from the program" (1977:636).

Todas estas áreas hacen referencia a sistemas formales, incluidos los programas informáticos. Pero esta operación tiene un riesgo: los programas informáticos no son meros sistemas formales, sino sistemas formales para que un ordenador digital los ejecute. Son un subconjunto del sistema informático, una función del mismo, como pueden ser los microchips de sus circuitos integrados.

Ya vimos al analizar el sistema operativo MACH que las abstracciones que introducía ("port", "message",...) eran abstracciones que funcionaban. No basta con introducir abstracciones en computer science. Estas han de funcionar, servir para escribir programas informáticos. En suma, hace falta abstracciones computables, "feasible".

La creencia en que la abstracción es aún más necesaria en la computer science que en la física pasa por alto que no se trata sólo de abstracciones matemáticas sino de abstracciones matemáticas computables como los números de la "Turing Machine". Sólo si son calculables por medios mecánicos, esas abstracciones son relevantes en computer science. No es la abstracción sino la abstracción computable lo que aprende a construir un "software engineer". No son matemáticas sino las matemáticas de la computación o aquella parte de la misma útil para

hacer tecnología.

Ello hace inviable que la lógica, siendo con todo útil, sea la fundamentación de la computer science, dado que su carácter en esta disciplina es instrumental.

Como señala el lógico italiano Giuseppe Longo, visiting professor en CMU en 1988: " There is a distinction which we feel a need to stress when talking (or writing) for an audience of Mathematicians working in Logic. It concerns the different perspectives in which Logic is viewed in Computer Science and in Mathematics. In the aims of the founders and in most of the current research areas of Logic within Mathematics, Mathematical Logic was and is meant to provide a 'foundation' and a 'justification' for all parts of Mathematics as an established discipline.... This is not so in Computer Science, where Mathematical Logic is mostly used as a tool, not as a foundation." (1988: 2). Longo pone el ejemplo del enfoque de Scott-Strachey para afirmar que dicho enfoque sí ha servido como fundamento para avanzar en los lenguajes de programación, pero añade: "However, the subsequent success of the topic, broadly constructed, is mostly due to use that computer scientists have made of the denotational approach in the design of new languages and software. ...In these cases, results in Logic, in particular in lambda-calculus and its semantics, were not used as a foundation, in the usual sense of Logic, but a guidelines for new ideas and applications...Thus Mathematical Logic in Computer Science is mostly viewed as one of the possible mathematical tools, perhaps a major one, for applied work". (ib:2).

Christopher Strachey (1916-1975), profesor de Computer Science en Oxford, donde lo conoció Dana Scott, y con quien elaboró su "denotational semantics", concebía el computing basándose en una filosofía de diseño todavía no muy extendida, según el propio Scott (Foreword in Stoy,J. 1977: xix). Como ha señalado éste: "His field was computing in a dynamic sense: how to get the machine to do what you want...Personally I think one of our greatest losses in his death is that his design philosophy was not yet mor widely known. Philosophy in design of machines (or algorithms or languages) is something that has to be learned by doing it.(1977: XIX).

La visión del computing de Strachey, había estado, según Scott, muy influída por Alan Turing, con el que Strachey había colaborado estrechamente.

Esta "design philosophy" se puede encontrar no sólo en Turing, sino en el propio Von Neumann, y su "logic design".

John Von Neumann es reconocido en la SCS como uno de los fundadores de la Computer Science. A diferencia de Turing, no obstante, Von Neumann no se anticipó en el diseño de ordenadores a los ingenieros. Según la historiadora Nancy Stern (1981) basó su "logical design" del ordenador llamado EDVAC en los trabajos ya en curso de Mauchly y Eckert que desde 1943 estaban construyendo el ENIAC en la Universidad de Pennsylvania. En 1945, Von Neumann publicaba su First Draft of a Report on the EDVAC, : "The first document to provide a logical framework for stored-program computers and for programming concepts". (Stern,1981:81).

Lo interesante del trabajo de Von Neumann es su idea de "logical design". Esta orientación hacía el uso de la lógica para diseñar máquinas de cálculo la siguió Von Neumann con sus colaboradores Herman H. Goldstine y Arthur Burks en el Institute for Advanced Study de Princeton University, produciendo un plan completo para el diseño de un ordenador electrónico, bajo contrato del U.S. Army. En 1946, Von Neumann y H. Goldstine escribían una serie de artículos para el diseño de dichas máquinas. El primero de ellos titulado, "On Principles of Large Scale Computing Machines". En él se decía: " During the recent war years a very considerable impetus has been given to applied mathematics in general, and in particular to mathematical physics, particularly in certain important fields which have not been in the past in the focus of most theoreticians' interest...Partly under the influence of war necessities, but partly also as a natural outgrowth of normal industrial development...these two streams of evolution have produced both an increased need for large-scale, high speed, automatic computing...In this article we attempt to discuss such machines from the viewpoint not only of the mathematician but also of the engineer and the logician, i.e. of the more or less...hypothetical person or group of persons really fitted to plan scientific tools. We shall, in other words, inquire into what phases of pure and applied mathematics can be furthered by the use of large-scale, automatic computing instruments and into what the characteristics of a computing device must be in order that it can be useful in the pertinent phases of mathematics". (1946:1).

Lo nuevo de este enfoque es que el sustantivo es el término "design" y el adjetivo es "logical". Se trataba de realizar la misma tarea que los ingenieros que estaban construyendo el ENIAC, pero en lugar de hacerlo en la práctica, realizarlo de una forma teórica, ésto es, diseñar la estructura lógica de dicho

ordenador.

El matemático se hacía ingeniero, cambiando el significado tradicional de la lógica desde Aristóteles, para transformarla en una herramienta para el diseño de máquinas para usos matemáticos. La lógica se hacía instrumento del diseño, o "logical design".

Von Neumann contribuyó a que esta nueva disciplina se la denominara "computer science", para distinguirla del mero diseño práctico de las máquinas, propio de los ingenieros. Pero con ello cambiaba todo el significado del concepto ciencia imperante hasta la fecha e introducía una confusión que aún no se ha despejado. El llamado "computer science" sólo puede entenderse como una rama de una teoría del diseño, y no al modo ni de la ciencia empírica ni de la lógica o las matemáticas.

Un paso más que dió Von Neumann fue considerar la actividad de programación de dichas máquinas más importante que su propia construcción, para diferenciar al "computer scientist" del ingeniero. En este informe se afirmaba que el "programming" o en palabras de Von Neumann, " the time of coding and setting up problems for such machine is the dominant consideration". (1946:26). Esta tarea daba a los matemáticos primacía sobre los ingenieros, entendidos como los que se encargaban del hardware. Así gracias a Von Neumann, la disciplina pasaba a llamarse "computer science", separándose de la ingeniería. Con el tiempo, esta definición ha mostrado sus limitaciones, creando una categoría intermedia entre el científico y el ingeniero, el "scientist of the artificial", que impide reconocer que existe una tecnología teórica y otra aplicada, y que la primera no necesita ser llamada "ciencia". Esa tecnología teórica tiene un conocimiento básico que es el diseño, una de cuyas versiones puede ser el "logical design". Después de 30 años de Computer Science se ve que no se puede avanzar en la programación de ordenadores sin sustanciales avances en el hardware, como se ha comprobado en la A.I.. Pero además, la misma programación de ordenadores se ha revelado que no es una tarea meramente científica sino ,sobre todo, ingeniera, de un nuevo tipo de ingeniería, la ingeniería del software. Es una tarea de diseño, de "logical design", donde la tarea es el diseño y la forma de hacer, la instrumentación lógica.

5.6. "ALGORITHMS AND ANALYSIS" EN CMU.

Según Merrick Furst, la área más característica de Computer Science es la de "algorithms".

La vieja matemática griega estaba basada ante todo en la geometría. Aristóteles introdujo la Lógica. La matemática moderna, por necesidades de la física, desarrolló ante todo el cálculo. Como instrumento para facilitar esta tarea se desarrollaron las primeras calculadoras mecánicas, (Pascal, Babbage). Finalmente para el desarrollo de éstas se ha tenido que desarrollar una matemática adecuada para escribir las instrucciones de cálculo de dichas máquinas. Y los algoritmos son ese tipo de matemática.

Pero se da la paradoja que no son los algoritmos los que han desarrollado los ordenadores digitales, sino al revés. Fue la idea de computabilidad, de números computables, la que ha hecho necesario el desarrollo de procedimientos para realizar esa tarea. Es una matemática discreta o finita para desarrollar un proceso mecánico. Es por tanto una matemática en función del ordenador, invirtiendo el planteamiento inicial, que era que el ordenador era un instrumento para el cálculo matemático. El ordenador ha desarrollado un tipo de matemática distinta para la que inicialmente se pensó, el cálculo numérico.

A veces para simplificar se compara un procedimiento algorítmico con una receta de cocina, que prescribe una serie ordenada de pasos para confeccionar un buen plato. Pero dicha receta es un simple instrumento del cocinero para confeccionar un menú. Es muy curiosa la lectura antropológica de este hecho. Como se sabe Levi Strauss asoció a la oposición "crudo-cocido" el propio nacimiento de la cultura. En efecto, la cocina es una de las primeras actividades culturales del sapiens. Es una actividad claramente tecnológica, de producción de un objeto artificial, que diferencia a esta especie de otras que no disponen de esta tecnología. Pues bien, el procedimiento algorítmico describe como realizar dicha secuencia de pasos para conseguir resolver el problema. Pero aún así deja sin resolver qué plato es el que ha de concinarse. Dice sólo cómo ha de hacerse. En este sentido es un estricto "know-how".

El algoritmo es la secuencia de operaciones para desarrollar la tarea. Pero reducir programa informático a algoritmo no permite entender que el programa tiene unos requerimientos previos y unas especificaciones que no están recogidas en el algoritmo, dado que éste es tan sólo un "effective procedure", esto es, un procedimiento efectivo, un proceso. La construcción de programas informáticos es más compleja, como vimos en el capítulo sobre software engineering, que la mera construcción de algoritmos.

No obstante, esta área de Theory es quizá la que donde más claramente se aprecia

la diferencia entre computer science y las matemáticas tradicionales.

Donald Knuth, matemático y profesor de computer science de Stanford University, dedicó en 1974 un artículo titulado "Computer Science and its Relation to Mathematics", donde analizaba este tema. Para él, computer science "is the study of algorithms. An algorithm is a precisely-defined sequence of rules telling how to produce specified output information from given input information in a finite number of steps". (1974: 323).

Un amigo matemático suyo al ser preguntado sobre el interés de este campo, le contestó que lo sería una vez que contuviera 1000 teoremas probados. El interés de la respuesta reside en que la matemática clásica se caracteriza por la construcción de teoremas y la construcción de sus pruebas correspondientes. Pero un algoritmo no ha de probar un teorema sino servir para ejecutar una tarea. Un programa no es un teorema. Los intentos de crear un "inferential programming" lo que han hecho es poner la inferencia lógica como instrumento para desarrollar ese nuevo concepto de programación. De nuevo vemos que la denominación "inferential programming", pone como sustantivo el "programming" y como adjetivo su carácter "inferential", reflejo que la cultura "profesional" o "technical" tiene una relación predominante sobre la "liberal", que la tiene como adjetivo.

Se afirma que los algoritmos son parte de la matemática. Y en efecto, han sido matemáticos los primeros en desarrollarlos, pero hasta la fecha ha sido un campo marginal. Todavía está por explorar si la computer science no conlleva un tipo distinto de matemáticas de las conocidas hasta ahora, una "Mathematics of Computation". Área en la que varios investigadores de CMU están trabajando: Furst, Kannan, G. Miller, Rudich.

Una de las constataciones de Donald Knuth en el artículo citado es que el uso de las matemáticas más apta para computer science eran del tipo llamado matemáticas discretas (aquellas que tratan con construcciones finitas). Y estas matemáticas están casi ausentes de los curriculums académicos. "The difference (between mathematics y computer science) is in the subject matter and approach --mathematics dealing more or less with theorems, infinite processes, static relationships, and computer science dealing more or less with algorithms, finitary constructions, dynamic relationships" (1974:326).

Es paradójico que una máquina que fue pensada en su inicio para efectuar cálculos, no se construya a base de cálculos sino de algoritmos. La razón para

ello es que su constructor no es un matemático sino un matemático constructor de máquinas matemáticas. No ha de realizar él mismo el cálculo, sino diseñar un artefacto que lo realice. Para ello, ha de saber matemáticas como el resto de matemáticos y algo más que el resto de matemáticos no necesitan saber: cómo hacer que la máquina efectúe esa computación. Esta asimetría cultural es la misma que vimos en el capítulo sobre Artificial Intelligence entre Bertrand Russell y Herbert Simon y su Logic Theory Machine.

El primero sabía lógica o matemáticas, el segundo debía saber también lógica y matemáticas y además construir una máquina que fuera capaz de razonar lógicamente.

Para ello, los computer scientists necesitan desarrollar un procedimiento efectivo, un algoritmo, para que ese cálculo o razonamiento lo realice un artefacto mecánico, lo cual no es el caso de los matemáticos o lógicos tradicionales.

Pero la construcción de algoritmos es un proceso de diseño. Los algoritmos a diferencia de los teoremas no se prueban, se inventan.

Ravindran Kannan es, junto con Merrick Furst, un líder del grupo de Algorithms en la School. Es "professor" en Computer Science and Mathematics. Obtuvo el doctorado en Cornell University como Merrick Furst. Según Kannan: " My primary interest has been to design fast (polynomial time) algorithms for problems that have an underlying mathematical structure." (CMU-CS-FRG. 1991-92:42).

Actualmente desarrolla un proyecto de investigación para la NSF, titulado "Random Walks, Parametric Integer Programming"(1991), cuyo objetivo es: " to improve the running times of algorithms to compute the volumes of convex sets, to do multivariate sampling and integration". (Kannan, 1992).

No son los algoritmos en general la área de interés de este grupo sino los algoritmos computables, "feasible", en palabras de Merrick Furst. Y este tipo de construcciones matemáticas ya se empiezan a poder patentar en Estados Unidos, como cualquier otro invento. Ello significa un cambio cultural de primer orden.

En 1972, La Corte Suprema de los Estados Unidos en el caso Gottschalk vs. Benson denegó la solicitud de Benson, un empleado de los Bell Laboratories, de patentar un algoritmo, alegando que por su naturaleza un algoritmo matemático era como una ley de la naturaleza o un principio científico tradicionalmente considerados descubrimientos, no patentables como los inventos técnicos. (Samuelson, P, 1990). Este caso suscitó un debate en todo el país que aún no se ha resuelto. En

el debate intervino Allen Newell alegando que habían una equivalencia entre algoritmos y procesos mentales y que de poder patentarse los primeros ello plantearía profundos problemas. (Samuelson, 1990:26).

El New York Times del 13 de Marzo de 1990 anunciaba : " The notion of intellectual property has also expanded into mathematics. Mathematical algorithms- the step by step procedures to solve complex problems -are now protectable as long as they are described in the context of a computer system. Before 1980, both the courts and the Patent Office rejected such claims on the grounds that algorithms were expressions about univerval laws of nature and therefore could not be owned" (1990: 12).

Desde el punto de vista de este estudio, lo interesante es que estos objetos matemáticos son diseños, y no sólo en el viejo sentido del constructivismo matemático de Brouwer y los intuicionistas de principios de siglo. Son construcciones matemáticas procesables por medios mecánicos, y por lo tanto, entran en en el campo de la tecnología, o mejor de la, high technology, pues los realizan científicos y matemáticos convertidos en un nuevo tipo de ingenieros. Al preguntarse a Kannan si se sentía más matemático o computer scientist, afirmó estar "on the border". (Entrevista.21 abril 1992).

5.7. PARADIGMAS TEORICOS EN COMPUTER SCIENCE DE CMU: "SEARCH", "LOGIC",... "DESIGN".

Programar ordenadores es una actividad eminentemente teórica, en el sentido que su contenido es el diseño de la acción que ha de ser ejecutada por la máquina. Pero es una actividad "teórica" en un sentido distinto al que tiene la teoría tanto en la ciencia experimental, como incluso en la matemática.

Según K. Popper, una teoría en la ciencia empírica es, ante todo, una teoría explicativa: "La decision de buscar una explicación causal es la misma por la que el hombre de ciencia teórico adopta su finalidad propia- o la finalidad de la ciencia teórica- Tal finalidad es la de encontrar teorías explicativas (si es posible verdaderas); es decir, teorías que describan ciertas propiedades estructurales del mundo que nos permitan deducir, valiéndonos de condiciones iniciales, los efectos que se trata de explicar". (1962:59).

Este no es el caso de la "Theory" en Computer Science. Un programa informático no explica una propiedad estructural de ningún artefacto del mundo, sino de una posible máquina a diseñar.

En los años 60s, el enfoque experimental cobró fuerza en Inteligencia

Artificial, basado en el paradigma del "heuristic search", impulsado ante todo por Herbert Simon. En realidad la Inteligencia Artificial nació con este paradigma. Con todo, el "search" es un instrumento del que se dota un programa informático para la selección de alternativas en la realización de una tarea.

La famosa búsqueda en el laberinto o "maze" que plantea Simon al modo de la clásica exploración de un científico experimental, es un dispositivo no para descubrir una realidad dada sino un mecanismo de toma de decisiones para ejecutar una tarea. La búsqueda se realiza para que el programa consiga los fines diseñados por el computer scientist, es un complemento del diseño del computer scientist no su fundamento.

En los 70s y sobre todo en los 80s, este enfoque fue progresivamente desplazado, tanto en A.I. como en Computer Science, por un enfoque formal que entroncaba con el de los matemáticos Turing o Von Neumann, padres fundadores de la disciplina, defendido en A.I. por John McCarthy. Como reconoce Herbert Simon: "Early A.I. research relied largely on the maze metaphor, but cognitive science research stemming from linguistics and logic preferred the reasoning metaphor...In recent years, the logic metaphor has been attracting increasing numbers of adherents, and I have become involved in the dispute". (1991:331). Este enfoque de la computer science teniendo como base la ciencia física está ahora potenciado en CMU por el equipo de Takeo Kanade dedicado a la Computer Vision.

El paradigma del SEARCH, propuesto por Simon, está en computer science, en función de una toma de decisiones del programa ante un conjunto de alternativas que éste tiene para efectuar una tarea. Como ha señalado Hans Berliner, se trata de un atributo de un programa informático, un "searching program" (1990: 311). Por lo tanto, "search", como paradigma de la ciencia empírica, no puede fundamentar la computer science.

Por otra parte, hemos visto que el paradigma LOGIC, sostenido por Dana Scott y la Logic Community, es considerado en computer science como una ayuda instrumental para la tarea de diseño de lenguajes de programación, de la actividad fundamental del programador.

La teoría de la "Complexity" fue iniciada en el ámbito de la computer science unida a la de Computability, y tiene apoyo en el desarrollo de una matemática basada en los algoritmos que ha tomado fuerza gracias al desarrollo de las máquinas informáticas. Pero concebir los algoritmos como la teoría de computer science deja fuera a la parte ingeniera, necesaria para la construcción de una

teoría unificada del sistema informático.

Mi tesis es que esta teoría puede conseguirse en torno al diseño como teoría unificada del "computer science and technology". El paradigma "KNOWLEDGE IS DESIGN" permite entender el marco cultural unificado donde se mueve la "computer culture" de CMU, y más allá de él la cultura ingeniera de CMU.

Esta idea de conocimiento como diseño es la conclusión de un cambio radical que empezó a introducir Alan Turing, con su visión de la "universal computing machine", y un nuevo tipo de teoría.

5.8. . "GREAT IDEAS IN COMPUTER SCIENCE": CONSECUENCIAS DE LA " TURING MACHINE"

Y LA "COMPUTABILITY".

Analizemos más a fondo este nuevo concepto de teoría introducido en Computer Science por Alan Turing, entre otros matemáticos-ingenieros. Curiosamente, hay dos grupos en la Theory de la SCS, "Logic" y "Algorithms". No hay un grupo sobre "Computability". Pero esta idea es quizá la más propia de Computer Science, e introduce un nuevo tipo de teoría.

Elegimos la figura de Allan Turing como la única en torno a la cual se reconocen corrientes distintas en CMU tanto los que provienen de la Artificial Intelligence como los del tradicional Computer Science. El "Turing Award" es, por otra parte, la máxima distinción de la Association for Computing Machinery de Estados Unidos que con sus 160.000 miembros puede considerarse representativa de lo que se denomina la "computer science community". Este premio lo han obtenido en CMU: Perlis, Simon, Newell, y Dana Scott, pudiendo considerarse que en lo referente a Computer Science, todos ellos se encuentran dentro del programa de investigación de Alan Turing.

Según Turing, el ordenador se puede considerar como una máquina abstracta, cuyo diseño no necesita la ingeniería a priori sino a posteriori, para validar esos modelos abstractos de computación. Este nuevo enfoque está en la base de la llamada "computer science".

Mary Shaw, profesora de la SCS, realizó en mayo de 1991 una encuesta electrónica denominada "Great Ideas of Computer Science" a través de la red de la escuela. El objetivo de la misma era recabar entre profesores y estudiantes aquellas ideas que en la SCS se consideraban propias de la disciplina: " Have you ever tried to explain why computer science is exciting, why it deserves to be a separate discipline, why it is a science, or even what computer science is?...In many

other disciplines, you can single out a few great ideas that capture the main concepts and phenomena that are central to the discipline...The cell doctrine in biology,...Doctrine of atomism..., Theory of Evolution,...What about computer science?. Do we have laws of qualitative or quantitative structure?. What topics do you cite to show that computer science is different from mathematics (or electrical engineering, or whatever)?".(15 abril 1991. Red Andrew. CMU).

Contestaron a las preguntas 20 profesores y estudiantes de la escuela. Las grandes figuras de computer science que se reconocían eran: Alan Turing, Alonzo Church y Von Neumann. Entre sus grandes ideas se destacaban la Máquina de Turing, la Thesis de Church y el modelo de ordenador de Von Neumann. Por su parte, entre las "rejected ideas and bad ideas" algunos encuestados situaban la Physical Symbol System, idea propia de la Inteligencia Artificial.

Fue Turing el que formuló un modelo teórico de máquina capaz de realizar todo tipo de cálculos, o "universal computing machine". Así mismo sostuvo la hipótesis que dichas máquinas servían no sólo para efectuar cálculos aritméticos sino que podían procesar símbolos, y como tales, eran capaces de realizar funciones que en el ser humano se definen como inteligentes. Estas máquinas podían considerarse también inteligentes, dando pie a la investigación en lo que más tarde se denominaría Inteligencia Artificial.

Turing inventó tres modelos de máquina: los denomina "Turing Machine", la "Universal Computing Machine" en 1936 y 10 años después, en 1946, la "Automatic Computing Engine" (ACE).

Los dos primeros son los que aparecen en su artículo "On computable numbers, with an application to the Entscheidungsproblem" (1936). El tema de este artículo es el de los números calculables por una máquina, o "computable numbers". Según la definición de Turing, " a number is computable if its decimal can be written down by a machine". (ibid:230). Anotemos la expresión "can be", pues salió ya como forma de presentar los problemas en la cultura tecnológica. Esta no se pregunta por el "ser" al modo filosófico sino por lo que "puede ser". El matemático Alonzo Church había introducido en abril de 1935 la idea de "effective calculability" (1936). Según esta tesis el conjunto de funciones matemáticas conocidas como funciones recursivas eran idénticas al conjunto de funciones calculables por medio de procedimientos efectivos. Pero Alan Turing dio un paso más: definió la "effective calculability" como "computability", ésto es, como funciones calculables por una máquina. Computability era la actividad

de la Turing Machine.

Como indica Josep Diaz, de la Facultad de Informática de Barcelona: " En 1936, el británico Alan Turing (1912-1954) dió de manera independiente de la de Church y Kleen una nueva formalización del concepto de procedimiento efectivo, utilizando un modelo de máquina general...La innovación en este modelo de computabilidad es que define perfectamente lo que es un paso de computación, cosa que en los otros modelos, cálculo lambda y funciones recursivas, no está bien definido. El concepto de paso de computación será clave cuando comience a estudiarse la complejidad de las computaciones" (1984)

Si acordamos ,con Alan Perlis, que lo nuevo en la informática no es el estudio de los algoritmos, que ya existían desde hacía milenios, sino su unión a una máquina, ("What will be new is the attachment of algorithms and languages to a machine -the electronic digital computer-"(1971:2), la gran idea de la computer science empieza con la "Turing Machine", y su noción de "computability", la actividad de cómputo de dicha máquina.

Al unir procedimiento efectivo con el ejecutado por la máquina llega a la conclusión que no todos los problemas matemáticos podían ser resueltos por un procedimiento mecánico. La máquina de Turing fue la primera en representar una función no computable y por lo tanto en abrir una escisión entre matemáticas y computer science.

Este paso podría considerarse decisivo en la separación de la computer science respecto al universo de las matemáticas y la lógica. Esta problemática es la que ha conducido al tema de la "complexity", subarea de la "Theory", en la que es un experto Joseph Traub, segundo director del Departamento de Computer Science de CMU(1971-79) . Esta área estudia qué problemas son intratables por medios mecánicos, esto es, cuáles son o no computables. Y si tenemos en cuenta que todo el edificio de la ciencia se basa en la idea de la comprensibilidad del Universo mediante métodos matemáticos, la informática pone límites al sistema de conocimiento científico. Si hay problemas científicos imposibles de calcular por el ordenador, ello significa que no son calculables y en términos matemáticos, cognoscibles, y por lo tanto el primer principio de la ciencia, la comprensibilidad del universo entra en crisis. Esta problemática ya se planteó en los años 30s por Heisenberg en la física y por Gödel en la matemática. Pero Turing no se limitó a ahondar la crisis del paradigma científico clásico, sino que abrió una salida a esta problemática, empujando a la ciencia hacia un nuevo

campo: la ingeniería.

La contrapartida de su enfoque es que permitía una sustancial ampliación del universo de problemas resolubles al convertirlos en problemas computables. Hoy día, gracias al ordenador, ha nacido una nueva rama de la ciencia, distinta a la teórica y a la experimental, la llamada "scientific computing" que permite utilizar algoritmos informáticos para descubrir nuevos fenómenos naturales imposibles de alcanzar para las otras dos ramas.

El enfoque de Turing fue claramente tecnológico: en lugar de centrar su interés en la serie de números definibles por la mente humana, la centró sólo en aquellos calculables por la máquina. Para ello construyó dos modelos teóricos de máquina: la denominada "finite state machine" y la "universal computing machine". La primera calculaba secuencias ininterumpidas de 0 y 1. La segunda, era una máquina que podía computar cualquier secuencia computable. Esta tesis fue formulada así por Turing: "Is it possible to invent a single machine which can be used to compute any computable number" (1936:241).

La noción de "computable numbers" expresa una idea central a la nueva disciplina: la limitación del universo de problemas de la matemática, a cambio de ampliar el campo de aplicaciones de las máquinas hasta entonces diseñadas por la ingeniería.

Estas máquinas teóricas trataban de ser un modelo práctico para estudiar los números computables, un subconjunto de todos los números definibles.

"I give you some arguments with the intention of showing that the computable numbers include all numbers which could naturally be regarded as computable...The computable numbers do not, however, include all definable numbers....Although the class of computable numbers is so great, and in many ways similar to the class of real numbers, it is nevertheless enumerable..."(1936:230-231).

Según Turing, una de las "applications" de este modelo de "computability" fue probar que el problema lógico matemático de la decidibilidad (Entscheidungsproblem) de la escuela formalista del lógico alemán Hilbert (1826-1943) no tenía solución. No había un procedimiento general o algoritmo, para determinar si un enunciado se podía o no probar. Este artículo de Turing es clave para entender la cultura de "computer science". Para Marvin Minsky, "it contains, in essence, the invention of the modern computer and some of the programming techniques that accompanied it"(1967:104).

Gödel con su teorema de la incompletud aparecido en 1931 había demostrado que en el sistema formal de Russell y Whitehead existían teoremas indecidibles, para los que ni ellos ni su negación podían ser demostrados dentro del sistema. Cinco años más tarde, Turing, probó que no todo enunciado matemático podía ser probado siguiendo un algoritmo general. No todo problema matemático podía ser resuelto por un procedimiento mecánico.

Tanto Gödel como Turing eran matemáticos, pero el enfoque de ambos era distinto. El primero era un lógico matemático dentro aún del universo teórico de los Frege, Russell, Whitehead, un mundo intelectual ajeno a la tecnología. Turing era también un matemático pero cuyo objetivo principal fue utilizar las matemáticas para desarrollar un modelo teórico de máquina universal, una de cuyas aplicaciones era el tratar de probar la no viabilidad del problema matemático de Hilbert.

Así explicaba Turing, el 20 de febrero de 1947, el significado del citado artículo, ante la London Mathematical Society: " Some years ago I was researching on what might now be described as an investigation of the theoretical possibilities and limitations of digital computing machines. I considered a type of machine which had a central mechanism, and an infinite memory which was contained on an infinite tape...It was essential in these theoretical arguments that the memory should be infinite. It can easily be shown that otherwise the machine can only execute periodic operations. Machines such as the ACE may be regarded as practical versions of this same type of machine".(:106-107).

Este enfoque dirigido a la exploración teórica sobre las posibilidades de estas máquinas lo repitió Turing para explorar si éstas podían pensar. En 1950, escribía en el artículo Computing Machinery and Intelligence: " Propongo que se considere la cuestión: '¿Pueden pensar las máquinas?'. " (1974:11). Para verificar esta posibilidad proponía un test denominado el juego de imitación, por el cual, un ser humano interrogaría a una máquina sin saber si las respuestas las emite la propia máquina o un ser humano. Cuando el interrogador no llegara a identificar quién es quién se habría comprobado que las máquinas podían pensar.

Lo interesante de este modo de plantear los problemas es que generan una nueva forma de teoría distinta a la clásica de la ciencia experimental. Se pregunta por cómo pueden ser las cosas, no como son. Pero estas posibilidades teóricas,

lejos de ser meros experimentos mentales, se pasan a continuación a su prueba, en términos ingenieros, para ver si, en efecto, son o no posibles. No se trata tampoco de meras construcciones matemáticas posibles pero irrealizables. El propio Alan Turing diseñó un plan para la construcción de una máquina denominada Automatic Computing Engine (ACE) para la Mathematical Division of the National Physical Laboratory en 1946. (Turing , 1986).

Antes que Alan Turing, Charles Babbage había ya construido en 1864 una Máquina de Diferencias para hallar valores de funciones. Se diseñó como una herramienta para formar tablas matemáticas más exactas. Pero Alan Turing no perseguía construir simplemente una herramienta, sino explorar las posibilidades teóricas de una "universal computing machine".

Si por ingeniería se entiende la simple tarea de construcción de sistemas.

Turing no fue un ingeniero. Se interesaba por las posibilidades teóricas no por las meras realizaciones prácticas. Hasta ahora se ha opuesto saber teórico a saber operativo. Los filósofos platónicos y la ciencia en general se ha puesto de la primera banda, y los pragmáticos y la ingeniería, de la segunda. Pero esta visión no ayuda a entender qué nuevo tipo de teoría está emergiendo en "computer science", y qué tipo nuevo de práctica.

Alan Turing hizo ambas cosas: primero formuló un modelo de máquina de calcular universal y luego pasó a construirla para probar que su modelo era viable. Su propósito era teórico no simplemente práctico. Pero su modelo teórico no era científico pues no se trataba de analizar una máquina existente sino una "theoretical possibility".

Hasta ahora a este nuevo tipo de teoría se le ha continuado llamando "science".

Por mi parte prefiero llamarlo ingeniería teórica o nueva ingeniería. Y su concepto de teoría es el diseño, o si se prefiere la invención, en un sentido no tan sólo de construcción de nuevos sistemas o "building systems", sino de diseño de nuevos sistemas teóricamente posibles tanto mecánicos como biológicos, sociales o culturales , y su validación a través de la construcción de sistemas prácticos que funcionen y confirmen la validez de los anteriores modelos.

Hasta Turing, la teoría se ha concebido desde Platón como contemplación de la Idea. "Theoros" en la tradición filosófica occidental desde Aristoteles significa el "espectador de la verdad". (M. Medina.1989:37). El ideal de vida teórico, el "bios theoretikos", era el ideal contemplativo. Frente a ella se situaba la vida práctica o artesanal, basada en las "technai" o técnicas. Y así

sigue siendo, tras más de 2000 años, en el esquema denominado "Science & Technology".

Se podrá arguir que la ciencia moderna no es contemplativa como la de Platón sino experimental como la de Bacon. Y así es. Pero la ciencia inductiva de Bacon, sigue teniendo como primera premisa, la comprensión de las leyes de la naturaleza, para una, vez comprendidas, utilizarlas en forma de tecnología. Por eso ese modelo se denomina "Science & Technology". En él, el primado teórico lo sigue teniendo la física, antes newtoniana, hoy relativista.

Pero en "computer science", la relación se ha acabado invirtiendo. La tecnología se adelantado a la ciencia empírica. El diseño de dicha máquina es y ha de ser previo a su descubrimiento experimental. Pero esa tecnología ya no es mera práctica o saber operativo, se ha hecho teórica, y ha empezado a destacar el elemento central de este nuevo tipo de saber: el diseño.

La Turing Machine es una teoría de distinto tipo. Turing no se basa en contemplar ninguna realidad o idea sino que proyecta una nueva realidad que se ve posible inventar. La nueva teoría de la "universal computing machine" la formula Turing así: " It is possible to invent a single machine which can be used to compute any computable sequence" (1936: 241). Este tipo de hipótesis es de la misma clase que la de Physical Symbol System, analizada en el capítulo de la Inteligencia Artificial, pero formulada más explícitamente.

Se afirma en la SCS que Computer Science es una ciencia. No obstante, la área de Theory es quizá el mayor punto de incertidumbre de toda la escuela. Y no sólo en esta escuela.

El Computer Science and Technology Board del National Research Council de Estados Unidos, veía así en 1988 esta área: "Computer Science is a young discipline, and its theoretical base is immature.... Since computer science is an artificial science (Simon 1981), theoretical computer science plays a very different role within computer science than, say, theoretical physics plays within physics. Theoretical physics seek to understand the physical universe, which exists independently. Theoretical computer scientists seek to understand all possible architectures or algorithms, which computer scientists create themselves". (1988:60).

Si es verdad la idea que March y Simon han apuntado, en el campo de teoría de las organizaciones, sobre lo que denominan "puntos de absorción de incertidumbre", según la cual es en ellos donde se generan las innovaciones en

una organización, donde se cambian los modelos (March, Simon, 1981: 209).

Computer Science es hoy , quizá, un punto de absorción de incertidumbre clave de todo el sistema de Science & Technology.

5.9. NEWTON Y TURING, DOS TIPOS DE TEORIAS DISTINTAS, DOS TIPOS DIFERENTES DE

SISTEMAS MECANICOS: DE NUEVO, SOBRE LA INVERSION DE CONOCIMIENTO.

Es interesante llegados a este punto comparar la Computer Science con la mecánica de Newton, paradigma de la ciencia moderna. Ello nos puede ayudar a entender la ruptura y los límites de la computer science. Newton, como es sabido, estableció un sistema del mundo inspirado en la mecánica, Turing, por su parte, pretendió construir una máquina de cómputo de carácter universal.

Mientras que para Newton, el objeto central de estudio era el universo, compartiendo en éste el enfoque que arranca de los filósofos jonios, para Turing, el objetivo principal es la máquina de cálculo universal, lo que era de escaso o ningún interés para la filosofía clásica griega. Para el primero, el modelo de la mecánica artesana le sirvió de ayuda para entender el funcionamiento del universo, para el segundo, la ciencia universal, las matemáticas, le servía para construir una máquina.

Esta relación invertida la detectó agudamente Marvin Minsky en su obra:

"Computation: finite and infinite machines" (1967). En el capítulo titulado "Machines as Physical Models of Abstract Processes" desarrolló esta idea. Dice Minsky: " There is a curious contrast between this idea of a machine and the idea of a 'theory'. Consider some 'theory' of physics, e.g., Newton's mechanics. This theory (or any other theory of physics) is supposed to be a generalization about some aspect of the behaviour of objects in the physical world. If the predictions that come from the theory are not confirmed, then (assuming that the experiment is impeccable) the theory is to be criticized and modified...After all, there is only one universe and it isn't the business of the physicist to censure it, much as he might like to.

For machines, the situation is inverted!. The abstract idea of a machine, e.g., an adding machine, is a specification for how a physical object ought to work. If the machine that I build wears out, I censure it and perhaps fix it. Just as in physics, the parts and states of the physical object are supposed to correspond to those of the abstract concept. But in contrast to the situation in physics, we criticize the material part of the system when the correspondence

breaks down." (1967:5-6). Podemos advertir que el mismo título de Minsky, "Machines as Physical Models of Abstract Processes", invierte lo que sería propio de una ciencia natural, la construcción de modelos abstractos de fenómenos físicos.

En efecto existe esa inversión. Pero no se basa en una contraposición entre máquina y teoría, como afirma Minsky sino a una diferencia entre dos tipos distintos de teoría : la que podríamos llamar una teoría científica y lo que sería una la teoría ingeniera. La primera basada en el descubrimiento y la segunda en el diseño.

Para la ciencia , la teoría se expresa mediante leyes que describen el comportamiento de fuerzas naturales. Para la ingeniería, la teoría es el nuevo sistema mecánico que el ingeniero o , en este caso. el "computer scientist" ve posible inventar.

Conviene profundizar en las diferencias entre estas dos disciplinas.

El paradigma dominante de la ciencia moderna fue establecido por Isaac Newton con sus *Philosophiae Naturalis Principia Mathematica* en 1686. Curiosamente su sistema del mundo de Newton se basaba en la mecánica, la rama de la física que estudia el movimiento de los cuerpos, tanto naturales como artificiales (como los proyectiles). La ciencia moderna comenzó con la visión del universo como una máquina, separándose del universo geométrico de raíz pitagórica.

El sistema de mundo de Newton parte de un movimiento ordenado de los cuerpos recurriendo a la analogía del artefacto mecánico. Esta analogía comienza con el filósofo jonio Anaximandro en el siglo VI a.C., el primero en explicar el mundo en términos mecánicos.

A diferencia de la versión pitagórico-platónica, el mundo para Anaximandro no estaba concebido a modo de una estructura geométrica sino como un mecanismo semejante a las ruedas de un carro. Este modelo físico se extrajo, pues, por analogía con un invento humano. Como indica S. Sambursky : " These two models -the revolving wheels and the fire appearing at the mouth of the forge- are perfect examples of technical analogy...The use of the mechanical analogy completes the picture of the scientific approach which distinguishes Ancient Greece from all that went before" (1956:15).

Pero esta visión no fue la más aceptada en la antigua filosofía griega. La disciplina hegemónica entre los filósofos griegos clásicos no fue la mecánica sino las matemáticas, y dentro de ellas, la geometría, y su aplicación a la

comprensión del universo entendido como Cosmos. El mundo de la filosofía griega clásica, griego es un universo de esencias (eidos), no de movimiento, con la excepción de Heráclito. La corriente mayoritaria en la filosofía clásica discurre desde Pitágoras, Parménides, Platón a Aristóteles, desde la matemática a la lógica.

La aceptación de la física, como disciplina hegemónica dentro de la ciencia, y por ende del pensamiento humano, y la visión del universo como un "mechanical world" es obra de la ciencia moderna, iniciada con la revolución copernicana, llegando a su cénit con la mecánica de Newton. Esta teoría permitió considerar el universo como un sistema de cuerpos móviles regido por la gravitación universal.

Este paso del universo geométrico al universo mecánico se aprecia en el prefacio a sus Principia: "To describe right lines and circles are problems, but not geometrical problems. The solution of these problems is required from mechanics...Therefore geometry is founded in mechanical practice, and is nothing but that part of universal mechanics..." (1946: XVII).

Las mitoculturas han considerado generalmente el universo y todo lo existente como creación de los dioses. Mitos sobre la creación del mundo se pueden encontrar en casi todas las culturas. La moderna civilización iniciada en Grecia, empezó a explicar el universo como Cosmos, como orden natural, regido por la Razón o Logos. Este proceso de explicación del orden natural y social en términos estrictamente racionales no es, con todo, exclusiva de Occidente. La filosofía china clásica de Confucio y Lao Tse o los Upanishad indúes introdujeron un pensamiento que podríamos calificar de logoconocimiento. Pero ha sido Occidente, el que lo ha llevado teorizado de forma más completa, y lo ha organizado culturalmente de manera más estructurada, como se aprecia en su sistema universitario.

En la etapa más reciente de la civilización occidental, este "orden natural" se ha considerado como un gigantesco sistema mecánico, inspirado en gran parte en la obra de artesanos y mecánicos. La ciencia moderna, a diferencia de la griega, ha ido progresivamente estrechando sus lazos con la ingeniería, cuyo mayor exponente es la ciencia de Francis Bacon, y su lema "Knowledge is power", lema que hoy recogen ingenieros del conocimiento como Edward Feigenbaum. Con todo este "orden artificial" o técnico, construido por artesanos y mecánicos, es visto generalmente como una consecuencia del orden mecánico

universal, descubierto por la física. Justo al revés de cómo nació la idea del universo mecánico. La oposición binaria cielo-tierra que aparece en diversas mitologías, se ha traducido en la cultura moderna en la oposición mecánica celeste-mecánica artesana. Durante milenios, los filósofos, matemáticos y científicos han utilizado la analogía mecánica para explicar el sistema del mundo. Incluso Platón considera a dios como un Demiurgo, un artesano, cuando los "demiurgois" en Grecia era el nombre que recibían los mecánicos y artesanos.

El propio Newton reconoce basar su mecánica en esta doble acepción de la mecánica: "The ancients considered mechanics in twofold respect: as rational, which proceeds accurately by demonstration, and practical. To practical mechanics all the manual arts belong, from which mechanics took its name".

(1946:XVII). Pero, aún siendo Newton un hábil mecánico, (construyó en su juventud juguetes y modelos mecánicos), prefirió dedicarse a esta disciplina como filosofía: " I consider philosophy rather than arts and write not concerning manual but natural powers, and consider chiefly those things which relate to gravity, levity, elastic force, the resistance of fluids, and the like forces...and therefore I offer this work as the mathematical principles of philosophy, for the whole burden of philosophy seems to consist in this- from the phenomena of motions to investigate the forces of nature, and then from these forces to demonstrate the other phenomena." (ib: XVII-XVIII).

Dada la pequeñez e insignificancia hasta ahora de los artefactos mecánicos humanos en comparación con la grandiosidad de la mecánica celeste, era lógico pensar, sobre todo si se tenía una creencia teísta como Newton, que el mundo de la mecánica celeste determinaba la mecánica artesana, en suma, el "natural world" determinaba el "man-made world". El mundo del descubrimiento dominaba el mundo de la invención. Y en gran parte así ha sido.

Pero el llamado "man-made world" se ha hecho cada vez más grande. El mundo cultural, lo que de específico, ha aportado el sapiens a un orden que surgió antes que él, es ya lo suficientemente adulto para al menos determinar el resto de fenómenos considerados hasta ahora "naturales", al menos en este planeta.

Vivimos en una cultura cada vez mas tecnológica. La Turing Machine y sus retoños son es máquinas artificiales humanas nos permiten hacer ahora los cálculos que nos pueden permitir modificar el medio ambiente en este planeta así como empezar a construir culturas humanas más allá del mismo.

Hasta ahora esta herramienta era un simple medio para un fin, la comprensión del

funcionamiento del universo. Pero empezamos a advertir que este mundo artificial humano, es también un fin en sí mismo, un artefacto diseñable. Las "computing universal machines" de Turing nos han ayudado a entender que puede abrir paso una teoría, un nuevo tipo de saber, basado en el diseño en un sentido general, esto, cultural. Y la cultura de CMU es un laboratorio interesante para ensayarla.

PARTE CUARTA.

CMU COMO " DESIGN CULTURE".

(DE "COMPUTER-INTENSIVE CAMPUS" A "GENERAL DESIGN INSTITUTION")

1. "NATURAL WORLD, "ARTIFICIAL WORLD": "DESIGN WORLDS"

Carnegie Mellon es uno de los centros principales a escala internacional en la computer science. A finales de los 60s, un profesor de esta universidad, Herbert Simon, empezó a reflexionar sobre la creciente importancia de la tecnología, partiendo de una visión de dos sistemas de mundo, "the natural and the artificial world", el primero el de la física tradicional, el mundo de Newton, y el segundo el mundo construido por el ser humano o "man-made world". Este segundo, para Simon, sería el mundo de la ingeniería basado en el "design". El ordenador sería su elemento más representativo. Este texto es importante en tanto que representa la etnocosmovisión de lo que Simon unas veces " the scientist of the artificial" y otras "a designer" (1981:157). Iniciamos el estudio con éste tipo humano y lo acabaremos haciendo referencia al mismo pues tiene relación directa con el mundo de la cultura, el mundo de la antropología. Esta división binaria "mundo natural"- "mundo artificial" no es nueva. Ya hemos visto que Newton ya la reconoce al separar el "mechanical universe" de la filosofía natural de la mecánica de los artesanos. Lo nuevo que apunta Simon es lo siguiente: The world we live in today is much more a man-made, or artificial, world than it is a natural world".(ibid:4-5). Lo que se deriva de aquí es que la tarea de diseño de este "man-made world" es crecientemente más impotante que la de descubrimiento del "natural world". Ello explica que el denominado "designer" vaya cobrando cada vez más relevancia en relación con el "natural scientist", que en épocas pasadas.

Pero el "man-made world" ha sido, precisamente, el tipo de mundo sobre el que se constituyó la antropología como ciencia social distinta a las restantes y dedicada a estudiar aquellos fenómenos específicamente propios del sapiens en su vida organizada, en suma, el mundo cultural. El "artificial world", el "man-made world", y mundo cultural pueden considerarse términos sinónimos, y describen el conjunto de aportaciones propiamente humanas, aquellas construidas e inventadas por las distintas culturas humanas.

Hasta ahora, la antropología se ha acogido al paradigma de las ciencias naturales, y como tal, se ha dedicado a estudiar, principalmente qué culturas humanas existían y cómo funcionaban, empezando por las llamadas primitivas inicialmente, siguiendo por las industriales, a partir de los años 30s, y llegando finalmente a las comunidades de la denominada "information society". De los pequeños grupos y clanes locales ha pasado a estudios de comunidades étnicas cada vez más amplias, incluso de carácter nacional durante la II Guerra Mundial. Finalmente, se empezó a plantear en los años 60s por Margaret Mead, Levi-Strauss y otros la posibilidad de una "world anthropology". La siguiente generación estamos ya empezando a estudiar ahora culturas de la "information society". Algunas de ellas son marcadamente tecnológicas como CMU y tienen la innovación como un valor principal.

Respecto al carácter de la disciplina, esta ha sufrido también una evolución. De un carácter predominantemente académico en su origen, y aún hoy, aunque no exclusivo, la antropología ha sufrido una evolución no distinta a las otras ciencias orientándose cada vez más, sobre todo en momentos de crisis social, a interesarse por resolver problemas del llamado "real world". Se ha ido fortaleciendo su carácter profesional, en forma de "applied anthropology".

John Van Willigen, de la Universidad de Kentucky, y uno de los principales promotores de este tipo de antropología la define como: "a complex of related, research-based, instrumental activities which produce socially desired change or stability in specific cultural systems through the provision of data, the initiation of direct action and /or the formulation of policy". (1980: 3).

Incluso en los años 70s se ha desarrollado en Estados Unidos una nueva versión denominada "practicing anthropology", para designar una antropología realizada desde fuera de la Universidad.

La pregunta sería cuál es la idea central de esta antropología aplicada, o destinada entre otras cosas a producir "social desired change". Para Erve

Chambers la idea es "policy": "the idea of policy is as central to the development of applied anthropology as the concept of culture has been to the anthropological profession as a whole" (1985:37-38). La tarea de diseño en los antropólogos profesionales sería la del diseño de "policy". En un sentido amplio del término, una de cuyas variantes es la denominada "public policy", pero puede haber políticas para cada tipo de comunidad humana.

De la antropología aplicada nace una vía hacia la antropología profesional que introduce en directo al antropólogo en la cultura tecnológica como profesional, en suma como "designer", con todas las posibilidades y dificultades que ello comporta. El antropólogo sería así un diseñador de políticas o programas de comunidades culturales a distintos niveles, lo que Claudio Esteva llamaba: "el arte de la programación de una nueva conciencia de la realidad" (1975:268).

Si las nuevas disciplinas tecnológicas se preguntan, como hemos analizado, qué nuevos sistemas tecnológicos pueden construir y cómo hacerlo, el antropólogo diseñador se plantearía que nuevos sistemas culturales se pueden edificar y cómo hacerlo. Esta línea de investigación ha de ser explorada en próximos trabajos.

El limitarse a ser meros analizadores las comunidades de científicos y tecnólogos o de los impactos sociales que esta tecnología elaborada por los ingenieros produce, es una opción que vemos ya insuficiente conforme más complejos y más complejos son los problemas culturales que plantea la tecnología.

Los "computer scientists", junto con bioingenieros, y otros investigadores de nuevas tecnologías, están construyendo nuevas naturalezas artificiales en forma de nuevos y cada vez más complejos sistemas tecnológicos, ya no meramente mecánicos, sino biológicos también.

Sin embargo, su límite estriba en que estos inventos técnicos quedan frenados cada vez que se intentan generalizar a escalas más complejas, a escalas que afectan en directo al propio ser humano. Entonces se recurre a la "liberal education", a la ética, a las ciencias sociales para socorrer a la tecnología.

Los ingenieros piden ayuda a las ciencias sociales (psicología, economía, antropología,...), a las matemáticas o la lógica, a la filosofía, e incluso a la religión, y a la ética, como está pasando ahora con la biotecnología.

En CMU, la necesidad de dar una formación más amplia al ingeniero, una "general education", que no fue otra que la de situar los problemas tecnológicos como problemas culturales complejos, ya ocurrió con el plan Doherty, y volverá a

pasar ahora ante los límites de la "computer culture". Esta necesidad se cubre, una y otra vez, con la llamada "liberal education" pero ésta no está pensada para resolver problemas sino para analizarlos. El instituto tecnológico se convierte en universidad, y el problema tecnológico se transforma en científico, diluyendo su especificidad.

2. UNA CULTURA PROFESIONAL GENERAL EN TORNO AL DISEÑO.

Lo que parece una dificultad insuperable es extender los principios del diseño propios de la tecnología, e incluso el Computer Science, a áreas más generales y complejas que las máquinas o sistemas técnicos más elementales.

La hipótesis del antiguo profesor de historia de CMU, W. Andrew Achenbaum resulta difícil de realizar: " Is it not possible that the concept of design is an appropriate basis for beginning to think about some interconnections among the liberal arts, science and technology?...If this hypothesis is tenable, then CMU is an ideal environment for demonstrating its salience." (1985, Focus, vol 14, No6.)

Esta hipótesis no tuvo en cuenta que la matriz de CMU es claramente ingeniera y que ésta no puede llevarse a cabo sin la participación de este grupo como dirección del proceso, al menos en un primer momento. No tuvo en cuenta, a su vez, el enorme atraso de las humanidades y las ciencias sociales para afrontar el reto de convertirse en disciplinas diseñadoras. Con todo, fue una idea sugerente.

Esta hipótesis para realizarse exigiría para empezar una revisión del propio Plan Doherty que dió lugar a la actual "liberal-profesional education" .¿ Puede CMU plantearse ahora una revisión del Carnegie Plan para adaptarlo a unas condiciones culturales distintas a las de hace 50 años?. Los años 90s son una década de reforma general de las universidades en Estados Unidos. ¿Puede CMU ponerse a la cabeza de esta reforma, como hizo en los años 30s con el Carnegie Plan?.

Esta reforma se planteará cada vez más urgente. La división "natural world"- "artificial world" resulta ya anacrónica. Los investigadores en nuevas tecnologías no se limitan ya a crear "artificial worlds". Están sintetizando crecientes áreas de lo que hasta ahora se ha llamado "natural world". No se trata sólo que está construyendo artefactos mecánicos distintos a "la naturaleza" , ni siquiera está ya construyendo simulacros o simulaciones de naturaleza sino una nuevas naturalezas artificiales, en forma de nuevas

combinaciones de sus elementos básicos, sus átomos, sus moléculas, sus códigos genéticos, las unidades básicas de la mente humana, como sus procesos de razonamiento lógico y finalmente sus propios valores culturales, en suma, nuevas culturas.

Los efectos de esta visión en el terreno de los sistemas de conocimiento apuntan cada vez más hacia una primacía de la ingeniería sobre la ciencia.

En realidad, se trata de una nueva ingeniería. El tema del diseño es relativamente reciente en la ingeniería como tema central. Hasta ahora por ingeniería se ha entendido "the art or science of making practical applications of the knowledge of pure sciences" (Florman, 1976, x.). O también, el arte de controlar las fuerzas naturales en beneficio de las necesidades de la sociedad.

El ingeniero en CMU no se concebía como "designer" sino como "problem solver" en la época de Doherty. Y su método no era el "design" sino el llamado "learning by doing". La emergencia del diseño como tema central de la ingeniería como más adelante veremos arranca en los años 60s.

Ha sido la emergencia de un nuevo tipo de ingeniería basada en la síntesis (Sanmartín, 1989) no en el control de fuerzas naturales, lo que ha puesto de relieve la importancia del diseño. Ahora se están sintetizando nuevos materiales, nuevos seres vivos, nuevos tipos de máquinas, en suma se trata de "ingenierías de síntesis", más que de ingenierías de control, y tienen su propias teorías en torno a éstos conceptos (A. Westerberg, 1979, 1988).

La nueva revolución no es una "control revolution" (Birbaum, 1986) sino una "design revolution". Un viejo cartel en un aula de ingeniería civil de Baker Hall, en CMU, que ha sobrevivido al menos hasta este año de 1992, decía: "The design revolution starts here".

Este tipo nuevo de ingeniería plantea grandes retos a las humanidades y ciencias sociales. El problema no es ya la pregunta clásica de la antropología filosófica: ¿Qué es el ser humano?, o de la antropología cultural tradicional, ¿qué son sus diferentes culturas y como funcionan?, Sino la de una antropología del futuro o tecnoantropología: ¿Qué puede llegar a ser el ser humano?. ¿Qué nuevas culturas humanas se pueden sintetizar, construir y cómo se puede hacer de forma consensuada este proceso de cambios culturales?.

Denominamos "nueva ingeniería" a este conjunto de saberes distintos tanto a las ciencias como a las ingenierías tradicionales.

La nueva ingeniería está sólo en sus comienzos. Una parte de ella está naciendo

de la computer science, através de disciplinas como "knowledge engineering", "software engineering", "robotics". Otra rama arranca de de la "bioengineering" (Bugliarello, 1968) o de la ingeniería de materiales. Otra empieza a desarrollarse como ingeniería espacial (terraformación , colonización de Marte, construcción de biosferas artificiales). La más atrasada es la que parte de las ciencias sociales, aunque existe una creciente experiencia en el campo profesional de estas áreas. Simultáneamente, se está ya desarrollando una "design theory", que puer servir de cuerpo teórico inicial de esta nuevo sistema de conocimiento.

3. CMU EN TRANSICION: LIMITES DE LA INVESTIGACION EN "COMPUTER SCIENCE" Y LA

VUELTA A LA CULTURA INGENIERA.

Después de este recorrido por los distintas áreas de investigación de Computer Science en CMU, hemos concluído que el diseño puede ser el paradigma que unifique el sistema de conocimiento cultural que se ha generado en torno a la tecnología informática, y al resto de tecnologías. Pero ello no es más que una posibilidad plausible.

La situación actual en CMU es de cambio, y de planificación del futuro.

El 21 de Octubre de 1991 el nuevo presidente Mehrabian hablaba al profesorado del campus sobre " CMU's Challenge for the 21st Century", planteando como perspectiva ("vision"): "to be a university of the first rank by leadership and innovation through excellence and quality in our chosen fields of endeavor" (Focus, vol. 21, Nov/Dec 1991).

La nueva administración de carácter ingeniero indica que el paradigma de CMU como "computer-intensive campus" está en crisis, y ello es un reflejo de las dificultades de la "information society" norteamericana para afrontar los retos de la era que ha seguido al fin de la Guerra Fría.

Uno de los rasgos más importantes de la misma es la crisis del modelo de investigación, así como de sus temas tradicionales, que han dominado ésta durante 30 años. A modo de ejemplo citemos un artículo de Times, del 26 de agosto de 1991, titulado "Crisis in the Labs": "It was the glory of America. In the decades following World War II, U.S. science reigned supreme, earning the envy of the world with one stunning triumph after another...Now a sea change is occurring, and it does not bode well for researchers -or for the U.S...The message from Washington is clear: science will receive no more blank checks and

will be held increasingly accountable for both its performance and its behavior". (1991:45-49).

Esta crisis sacude a su vez las economías de las "research universities". Entre ellas, las más débiles son precisamente aquellas que más dependen del dinero de la investigación como CMU, el dinero que se llama "soft money", a diferencia del "hard money" o fondos procedentes de donaciones ("endowment") o matrículas de estudiantes como el que disponen las universidades tradicionales como Harvard o Yale.

Para CMU, la investigación es lo que le ha permitido adquirir niveles de excelencia. Las "research universities" necesitan fondos para seguir subsistiendo. Y estos fondos, al menos en CMU, se sabe que han de provenir en el futuro de más investigación no de menos, con una creciente atención a los que provienen de la educación.

Pero estos fondos federales ya vemos que ahora exigen que la investigación sea enfocada a resultados concretos y transferibles a la industria, según el modelo seguido, últimamente, por agencias como DARPA. Se abre paso en el país una clara conciencia que es preciso adoptar una "industrial policy".

La nueva administración de CMU no ha salido de la mentalidad de Computer Science sino de la ingeniería. Ante la crisis, esta institución ha realizado un primer movimiento de vuelta a sus orígenes ingenieros. Se plantea recuperar elementos valiosos del antiguo Carnegie Plan de Robert Doherty.

Este plan se centró en dar una educación al ingeniero lo suficientemente amplia, humanística y social, para prepararlo para afrontar los retos de una grave crisis nacional durante la época de la Depresión.

Tres cambios principales planteó la Administración Doherty: "First, a new philosophy and a new outlook which will comprehend the human and the social as well as the technical. Second, the development in all professional men of genuine competence in the professional way of thought -a way of thought which embodies an analytical and creative power that is as effective in the human and social realm as that developed in engineering in the application of the physical sciences. Then when they face a problem involving human and social elements either in relation to their regular professional work, or to their activity as citizens, or as individuals, they will deal effectively with the whole problem, not merely the technical part. Third, the development of the ability to learn from experience so that in the unfolding future they can continue to expand

their fundamental knowledge, deepen their understanding, and improve their power as professional men and women and as leading citizens" (Doherty, 1950: 2-3).

Fue esta reforma la que abrió paso a la "liberal education" en Carnegie Tech. Y fue este tipo de educación el que fortaleció en la institución la cultura universitaria dándole un carácter más científico al modo tradicional y menos aplicado. A cambio, aquella reforma trató de extender la mentalidad ingeniera, el entonces llamado "problem solving" o "engineering method" a los campos científicos. No obstante durante estas pasadas décadas la cultura predominante ha sido la científica no la ingeniera. Con todo ha sido una cultura científica muy particular, la "computer science culture", que tiene una irresoluble contradicción interna, al haberse construido sobre un invento tecnológico.

4. CMU COMO "DESIGN CULTURE": POSIBILIDADES REALIZABLES.

4.1. EL DISEÑO, CONOCIMIENTO BASICO DE LAS NUEVAS INGENIERIAS.

La contrapartida a la etapa en la que el "científico se ha hecho ingeniero, ha sido que el ingeniero se ha hecho académico, consiguiendo un status intelectual en la cultura actual, al menos en Estados Unidos, impensable antes de la II Guerra Mundial. A mediados de los 60s cuando los científicos sociales por primera vez eran admitidos de pleno derecho en la National Academy of Sciences, como Herbert Simon, se formaba la National Academy of Engineering, así como el Institute of Medicine, formando junto a la primera la denominada "Academy Complex", donde la primera ocupaba, y aún lo hace, el lugar de primus inter pares (H. Simon, 1991:333).

Este cambio es significativo. Tradicionalmente el ingeniero había estado empleado en la industria. A diferencia del abogado o el médico, no se consideraba una profesión "liberal", dado que dependía o de la industria, o antes de los militares. Su conversión en académico le va a permitir por primera vez tener libertad para iniciar la elaboración de un cuerpo teórico propio.

Ese mismo periodo coincide con las primeras elaboraciones sobre la posibilidad de una "science of design", nacidas en la ingeniería.

Una línea de investigación a seguir es el surgimiento de la "design culture" en CMU a partir de estas elaboraciones sobre el tema del "design", una de ellas llevada por el Design Research Center organizado por equipo profesores en 1974, entre los que se encontraba Steve Fenves, ingeniero civil. Este centro, hoy llamado EDRC, Engineering Design Research Center, fue un empresa interdisciplinaria con base en el colegio de ingeniería de CMU y dedicada al

"improvement of the quality and diversity of computer aids for all phases of engineering design and across all engineering and allied design disciplines" (Fenves, 1983:1).

Según Fenves la inspiración inicial del centro fue debida a Herbert Simon y su libro *The Sciences of the Artificial* (1969). En ese libro, hay ideas interesantes sobre el diseño como base de una cultura profesional general.

Pero lo cierto es que el interés por el tema del diseño como posible ciencia en CMU es anterior a esta obra. En 1967, el 19 de junio, Carnegie Mellon University sirvió de base para un Workshop de la Association of Engineering Colleges of Pennsylvania, dedicado a "to promote genuine interest in design and design education on the part of the faculty members and their associates". (*Engineering Design and Design Education*. R. Rothfus, 1967:9). Las disciplinas participantes fueron ingeniería química, mecánica, marina y diseño industrial. Ese mismo año se celebraron otros seminarios en Oklahoma State University, Stanford University, y la Universidad de Washington.

Estos seminarios habían empezado en 1965 en Berkeley, patrocinados por Commission on Engineering Education y con el apoyo de la National Science Foundation, interesado en educar a los estudiantes mediante "design problems".

Dos años más tarde se celebró la Fourth Conference on Engineering Design, ya con el lema: "Toward A Science of Design", celebrada en Dartmouth el 17 y 18 de julio de 1967. En el prólogo de sus Proceedings, R.C. Dean Jr. y B.R. Teare, Jr, este último decano de la escuela de Engineering and Science de CMU definían así el diseño: "...Design is the creative management of information to yield useful means for resolving man's needs and wants" (*Engineering Design...1967:i*), y por ciencia se entendía en términos muy amplios, "a generalized and structured body of knowledge". (*ib.:i*)

Algunas décadas después podemos comprobar que ciencia no es algo tan amplio sino un cuerpo ordenado y determinado de conocimientos organizado socialmente. Pero lo interesante es que esta definición de diseño nacida desde la ingeniería incluía según estos ingenieros las siguientes disciplinas dentro de estas "sciences of Design":

" We can distill from this from this statement that the Sciences of Design certainly include:

Creativity -psychology, sociology, physiology, information management (i.e. the science of man and information).

Management of information -information theory, decision theory, computer technology, operations research (i.e. the sciences of information).

Useful means -economics, finance, law, supply and demand, sociology, psychology (i.e. the sciences of economics and man).

Needs and wants -identification, specification, information, values, ethics, sociology, psychology (i.e. the sciences of information and man).

This analysis leads us to two of the sciences of Design:

The science of man

The science of information." (1967: i)

No se citaba la antropología, entonces inexistente en CMU. Finalmente acababan los autores incluyendo la ciencia del mundo físico también: "Yet to realize our goal, that is 'to yield means', we need all the natural sciences too. So we have in the end as the sciences of Design:

The science of man

The science of information

The sciences of the physical world. " (ib: ii).

Estas afirmaciones se hacían en una conferencia de ingenieros. Estos se orientaban en dirección inversa a la de algunos científicos que se pasaban a la ingeniería como Von Neumann o Herbert Simon. Eran ingenieros a la búsqueda de un cuerpo teórico de conocimientos, incluyendo valores, que en torno al diseño hicieran avanzar la "engineering education". A este conocimiento le llamaban ciencia. Durante estas pasadas décadas se ha considerado que la ciencia podía servir a este propósito.

Es preferible ahora denominar este nuevo sistema de conocimiento "design", como ingeniería teórica o nueva ingeniería, para distinguirla de la antigua ingeniería como mera aplicación de la ciencia, en lugar de seguir confundiendo dándole el nombre de "ciencia del diseño".

Esta línea es básica para elaborar las bases de una tecnocultura.

Pero aún más importante es unir a estas ingenierías de diseño, su correspondiente en el campo social y cultural. De las escuelas profesionales de ciencia política, de administración, vemos preciso elaborar disciplinas basadas en el diseño a distintos niveles, de organizaciones, de empresas, de sistemas sociales donde puedan desarrollarse las anteriores tecnologías. Por último, vemos posible un antropología profesional que, con un pie en la academia y otro en el resto de la sociedad, vaya elaborando los programas culturales de este

nuevo tipo de comunidades. Quizá la tarea más delicada y compleja de las tecnoculturas.

Las tecnoculturas han empezado a elaborar valores propios: la "innovation", la "synthesis", un enfoque sistémico e interdisciplinar para la solución de problemas complejos, y una colaboración con disciplinas humanísticas y sociales. Nacen con ellas nuevas profesiones con sus códigos de valores propios que han de ser investigados con más detenimiento. En cualquier caso, la cultura profesional, a diferencia de la científica, no puede escaparse al diseño de su propio código de valores, pues en ella, los juicios de hecho son un resultado de los objetivos que persiguen, en suma, de los valores tanto de los profesionales como de la sociedad en la que trabajan y para la que trabajan.

Hasta ahora los estudios se han centrado sobre los valores que los nuevos tecnólogos "deben" respetar, la ética que han de obedecer. Los límites que sus investigaciones han de tener. Pero son escasos los estudios sobre los nuevos valores de innovación que estos mismos se plantean en sus investigaciones y práctica profesional y que ofrecen al resto de la sociedad como elementos para su renovación cultural. La línea que me parece más interesante es la segunda, aquella que se dedique a estudiar los valores nuevos que están surgiendo o se pueden diseñar, dentro de las nuevas tecnoculturas, más que los que la sociedad actual pretende imponer a las "nuevas tecnologías". Por otra parte, estos llamados valores de la "sociedad" son a su vez valores también diseñables y cambiables.

Ello nos conduce a una nueva relación entre la llamada "liberal" y la "professional culture", entre el valor denominado "libertad", y el valor "innovación" o diseño tal como lo hemos visto en esta cultura tecnológica. Hasta ahora una sociedad tecnológica era vista como una sociedad de control burocrático del Estado sobre el individuo y su libertad. Tecnocracia era sinónimo de burocracia.

Pero la burocracia de más interés en una "information society" es la del conocimiento, dado que ésta es la que legitima a la burocracia administrativa. Y la burocracia del conocimiento hasta ahora ha tenido en las universidades su principal lugar de empleo, y en la ciencia su sistema de conocimiento básico. Fue el sistema de conocimiento científico el que, frente al sistema tutelado por la religión, hizo suyo el valor de la libertad, pasando a considerar la "liberal education" como el sistema de educación de la persona educada científicamente.

Pero al mismo tiempo, el propio sistema de la ciencia, se ha conducido mediante un pensamiento determinista, difícilmente conciliable con el anterior objetivo. ¿Que ha entendido por "libertad" hasta ahora las comunidades científicas. Ha entendido lo que la ciencia como "conciencia de la necesidad". Pero hay otra acepción de la misma: Libertad como invención. Algunos autores como S. Giner así la definen. (1987).

En este segundo sentido, tecnología y libertad no serían términos antagónicos sino concordantes, el campo común de las posibilidades humanas. La cultura liberal sería aquella propia de los profesionales que ejercen libremente su profesión. Si antes eran los ingenieros los que precisaron de conocimientos científicos, son ahora los científicos los que tienden a hacerse ingenieros y profesionales, en suma, a adoptar una cultura de la innovación de la invención,, lo cual conlleva una menor necesidad de protección por la Administración y el Estado. Mayor asunción de riesgos y de oportunidades, ligada a una mayor competitividad intelectual, nos lleva a un nuevo tipo de innovaciones e invenciones consensuadas, pactadas socialmente.

La libertad en las tecnoculturas sería una libertad de y para la invención, incluyendo los riesgos de la invención en las nuevas tecnologías y las estructuras sociales que comporta. Estas invenciones son obras humanas y sus autores son responsables de las mismas, en función del uso para el que sus autores las construyen.

4.2. DE LA "RESEARCH UNIVERSITY" A "GENERAL DESIGN INSTITUTIONS".

Una segunda línea de investigación hace referencia los cambios posibles dentro de las propias instituciones de conocimiento como centros de investigación y educación, en particular en CMU.

Estos últimos 30 años han sido la época dorada de la "Science and Technology" en Estados Unidos. A su vez, este período ha coincidido con el de expansión de la "Computer Science" en esta institución, y de aproximación al modelo clásico de la cultura universitaria. El computer scientist ha desplazado en investigación al antiguo ingeniero en esta research university, a costa de hacerse él mismo diseñador. Este modelo del "scientist of design", que a su vez, ha sido el de la "computer science", tiene sus límites.

El principal es frenar una real teoría del diseño, como un nuevo sistema de conocimiento cultural distinto, independiente y más complejo que la propia cultura científica. Esa es la otra cara de la reforma de la ciencia convertida

en diseñadora. Impide que el diseño se convierta en una teoría ingeniera independiente.

CMU es una "research university". La generación educada bajo el Plan Doherty como Herbert Simon ha tratado de hacer científicos (lo que es sinónimo aquí de cultura "liberal") a los ingenieros. Esa batalla histórica ha conseguido convertir este instituto tecnológico en una "national university". Esta empresa ha correspondido al periodo de transformación del país de una nación de ingenieros e industriales con horizontes locales en una potencia científica, tecnológica, y militar, líder a escala internacional, que ha sustituido durante las pasadas décadas al imperio británico y las antiguas potencias europeas en la dirección de la civilización occidental.

Hoy este modelo cultural ha acabado agotando las fuerzas de esta nación. Por un lado, ha empleado sus mejores cerebros inventivos en la construcción de una gigantesca maquinaria defensiva, que ahora está reconvirtiéndose, y por otro, ha hecho a Estados Unidos la nación responsable de la defensa de los valores de una civilización que muestra síntomas de agotamiento, y el primero de ellos el sistema de la ciencia, inmovilizando la renovación de su sistema de valores y haciéndole incluso relegar sus aportaciones culturales más específicas, como la idea de tecnología ligada al espíritu inventivo y pionero de sus fundadores, en suma, el constituirse como una "republic of technology" en palabras de D. Boorstein (1978), un laboratorio de inventos de la humanidad.

Ahora el país, y CMU, busca recuperar, un conjunto de valores, de objetivos nacionales, que le permitan salir de su actual crisis que ya es económica, educacional, de servicios sociales, de sus sistemas urbanos, de sus relaciones étnicas, en suma, general.

La tendencia que apunta ahora es una tendencia inversa a la de los últimos 30 años, no hacer los institutos de tecnología universidades sino el convertir a las "national universities" en "institutes of technologies" de un nuevo tipo, en el sentido que orienten su investigación en función de sacar al país de la crisis actual. Las agencias federales presionan ahora para que se vuelva a un tipo de investigación que esté al servicio de las necesidades del país, de su economía y de aumentar el nivel educativo de sus ciudadanos. En suma, se quiere más "applied research" orientada al diseño y la producción y menos ciencia pura. Y como europeos nos interesa colaborar en este esfuerzo. Con Estados Unidos puede ocurrir, en los próximos años, lo que está ocurriendo con Rusia ahora.

Europa y Japón están especialmente interesadas en colaborar con estas naciones para que salgan de la presente crisis, y ello, en primer lugar, por su propio interés económico, y en general, cultural. La construcción de un nuevo modelo cultural, de un sistema complejo de tecnoculturas, no es ya tarea de un sólo país. Y en esa tarea los Estados Unidos juegan un papel decisivo. Su crisis afecta al conjunto de la comunidad internacional.

Un modelo que propongo considerar para la transformación de estas magníficas "research universities" como CMU en instituciones "mission oriented", formando dentro de ellas, un nuevo tipo de institución que podríamos denominar: "General Design Institutes".

Un tipo de instituciones de investigación superior interdisciplinarias, orientadas a la investigación de las tecnoculturas, en un sentido amplio, con el conjunto de cambios en los valores, instituciones sociales y nuevas tecnologías que ellas pueden conllevar, a partir de problemas actuales, como pueden ser el de la crisis industrial, el deterioro urbano, o la crisis educativa. Un tipo de institución parecida ya fue planteada por Harold Lasswell a principios de los 60s, denominándolos "centros superiores de ciencia política" : "El ambito de un centro de ciencia política ...destacaría sobre todo los objetivos normativos del hombre o el futuro de un amplio contexto colectivo." (1971:217).

Estos centros de diseño de tecnoculturas podrían relacionarse con la visión futurística planteada por Dana Scott de CMU: " The University should not minimize the fact that by creating a campus-wide integrated computing environment it is becoming a microcosm of Techonopolis, the information city of the future" (Scott, Search for the President, 1989:33).

Esta propuesta se basa en elementos que ya están en germen en algunas de estas instituciones , como CMU. La idea central de las mismas es que el diseño, la invención, no es un medio, es un valor cultural en sí mismo y puede extenderse al conjunto de campos de estas universidades. Una de sus consecuencias es que puede dar lugar incluso a un nuevo tipo de máquina, no basada en el cómputo o en el procesamiento de símbolos, sino en la tarea de diseño e invención del conjunto de elementos de un sistema cultural.

La "research university" o los organismos científicos de investigación precisan un nuevo tipo de institución, unos "general design institutes" que se planten afrontar este tipo de problemas.

4.3. NUEVOS SISTEMAS TECNOLOGICOS: DEL "COMPUTER" Y LA "TELEVISION" A

SISTEMAS

DE "DESIGN MACHINES".

Una lección que aprende el antropólogo en una cultura tecnológica es que toda propuesta de reforma de conocimientos o instituciones que no termine en un nuevo tipo de investigación sobre un nuevo artefacto mecánico, tiene un valor escaso, a los ojos de esta cultura.

En CMU existen ya intentos de construir un tipo de nueva máquina que supere las limitaciones del ordenador, en tanto que máquina más acabada de las logoculturas, cada vez más empleada para objetivos distintos a los iniciales, pero sin cambiar su definición inicial, máquinas que integren video, ordenador y comunicaciones.

La vida cotidiana del ser humano en países industrializados, y también en CMU, se reparte entre tres o cuatro tipos de máquinas fundamentales: las de trabajar, crecientemente, ordenadores, las de descansar, normalmente, el televisor doméstico, y las de comunicarse, teléfono, y trasladarse de la residencia al lugar del trabajo, el automóvil. Las primeras le permiten realizar las tareas de producción económica e industrial tradicionales, las investigaciones científico-técnicas, los negocios, los cálculos y las comunicaciones formalizadas. Las segundas, le permiten consumir imágenes realmente creativas de mundos idealizados, en forma preferente películas, dibujos animados para los niños, o anuncios publicitarios. Ordenador y televisor son tecnologías que canalizan principalmente modelos de conocimiento que podríamos denominar logo y mitoculturales respectivamente., aunque no de forma exclusiva.

El ordenador tiende a utilizarse para fines muy distintos de los iniciales, centrados en el cálculo. Por ejemplo, los ingenieros hoy día, utilizan el ordenador para hacer diseño asistido, o CAD (Computer Aided Design). Las redes informáticas se utilizan para la comunicación personal. Por su parte, la televisión se utiliza para transmitir noticias y documentales científicos, y tiende a hacerse interactiva. Con todo, computador y televisor no son máquinas inventadas especialmente para diseñar, sino para calcular, observar, y entretenerse, y para controlar.

Los ingenieros e inventores que las han construido lo hicieron para fines ajenos a la actividad de diseño, propia de la ingeniería. Esto no es nuevo. Así se inventó también la máquina de vapor o el motor de explosión. Ciertamente es que a diferencia de esas máquinas, el ordenador y la televisión manipulan elementos

culturales directamente relacionados con el sistema de conocimiento humano. No obstante aún está por inventar un tipo de máquina, un sistema tecnológico, que sirva preferentemente para inventar, para diseñar, y hacer viables estos diseños, tanto en los campos tecnológicos como de diverso tipo

Estas máquinas de diseño o "design machines" , propias de las tecnoculturas, pueden ser artefactos o sistemas de artefactos más acordes con un tipo de cultura donde el ser humano tenga más posibilidades de inventar sistemas reales tanto sociales como tecnológicos, no sólo sus simulaciones.

Una antropología de las tecnoculturas, para su diseño y estudio, precisa también colaborar con los nuevos ingenieros en el diseño de este nuevo tipo de máquinas. Igual que se le pide al ingeniero que participe y comprenda que es un sistema cultural. Es hora que los antropólogos y demás científicos sociales pasen a colaborar en el diseño de sistemas tecnológicos, como parte del diseño de nuevos sistemas culturales complejos.

Estas son algunas de las posibilidades plausibles que la antropología puede abordar en las próximas décadas.

Como epílogo, sólo destacar que ésta investigación nacida y patrocinada por instituciones de Catalunya, pueda servir para el diseño de políticas y actuaciones avanzadas en esta comunidad y más allá de la misma en la propia comunidad europea que está diseñando en estos momentos un complejo proceso de integración y cambio cultural.

Arturo Serra.

Pittsburgh. 3 de mayo de 1992.

PARTE QUINTA.

METODOLOGIA DE LA INVESTIGACION Y BIBLIOGRAFIA.

1. METODOLOGIA DE LA INVESTIGACION.

1.1. EL TRABAJO DE CAMPO, COMO PREMISA DEL ESTUDIO.

La técnica que en general se acepta como la más propia de estudio antropológico es el trabajo de campo o "fieldwork".

Este estudio se ha realizado siguiendo esta tecnología de base. El investigador ha residido en la comunidad de Carnegie Mellon dos años y cinco meses, desde febrero de 1990 a la finalización del estudio en mayo de 1992. Excluyendo los viajes realizados a España, el periodo neto de permanencia en dicha comunidad ha sido de 21 meses (seis meses en 1990, 10 meses en 1991, y 5 meses en 1992.).

Durante ese periodo, he dispuesto de tres despachos distintos situados, el primero de ellos en el Engineering Research Design Center, el segundo, en el Robotics Institute, centro situado en Wean Hall, el edificio principal de la School of Computer Science, donde he trabajado la mayor parte del tiempo. Y el tercer despacho, en el Information Technology Center, otro centro de dicha escuela. Durante ese tiempo he asistido regularmente a aquellas actividades como charlas, seminarios, reuniones de proyectos, cursos, y actividades de la comunidad que tenían relación con el objeto de estudio.

Pero la presencia del investigador en el terreno, siendo una premisa necesaria, no es suficiente. El trabajo de campo precisa de un enfoque sistemático para abordarlo.

1.2. WERNER & SCHOEPFLE: "SYSTEMATIC FIELDWORK".

La obra de referencia metodológica básica que he utilizado ha sido la de Oswald Werner y G.Mark Schoepfle, "Systematic Fieldwork" (1987), dos volúmenes de 400 páginas cada uno, que recogen un completo recuento de técnicas propias de una antropología orientada al estudio del sistema de conocimiento de un grupo social determinado, de lo que los autores denominan su "cultural knowledge" (vol I: 20). Ha sido particularmente sugerente su idea de concebir la etnografía como la construcción de lo que la Inteligencia Artificial denomina un sistema experto: "An ethnography can be seen as an expert system -a data base to which anyone interested in the lives of the natives can refer". (1987, vol II: 23). Idea que he encontrado a su vez en la colega Diana Forsythe de la University of Pittsburgh.

Los dos principales puntos de partida de estos autores son la importancia de una epistemología que explote la anomalía o "anomaly" (vol 1: 22), y en segundo lugar que se base en los terrenos semánticos y terminológicos,

"lexical/semantic fields" (ib:22). Este enfoque ha sido básico para detectar aspectos de esta comunidad tales como la diferencia de la cultura tecnológica y científica de CMU y su relación con respecto al mundo universitario tradicional, el significado que conceptos como "problem-solving", "design", "artificial intelligence", "heuristic search", "information processing", y otros más tradicionales como "inferential programming", o "computer science" en esta comunidad.

Con todo este enfoque es insuficiente para entender el contexto social más amplio en que estos significados culturales están insertos, así como las relaciones organizadas de los mismos. La cultura no es sólo conocimiento sino también organización, y la "computer science culture" de CMU no es una excepción. Ello ha hecho preciso otros autores y distintas metodologías.

Otros estudios metodológicos usados han sido los de H. Russell Bernard, "Research Methods in Cultural Anthropology" (1988), Erve Chambers, "Applied Anthropology" (1985), y John Van Willigen, y Timothy L. Finan, "Soundings: Rapid and Reliable Research Methods for Practicing Anthropologists" (1991).

1.3. ELABORACION DE HIPOTESIS COMO TECNICA DE INVESTIGACION.

Tradicionalmente, la metodología del trabajo de campo se divide en dos grandes apartados, la recogida de datos, y el análisis de los mismos. Pero lo más difícil de un estudio etnográfico, según la experiencia del estudio realizado, es un punto previo a estas técnicas al que no se le dedica atención en dichos trabajos metodológicos: el diseño de las hipótesis que el estudio tratará de probar, mediante dichas técnicas. Este diseño ha de concebirse también como una tecnología a desarrollar, a aprender.

Las técnicas de la etnografía variarán en función de dichas hipótesis. Si de lo que se trata es de construir un mapa sobre el sistema de conocimiento cultural de la comunidad dada, hay que partir de una ideas previas acerca de cuál puede ser un primer esbozo de ese mapa, dado que ese objetivo guiará la búsqueda. En mi caso, la búsqueda parte de la constatación de las limitaciones de los marcos de referencia culturales en presencia, tanto en el caso de las sociedades industrializadas como en las denominadas primitivas, y la apertura hacia un diseño de nuevos modos de cultura y de conocimiento organizados.

Una pista inicial de este estudio fue una traducción al castellano del libro de Herbert Simon, "Las ciencias de lo artificial" (1973), donde se introducía la figura del "designer". Otra pista fue el folleto de J. J. Servan Schreiber

titulado "La Revolución de Conocimiento" (1986). Estos libros vinieron a confirmar la idea que estaba surgiendo un nuevo sistema de conocimiento distinto a la ciencia, y que yo denominaba "saber inventivo" en 1989.

Pero el trabajo de campo también sirvió para cambiar la idea inicial acerca del interés principal de Herbert Simon. Este no estaba en el diseño sino en la clásica ciencia experimental, usando el diseño como complemento de ésta. Así las hipótesis iniciales se ven luego cambiadas a lo largo del trabajo de campo, pero sin ellas, el antropólogo no sabe que datos seleccionar de la montaña de información de la que puede disponer, y más aún en las llamadas "information societies".

El diseño de las hipótesis va unido al recorte del objeto de estudio, tema que tampoco se le da suficiente importancia en los estudios metodológicos pero que es el que permite finalizar un estudio que podría ser interminable, y acota las hipótesis. Así, durante casi un año, la idea inicial fue hacer un estudio sobre toda la comunidad de Carnegie Mellon como sistema de conocimiento centrado en el diseño, luego fui centrándolo en las áreas de investigación, hasta quedarme con la que me parecía hoy la principal, Computer Science.

El recorte del objeto de estudio es tanto más importante en antropología dada la complejidad de una disciplina que pretende abordar los fenómenos humanos desde un enfoque denominado "holista" o sistémico. Cuando conocen las fuerzas que se emplean en CMU, o en cualquier otra universidad de investigación seria, para el estudio y diseño de un sistema tecnológico elemental, se comprende que la ciencia social está aún muy lejos de estos niveles de complejidad en la organización de su propio trabajo, y que el diseño cultural sólo es posible a escala experimental, demostrativa, y muy modesta. En todo caso, cuanto más compleja es la comunidad analizada menos holista puede ser el enfoque del antropólogo, a menos que disponga de un ejército de colaboradores y medios económicos abundantes, lo que de momento no es el caso. Más ha de centrarse en aspectos cualitativos centrales de dicha comunidad. Ha de seleccionar de forma sistemática.

Una vez recortado el campo (proyectos de investigación en Computer Science de CMU), y formuladas las hipótesis, las técnicas se pusieron en consonancia.

1.4. EL INFORMANTE CLAVE Y EL ENFOQUE DENOMINADO "STUDYING UP".

A lo largo del estudio he realizado 104 entrevistas personales a distintos miembros de la comunidad de Carnegie Mellon, y en particular, a distintos

profesores de Computer Science o relacionados con esta área. Conforme fui recortando el objeto de estudio las entrevistas fueron focalizándose en Computer Science.

En cualquier caso, he podido comprobar que en efecto, el papel del informante clave y las entrevistas al mismo es, como sostiene Van Willigen (1991:1) uno de las dos técnicas claves de la antropología.

A una comunidad como Carnegie Mellon es difícil acceder si no se cuenta con el llamado "informante clave" ("key informant"). En este caso, este papel lo ha jugado el Dr. Angel Jordan sin cuya colaboración y asesoramiento, las propias entrevistas personales, al menos las claves hubieran sido imposibles.

Este informante clave tiene dos particularidades, primera que es un "university profesor" en Electrical and Computer Engineering, lo que le permite estar a caballo entre la ingeniería y la computer science, y segunda, que ha ocupado el cargo de Provost en dicha comunidad durante ocho años, lo cual significa que posee conocimiento extenso sobre el conjunto de la comunidad y sus características. Su elección como informante clave responde a una orientación elegida para hacer esta etnografía, estudiar la comunidad desde sus principales líderes intelectuales y las personas que ejercen la mayor influencia en la comunidad. A este tipo de enfoque lo denominó la antropóloga Laura Nader, "studying up" (1974: 284-311) .

Este enfoque tiene sus ventajas e inconvenientes. Permite una visión de la comunidad desde la jerarquía, desde las personas que dirigen intelectual y administrativamente dicha cultura, pero no resulta un método exhaustivo.

Prioriza la calidad sobre la cantidad, y dado que la jerarquía en una organización es donde se producen los mayores conflictos y rivalidades, puede dar al antropólogo una visión sólo de una parte de la comunidad, y no del conjunto de la misma. En cualquier caso, es el método de información escogido en esta investigación.

El Dr. Jordan no ha sido, no obstante, el único informante del estudio. He contado con la colaboración de otras personas claves que me han proporcionado información valiosa tanto sobre Computer Science como sobre en general la comunidad de CMU. Herbert Simon, Allen Newell, Richard Cyert, y otros líderes de la institución, proporcionaron una valiosa información sobre la misma. Así como Preston Covey del CDEC , y las antropólogas de CMU, Barbara Lazarus y Judith Modell.

Dentro de la School of Computer Science he contado con la colaboración de informantes claves, como Catherine Copetas, Assitent Dean for Development, Karl Love, Project Manager del proyecto NECTAR, Mary Thompson, Project Manager del proyecto MACH, David Pahnos, Associate Director del Field Robotics Center. Así mismo han mostrado una colaboración Sharon Burks, Assistant Dean de la SCS, y Maggy Muller, Assitant Business Manager.

1.5. LAS ENTREVISTAS ETNOGRAFICAS ABIERTAS.

Las 105 entrevistas realizadas han sido de las que tengo referencias, y transcrito como textos etnográficos del tipo (2) (Werner&Schoepfle: vol 1, 73). Según estos autores, los textos que construye el antropólogo se pueden dividir en dos tipos: los más tradicionales son las notas de campo nacidas de la tradicional observación participante, método que se ha considerado desde Malinowsky como un método fundamental del trabajo de campo. En ellas, el antropólogo describe los comportamientos observados en la comunidad. A este tipo de textos, Werner y Schoepfle lo denomina "journal" "text (1)" o "journal". Pero al estar estos mezclados con descripciones subjetivas del antropólogo se convierten según estos autores en ambiguos.

Un tipo de textos etnográfico que recoge la visión "emic" del nativo de forma más rigurosa son las que esos autores llaman "text 2", esto es, las transcripciones de los propios discursos de los nativos o "transcriptions" o "text (2)": "Only TEXT (2) remains unambiguously ethnographic data" (ib.: 73). Con todo, la selección de estos discursos la realiza el antropólogo, pero al menos ha de atenerse a los propios términos de los nativos estudiados.

Con todo el resultado final de la etnografía ha tratado de ser la visión estereoscópica o binocular resultante de la visión del antropólogo y la del nativo a la que se refería Kluckhohn.

Siguiendo esta idea el primer método utilizado fue la clásica entrevista abierta ("open interview").

Una experiencia observada es que conforme más sofisticada intelectualmente es la persona entrevistada he comprobado que menos se presta a cuestionarios rígidos, esquemáticos y simples. La conversación, más o menos, guiada por el antropólogo resulta, en este tipo de medio cultural, más útil para recabar la información necesaria. Las mejores entrevistas fueron las conversaciones donde el entrevistado mostraba interés en el estudio del antropólogo. Las más pobres, aquellas donde el profesor debía de jugar el papel de mero "objeto" de estudio.

No obstante, las entrevistas tienen un límite en un medio tecnológico dado el escaso tiempo con que cuentan estos profesores y el propio antropólogo. Por otra parte, la transcripción de las mismas, de la cinta magnetofónica al papel consume excesivo tiempo, dada la tecnología actual. (Incluso los sistemas de reconocimiento de voz por medio de ordenador en los que CMU es puntera y, que exploré a fin de ahorrar tiempo, no están tan avanzados como para que el ordenador pueda recoger una conversación y la transcriba impresa en papel). Paralelamente a las entrevistas, decidí optar por la recogida de información directa de la propia comunidad, de textos escritos donde se pudiera recoger los sistemas de conocimiento de los "computer scientists".

Esta es una cultura no de palabras sino de proyectos, y los proyectos están escritos. Si la cultura está en el lenguaje, la cultura científica y cada vez más la tecnológica está en textos, en lenguaje escrito, en libros, artículos, revistas de dentro y fuera de la comunidad de CMU, en archivos, en "memos". Tras las primeras entrevistas, lo que se impuso fue centrarme en lo que B. Colby una "text ethnography"(1979) o etnografía de textos, y en el análisis de los mismos. El problema era encontrar qué tipo de textos podían recoger mejor el sistema de conocimiento de dicha comunidad.

1.6. UNA NUEVA TECNICA : "ANALISIS DE PROYECTOS"

Ello me llevó a inventar una nueva técnica , la de "análisis de proyectos". La técnica que ha unificado finalmente todo el estudio ha sido la que denomino "análisis de proyectos", y se basa en el análisis de las llamadas "proposals" o documentos que presentan periódicamente los investigadores para recabar fondos donde se plantean los objetivos, los puntos de partida , los resultados de la investigación anterior. Es un documento básico para el estudio de comunidades de investigadores, en especial para aquellas que han de someterse a la disciplina de una investigación denominada "mission oriented".

Este estudio se centra en un análisis de determinados proyectos de investigación de una institución universitaria de investigación básica : la SCS, y de las personas que los llevan a cabo. Es el propio proceso de investigación del computer scientists lo que se pretende analizar como fenómeno cultural.

La forma en que llegué a esta técnica fue al observar que CMU era una universidad que se empezó a regir bajo la Administración Cyert por medio del llamado "strategic planning". Como comunidad tecnológica, los planes forman parte de su propia naturaleza, y está me llevó a explorar cuáles eran los planes

de la School of Computer Science. Si su clave era la investigación tecnológica basada en el diseño, esta no se improvisa. Era una comunidad planificada, y sus investigaciones también lo eran. Una comunidad de diseñadores acaba diseñando su propio sistema de conocimientos, y esto me llevó a interesarme acerca de cómo tenían organizado su sistema de conocimiento clave, su sistema de investigación. Y un punto clave de las mismas son las denominadas "proposals".

Este método de los proyectos de investigación nació en el medio tecnológico y más tarde se extendió a la ciencia. La ciencia se regía tradicionalmente por las llamadas "líneas de investigación" de los departamentos abiertas por catedráticos y profesores titulares. Ahora ya es común en la ciencia presentar "proyectos de investigación". Pero su estructura sigue siendo una anomalía en el sistema de conocimiento que se rige por objetos de estudio no por "objetivos" o como dicen las propuestas de la SCS, por "claims", término tomado de las patentes de los inventores y que designan el invento ("claim") que quieren reclamar protección para el mismo.

El trabajo realizado es incompleto. No he tenido tiempo para estudiar todo el proceso de invención, desde la concepción del sistema, su realización y su finalización. Pero creo que la guía de seguir las "proposals" permite al antropólogo una vía de acceso al programa cultural del investigador de una "high tech", esto es, a su sistema de conocimiento organizado. Las "proposals", al menos las más representativas, son obras colectivas y reflejan un sistema de saber comunitario. El conjunto de estas investigaciones es lo que forma la clave del sistema de conocimiento de un colectivo como la SCS, que podríamos resumirlos en el conjunto de planes de investigación de sus profesores y estudiantes.

Esta técnica pienso que se puede utilizar a su vez para el estudio de otros campos, como biotecnología e incluso la propia antropología. Es muy interesante la iniciativa en este sentido de Van Willigen, que está organizando un centro en la University of Kentucky que reúne todos los proyectos realizados por los antropólogos. En CMU no existe hoy por hoy una memoria organizada de los proyectos realizados por la institución en Computer Science.

Esta técnica podríamos compararla, en el campo de las tecnoculturas, con la técnica de estudio del sistema de mitos de una comunidad primitiva, o mitoculturas, para la comprensión de su sistema de conocimientos organizado, línea que ya inició Levi Strauss con sus Mitológicas, y que a pesar de las

críticas recibidas, aún queda como una obra clave de la antropología contemporánea.

1.7. TEXTOS INTERNOS Y EXTERNOS A LOS PROYECTOS.

Los proyectos tienen como documentos básicos las denominadas "proposals", pero existen otros muchos textos necesarios para abordar un análisis de éstas, sin los cuales el antropólogo no entenderá nada acerca de ellas. Las "proposals" viven en un contexto de textos culturales más amplio.

Estos se dividen en textos de la propia comunidad de investigadores y textos exteriores a la misma, provenientes de la cultura general de la institución donde está el "computer scientist", esto es, CMU, de su propia comunidad de investigación más amplia, la comunidad nacional de "computer science", de las instituciones federales de donde obtienen los fondos y sus propios programas de investigación. A su vez son interesantes los textos de instituciones representativas de la comunidad científica y tecnológica nacional.

-DOCUMENTOS INTERNOS DE LOS PROYECTOS.

Una ventaja del trabajo antropología en una institución universitaria es la existencia de documentos escritos con los que se puede contar para realizar un análisis de contenidos.

A diferencia de un laboratorio industrial, donde a fin de no facilitar el terreno a la competencia, los documentos son en su mayoría reservados, en un centro de investigación universitario como la SCS, la publicidad es mayor.

Incluso, en un centro como este financiado con fondos, principalmente, del Departamento de Defensa.

La investigación está escrita. Y se da a conocer de diversa forma: informes internos ("reports"), artículos en revistas científicas, libros, tesis doctorales ("doctoral dissertations"), y eventualmente, entrevistas en los periódicos y revistas locales. Los departamentos como el de Computer Science de CMU publican regularmente artículos de sus propios investigadores en forma de "papers" relacionados con los proyectos en curso en los que trabaja el investigador y que circulan primero internamente y luego pasan a ser presentados en congresos y reuniones de dichos investigadores. Estos han jugado un lugar importante en este estudio.

Pero los documentos más interesantes son los del propio proyecto aquellos que describen los objetivos, ("goals, desiderata") las especificaciones, el calendario, el presupuesto, la organización y la localización del mismo. El

básico es el que se denomina "proposal", donde se describen los temas antes citados. Pero del proyecto inicial al resultado final pueden haber cambios sustanciales. Los "memos" de las reuniones del equipo son interesantes como documentos que permiten comprobar los cambios en el proyecto. Pero, si este ha sido comprometido con una institución como DARPA, estos cambios no suelen ser radicales, manteniendo lo sustancial de la propuesta inicial, que ya se realiza con la suficiente seguridad de que lo que se ve posible construir, en efecto, lo será.

-DOCUMENTOS EXTERNOS AL PROYECTO.

Ha sido de esencial ayuda al estudio la revista FOCUS, publicada en CMU por y para el profesorado de esta universidad. En ella se han encontrado artículos y puntos de vista de profesores en momentos decisivos de la institución, así como debates interesantes. La revista TARTAN de los estudiantes ha jugado un papel menor.

Respecto a la comunidad de Computer Scientists norteamericano una revista importante es Communications of the Association for Computing Machinery. En relación con el Departamento de Defensa, ha sido útiles el documento de la Strategic Computing Initiative y otros documentos publicados por DARPA como resultado de las reuniones que organiza. Así existe una documentación abundante y poco explotada que editan una variada serie de instituciones académicas como la National Science Foundation, la National Academy of Engineering y la National Academy of Science que proporcionan una visión colectiva de la comunidad de la ciencia sobre temas relevantes de actualidad.

1.8. LA OBSERVACION PARTICIPANTE ELECTRONICA.

Esta técnica ha jugado un papel escaso en este estudio. Los investigadores trabajan en sus despachos de forma individual la mayor parte del tiempo, a excepción de las reuniones semanales que tienen. He asistido en la gestación del proyecto de investigación High Definition Television, y a una primera demostración del mismo ante un representante de DARPA.

Una observación que sí he realizado repetidamente es la observación electrónica de las interacciones en la red ANDREW. Esta red es un lugar privilegiado para observar los debates, las opiniones y las colaboraciones que se establecen en una comunidad de Computer Scientists. Las interacciones observables son las que aparecen en forma de "bulletin boards" o boletines electrónicos. En la School of Computer Science, principalmente, y también en el resto de CMU, el real foro

de discusión político es la red, viene a representar lo que puede ser la plaza del pueblo en un medio campesino tradicional. Un profesor de filosofía de CMU, Robert Cavalier, habla del "electronic agora". Y en efecto lo es.

En la red apareció la encuesta de Mary Shaw , "Great Ideas in Computer Science". Se pasan a discutir los temas de actualidad más variados (Guerra del golfo, huelga de transporte publico de Pittsburgh, actos racistas en Stanford University. Se proporciona información sobre la vida de la comunidad, ausencias de profesores, visitas de otros, charlas públicas. Realmente, la comunidad de Computer Science sea de las más politizadas del campus, la que tiene un mayor interés por estos temas.

No obstante, y a pesar de ser interesante, este estudio no ha podido explotar a fondo esta nueva técnica de observación de una comunidad mediante sus redes de ordenadores. Pero en un futuro puede explotarse como via de trabajo etnográfico, y al parecer ya se está empezando a hacerlo.

1.9. ANALISIS DE LOS DATOS: LAS SECUENCIAS DE FINES Y MEDIOS.

La metodología principal empleada para el estudio de los citados proyectos ha sido establecer la relación entre los fines y los medios en el mismo. Igualmente esta pregunta ha servido para aclarar la relación en Carnegie Mellon entre cultura tecnológica y cultura científica.

Me ha interesado sobre todo comprender la relación de metas de finalidad, en particular, la relación invención-descubrimiento en la actividad del computer scientists de CMU, expresada en la construcción de su propio proyecto de investigación. Que se persigue con el ?¿ Qué objetivos se pretenden cubrir?¿ ¿Qué subobjetivos se marca?¿ ¿Cómo los cubre?¿.

Si la actividad principal de personas como H.T. Kung, o R. Rashid, es la investigación, lo primero ha sido preguntar por los propósitos de la investigación, dado que estos dan significado al resto de sus estrategias para conseguirlos.

El método de "cadenas causales" B.N. Colby (1975, cit Werner & Schoepfle, vol 2, 187,ss) desarrolló sirvieron para conocer como una cultura como la maya explicaba sus fenomenos, a que atribuía las causas, el por qué de sus acciones, pero no el para qué (what for). Por otra parte, el "análisis de protocolos"(J. R. Hayes. 1989) o "thinking aloud" elaborados por la sicología cognitiva, sirven para conocer el cómo, (how), un individuo esta resolviendo un problema pero no indica el para qué lo hace. Lo mismo ocurre con las "flow chart" (H. Russell

Bernard. 1988), mapas causales o planes , inspirados en el pensamiento algorítmico. Solo indican una relación causal, "a" implica "b", pero ello no permite apreciar un aspecto nuevo que no se tiene demasiado en cuenta en general en antropología, : el para que, el significado final de esa relación. Y eso es decisivo en una cultura tecnológica en la que los llamados "requerimientos" del sistema a construir son cada vez más importantes.

La clave en las culturas prefigurativas es el análisis de los planes, proyectos, ideas de futuro que se hace esa comunidad y en torno a los cuales orienta su actuación.

BIBLIOGRAFIA DE LA TESIS.

- ABELSON, Harold.

1991. "Computation as a Framework for Engineering Education". En Meyer, A., Guttag, J., Rivest, R., Szolovits, P. . Research Directions in Computer Science. An MIT Perspective. Cambridge, M.A. The M.I.T. Press.

-ABELSON,H. & SUSSMAN,G..

1985. Structure and Interpretation of Computer Programs. Cambridge. MA.: MIT Press.

-ACHENBAUM, W.Andrew.

1985. "The University-Wide Core Curriculum. A Rational Strategy for CMU's General Education Reform". Focus. vol 14, No 6, March 1985.

1986. " Teaching Design at CMU: From the Carnegie Plan to the 'Core'. in Covey, P.(ed.) The Role of Design in Liberal/Professional Education. CDEC. CMU.

- AMERICAN ASSOCIATION FOR THE ADVANCEMENT OF SCIENCE.

1992. AAAS Report XVI. Research and Development FY 1992. Washington D.C. Committee on Science, Engineering and Public Policy.

-ARMS, W.Y.

1990. "Report of the President's Computer and Network Planning Committee". Cursor. Academic Service Newsletter CMU., vol 7, No 2, March 1990.

1990. "Reflections on ANDREW". EDUCOM. Fall 1990.:33-43.

1992. Entrevista. 11 febrero. Warner Hall. CMU.

- ASPRAY, Jr. William F.

1980. From Mathematical Constructivity to Computer Science: Alan Turing, John Von Neumann, and the Origins of Computer Science in Mathematical Logic. Doctoral Thesis. University of Wisconsin-Madison.

- BABA, Marietta.

1985. University Innovation to Promote Economic Growth and University/Industry Relations. In Promoting Economic Growth through Innovation. Proceedings of the 1985 Conference on Industrial Science and Technological Innovation. :199-238. Washington, DC. NSF.

1988. The Academic-Industrial Connection: Building Linkages in the Field of Anthropology. In Anthropology for tomorrow: Creating Practitioner-Oriented Applied Anthropology Programs. R. Trotter, ed.: 128-160. Washington DC. American Anthropological Association.

- BARBACCI, Mario.

1992. Entrevista . Software Engineering Institute. 21 de febrero.

- BARDEEN, J. & BRATTAIN, W.H.

1948. "The Transistor, A Semi-Conductor Triode". June 25. The Physical Review. Vol 74, Second Series, July1-Dec. 15, Published for the American Physical Society by the American Institute of Physics.:230-231.

-BARNES, Barry et. al.

1980. Estudios sobre sociología de la ciencia. Madrid. Alianza Ed.

-BARNES, J.A.

1972. Social Networks. Addison-Wesley Module in Anthropology. No 26.

-BARNETT, Homer.

1942. "Invention and Cultural Change". American Anthropologist. no44 :14-30.

1953. Innovation: The Basis of Cultural Change. New York: Mc Graw-Hill.

-BARR, Avron & FEIGENBAUM, Edward. (ed.)

1982. The Handbook of Artificial Intelligence. Vol III. Stanford.CA. HeurisTech Press. William Kaufmann, Inc.

-BASTIDE, Roger.

1971. Antropología Aplicada. Buenos Aires. Amorrortu editores.

-BAUMANN, D.W.

1973. "Design Education: The Art of Engineering". Focus, No 10. March 19.

-BELL, Daniel.

1976. El advenimiento de la sociedad postindustrial. Madrid. Alianza Editorial.

(1973. The Coming of Post-Industrial Society: A Venture in Social Forecasting. New York: Basic Books.

1977. Las contradicciones culturales del capitalismo. Madrid. Alianza Editorial.
(1976, The Cultural Contradictions of Capitalism. New York. Basic Books.).

- BENVENISTE, E.
1969. Le Vocabulaire des Institutions Indo-europeenes. 2 vol. París.
- BEN-DAVID, Joseph y ZLOCZOWER, A.
1962. " El desarrollo de la ciencia institucionalizada en Alemania". en Kuhn, T., Merton, R.K., y otros. Estudios sobre sociología de la ciencia. 1980. Madrid. Alianza Editorial: 46-59.
- BENEDICT, Ruth.
1974. El crisantemo y la espada. Patrones de la cultura japonesa. Madrid. Alianza Ed.
- BENIGER, James R.
1986. The Control Revolution. Technological and Economic Origins of the Information Society. Cambridge, M.A.: Harvard University Press.
- BERLINER, Hans J.
1990. "Producing Behavior in a Searching PProgram". En Rashid, R. (Ed.). CMU Computer Science. 25th Anniversary Commemorative. New York. ACM Press. : 311-344.
- BIGELOW, Jacob.
1829. Elements of Technology. Boston: Boston Press.
- BOEHM, Barry W.
1976. "Software Engineering". IEEE Transactions on Computers. Vol C-25, no 12, December 1976.: 1226-1241.
- BOOCH, Grady.
1987. Software Engineering with ADA. 2nd ed. Menlo Park, CA: The Benjamin/Cummings Publishing Company, Inc.
- BOORSTIN, Daniel J.
1978. The Republic of Technology. Reflections on Our Future Community. New York. Harper & Row Publishers.
- BRAND, Steward.
1988. "Inventing the Future: The Media Lab at MIT". Computers and People, January-February, 1988.7-27.
- BRANSCOMB, Lewis M.
1991. " Toward a U.S. Technology Policy". Issues in Science and Technology. Summer 1991: 50-55.
- BROOKS, Frederick P.
1982. The Mythical Man-Month. Essays on Software Engineering. Reading, MA:

Addison-Wesley Publishing CO.

-BRUNER, Jerome.

1986. Actual Minds, Possible Worlds. Cambridge, MA. Harvard University Press.

1990. Acts of Meaning. Cambridge, MA. Harvard University Press.

-BUCHANAN, Bruce G.

1985. Expert Systems: Working Systems and the Research Literature. Report No KSL-85-37. Knowledge Systems Laboratory. Department of Computer Science. Stanford University.

- BUGLIARELLO, George.

1991. "The University- and Particularly The Technological University: Pragmatism and Beyond". en D.S. Zinberg (ed.) The Changing University, Kluwer Academic Publishers. Netherlands, 31-37.

1968. "Advanced Courses and Laboratories for Bioengineering". En Proceedings of the Rose Polytechnic Institute. Fifth Bioengineering Symposium. Progress in Bioengineering Education. Terra Haute. Indiana. October 21-22. Robert M. Arthur, Chairman.: 68.

-BUSH, Vannevar.

1945. Science- The Endless Frontier. A report to the President on a program for Postwar Scientific Research. Washington, D.C.: National Science Foundation.

-BUXO, M.J.,

& Santalo, C.A., Rodriguez Becerra, S. (Coord).

1989. La religiosidad popular. Barcelona. Anthropos.

1987. "Antropología Cultural". Enciclopedia Universal Ilustrada. Suplemento 1983-84. Madrid. Espasa Calpe, S.A.:21-30.

1984. "La cultura en el ambito de la cognición" :31-60. En Berenguer et al. Sobre el concepto de cultura. Barcelona. Ed. Mitre.

- CARBONELL, Jaime.

1990. "Artificial Intelligence: Past, Present & Future". VIA-90. Conference, Barcelona.

& GIL, Yolanda.

1987. "Learning by Experimentation: The Operator Refinement Method. Technical Report AIP-293. The Artificial Intelligence and Psychology Project. Departments of Computer Science and Psychology. CMU & Learning Research and Development Center. University of Pittsburgh.

& GIL, Yolanda., JOSEPH, Robert., KNOBLOCK, Craig A., MINTON, Steve., VELOSO,

Manuela M.

1990. "Designing an Integrated Architecture: The PRODIGY View.". SCS. CMU.
& HAYES, Philips.

1987. " Natural-Language Understanding". en Shapiro S. (Ed.). Encyclopedia of
Artificial Intelligence. Vol 1. New York. John Wiley & Sons.:661-677.

& MICHALSKI, R., MITCHELL, T.

1983. Machine Learning, Part I: A Historical and Methodological Analysis.
CMU-CS-83-135. Department of Computer Science. CMU.

& LANGLEY, Pat.

1987. "Machine Learning", en Shapiro S.C. (ed.). Encyclopedia of Artificial
Intelligence. Vol 1. New York. John Wiley & Sons.: 465-488.

- CARNEGIE, Andrew.

1905. James Watt. Garden City, New York: Doubleday, Doran & Company, Inc.

1979. " A New Idea of Education is Now Upon Us" From Carnegie's address
dedicating a library to the steel workers in Braddock in 1889. Focus. Vol. 8. No
8-9. May 1979.

-CARNEGIE MELLON.

1990-91. Faculty/Staff Directory.

1991-92. Faculty/Staff Directory.

-CARNEGIE MELLON. COMPUTER SCIENCE.

1989-90. Faculty Research Guide. School of Computer Science.

1990-91. Faculty Research Guide. School of Computer Science.

1991-92. Faculty Research Guide. School of Computer Science.

-CARNEGIE MELLON. ENGINEERING DESIGN RESEARCH CENTER.

1991-92. Report and Proposal.

-CARNEGIE MELLON UNIVERSITY.

1990-92. Undergraduate Catalogue. CMU.

-CARNEGIE MELLON UNIVERSITY.

1983. Proposal for Research on Supercomputers. Submitted to DARPA by CMU.

Duration: July 1, 1983 to June 30, 1986. Principal Investigators: H.T. Kung,
Allen Newell, Raj Reddy.

-CARNEGIE MELLON UNIVERSITY.

1988. Plans and Strategic Goals.

-CARNEGIE MELLON UNIVERSITY. DEPARTMENT OF COMPUTER SCIENCE.
1975-1976. Proposal for Continuation of Research in Information Processing.

Submitted to Advanced Research Project Agency (ARPA) of The Defense Department.

Principal Investigators: Allen Newell, Joseph Traub. Duration: July 1,

1975-Dec.31 1971. CMU Proposal No. 03027.

1976-1978. Proposal for Continuing of Research in Information Processing.

Submitted to ARPA of DoD. Principal Investigators: Allen Newell , Joseph Traub.

Duration: July 1976 to June 1978.

1980-1983. Proposal for Information Processing Research. Submitted to Defense

Advanced Research Projets Agency. Principal Investigators: A. Nico Habermann,

Allen Newell, Raj Reddy, Wm. A. Wulf. Duration: January 1, 1980 to December, 31,

1983.

1981-1984. Proposal for Information Processing Research. II. Submitted to DARPA.

Principal Investigators: A. Nico Habermann, Allen Newell, Raj Reddy, Wm. A.

Wulf. Duration: July 1, 1981 to June, 30, 1984.

1984-1986. Proposal for Information Processing Research. Submitted to DARPA.

Principal Investigators: A. Nico Habermann, Allen Newell, Raj Reddy.

1987-1990. Proposal for Information Processing Research. Submitted to DARPA for
the period 1 June 1987 to 31 May 1990. Principal Investigators: Scott E.

Fahlman, Allen Newell, Alfred Z. Spector.

- CARNEGIE MELLON UNIVERSITY. DEPARTMENT OF COMPUTER SCIENCE.

1976-77. Speech Understanding Systems. Summary of Results of the Five-Year
Research Effort at CMU.

- CARNEGIE MELLON UNIVERSITY. DEPARTMENT OF COMPUTER SCIENCE.

1988. "The Ergo Project: Semantically Based Program-Design Environments". A
research proposal to the DARPA and the Office of Naval Research.

-CARNEGIE MELLON UNIVERSITY. DEPARTMENT OF COMPUTER SCIENCE.

1986. Proposal for Research on Parallel Computing. Submitted to DARPA for the
period 1 October 1986 to 30 September 1989. Principal Investigators: H.T. Kung,

Richard F. Rashid.

-CARNEGIE MELLON UNIVERSITY FACTS.

1990. VOL 4. CMU. Planning Information Document.

-CARNEGIE MELLON UNIVERSITY/INTERNATIONAL BUSINESS MACHINES, INC.

1982. "News conference announcing development fo personal computing environment.

Oct. 20, 1982, in the Wherrett Room, Skibo Hall.

-CARNEGIE MELLON UNIVERSITY. MERCURY TECHNICAL REPORTS SERIES.

1992. "The Mercury Electronic Library and Library Information System II. The First Three Years". No 6. February 1992.

- CARNEGIE MELLON UNIVERSITY. ROBOTICS INSTITUTE.

1987. "Autonomous Planetary Rover". Principal Investigators: Takeo Kanade, Tom Mitchell, William L. Whittacker. Submitted to: National Aeronautics and Space Administration.

-CARNEGIE MELLON UNIVERSITY. SCHOOL OF COMPUTER SCIENCE.

1989. "Basic Research in Computer Science: Integrated Architectures for Intelligent Systems", Vol I. Technical. Submitted to DARPA for the period 1 August 1990 through 31 July 1993. Topical Category: E. Machine Intelligence.

-CARNEGIE MELLON UNIVERSITY. SCHOOL OF COMPUTER SCIENCE .

1989. "Basic Research in Computer Science: Object Management." Vol I: Technical. Submitted to DARPA for the period 1 August 1990 through 31 July 1993. Topical Category: C. Software.

1989. "Basic Research in Computer Science: Creating Graphical Applications". Vol I: Technical. Submitted to DARPA for the period 1 August 1990 through 31 July 1993. Topical category: C. Software, D. Networking/Command, Control and Communications.

1989. "Basic Research in Computer Science: Types in Programming". Vol I: Technical. June 1989. Submitted to DARPA for the period 1 August 1990 through 31 July 1993. Topical category: C. Software.

1989. " Basic Research in Computer Science: VLSI Logic Validation and Parallel Computation. Vol I: Technical. Submitted to DARPA for the period 1 August 1990 through 31 July 1993. Topical category: B. Microsystem Design and Prototyping, A. Parallel Computing Architectures.

-CARNEGIE MELLON UNIVERSITY. SCHOOL OF COMPUTER SCIENCE.

1989. "Research on Parallel Computing: Network-based Multicomputers. Vol 1: Technical. Submitted to DARPA for the period 1 March 1990 through 28 February 1993. Topical category: A. Parallel Computing Architectures.

1989. "Research on Parallel Computing: Very Large Scale Operating Systems". Vol I: Technical. Submitted to DARPA, for the period 1 March 1990 to 28 February 1993. Topical category: A. Parallel Computing Architectures.

-CARNEGIE MELLON UNIVERSITY-SCHOOL OF COMPUTER SCIENCE-ROBOTICS INSTITUTE.

1991. New Concepts for Advanced High Definition Systems. Submitted to DARPA, 30 mayo. Technical Contact: A. G. Jordan.

-CARNEGIE MELLON UNIVERSITY.

1989. The Search for the President. March 17. A report prepared for the Faculty Presidential Search Committee. John G. Fetkovich, Chair, by the Selection and Plans Committee, Dana Scott, Chair.

- CMU. SCHOOL OF COMPUTER SCIENCE.

1990. 25th Anniversary Symposium. Sept 23-26. School of Computer Science.

- CARNEGIE SCHOOL OF TECHNOLOGY.

1903. Plan and Scope of the Proposed Carnegie School of Technology. Preliminary Report.

-CASTELLS, M. et al.

1985. Nuevas Tecnologías, Economía y Sociedad en España. 2 vol. Madrid. Alianza Editorial.

-CERF, Vinton.

1991. "Networks". Scientific American. September, 1991. vol 265, No3. :72-86.

- CHAMBERS, Erve.

1985. Applied Anthropology. Englewood Cliffs, N.J. Prentice-Hall, Inc.

- CHANNING, William Ellery.

1957. Unitarian Christianity and Other Essays. New York. The Liberal Arts Press.

-CHIGIER, Norman.

1983. "The Design of Flames". Focus, a publication of the faculty and staff of Carnegie Mellon University. vol. 12, No 5. February.

-CLEETON, GLEN U.

1965, The Doherty Administration. 1936-1950. The Story of Carnegie Tech, II. The Carnegie Press. Carnegie Institute of Technology.

- COMISION INTERMINISTERIAL DE CIENCIA Y TECNOLOGIA.

1989. "Nomenclatura internacional de la UNESCO para los campos de ciencia y tecnología". en Clasificaciones Científicas. Madrid. Centro de Publicaciones.

Ministerio de Educacion y Ciencia.

-CONWAY, LYNN.

1981. The MPC Adventures: Experiences with the Generation of VLSI Design and Implementation Methodologies. Xerox Palo Alto Research Center. VLSI-81-2.

-COVEY, Preston. (ed.)

1988. The Role of Design in Liberal/Professional Education. CMU: Center for Design of Educational Computing.

1988. "Advanced Computer Applications for the New Liberal Arts". CDEC. CMU.

1991. "CDEC Overview". Sept 25, 1991.

1992. "Notes on 'Science of Design, High Technologies & Cultural Tradition". CDEC. CMU.

& Schawartz, Tom.

1974. "Toward A Philosophy Department". Focus, vol 4, No 4, Dec., 19, 1974.

- CRECINE, Pat.

1985. "The Role of Universities in Knowledge Dissemination: Site Licensing is Secondary". EDUCOM Bulletin, Summer 1985: 2-5.

-CYERT, Richard.

1983. "The Cognitive Revolution on Campus". En "Personal Computing". March 1983:36.

1982. CMU-IBM News Conference announcing development of personal computing environment. October, 20 in the Wherrett Room, Skibo Hall.

1986. "The Impact of Microcomputers on Education". Perspectives on Computing. Vol 6. No. 2: 4-8.

1986. Planning in a Decentralized Society. 30 de Septiembre.

1988. "Carnegie Mellon University". en Steeples, D.W. (ed.) Successful Strategic Planning: Case Studies. New Directions for Higher Education. No 64. San Francisco: Jossey-Bass, Winter 1988: 91-98.

-D'ANDRADE, R.

1981. "The Cultural Part of Cognition". Cognitive Science, 5. 1981: 179-195.

-DEFENSE ADVANCED RESEARCH PROJECTS AGENCY.(DARPA)

1983. Strategic Computing. New Generation Computing Technology: An Strategic Plan for its Development and Applications to Critical Problems in Defense. DARPA.

-DENNIGN, Peter J. et al.

1989. "Computing as a discipline". Communications of the ACM. January 1989. Vol 32. No.1: 10-23).

-DEPARTMENT OF DEFENSE.

1983. "Military Standard, ADA Programming Language". American National Standards Institute, Inc.

-DERTOUZOS, Michael et al.

1988. The National Challenge in Computer Science and Technology. CSTB.
Washington D.C.: National Academy Press.

1991. " Communications, Computers and Networks" Scientific American. vol 265, No 3: 62-69

&LESTER, R.K. , SOLOW, R. M.

1989. Made in America.Regaining the Productive Edge. New York. MIT Press.

-DEUTSCH, Claudia.

1991. " What Makes a Given Industry Tick?". Entrevista a Ralph Gomory. New York Times. August 18, 1991.

- DEWEY, John.

1950. Lógica: Teoría de la Investigación. Mexico. FCE. (1a ed. 1938)

1989. Cómo pensamos. Buenos Aires. Ed. Paidos (1a. ed. 1910. How We Think.).

-DIAZ, Josep.

1984, "Génesis y evolución de la Informática Teórica". RT 84/05. Facultat d'Informatica de Barcelona. Universitat Politecnica de Catalunya.

-DIJKSTRA, Edsgar, W.

1971. "A Short Introduction to the Art of Programming". EWD 16.
August, 1971, 19 pp.

1989. " On the Cruelty of Really Teaching Computing Science". Communications of the ACM. Dec. 1989. Vol 32., No12: 1398-1404.

-DIXON, Roland B.

1928. The Building of Cultures. New York. Charles Scribner's Sons.

- DOHERTY, Robert E.

1950. The Development of Professional Education. The Principles which have guided the Reconstruction of Education at Carnegie Institute of Technology as Stated by its President. Published by Carnegie Institute of Technology.

- DONAHUE, H.C.

1989. " Choosing the TV of the FUTURE". Technology Review. April.

-DOWNEY, Gary L.

1991. "CAD/CAM SAVES THE NATION?. Toward and Anthropology of Technology." (En proceso de revisión en la revista Knowledge in Society, como parte de un número especial sobre "anthropology of science and technology").

-&DONOVAN, Arthur, ELLIOTT, Timothy J.

1989. "The Invisible Engineer: How Engineering Ceased To Be A Problem In Science And Technologies Studies". Knowledge and Society: Studies in the Sociology of

Science Past and Present. Vol 8.: 189-216.

-EDDY, E. & PARTRIDGE, W.

1987. Applied Anthropology in America. New York. Columbia University Press.

-ELKANA, Yehuda.

1981. "A Programmatic Attempt at an Anthropology of Knowledge". en Mendelsohn, Everett y Elkana, Yehuda, Sciences and Cultures. Dordrecht, Holland. D. Reidel Publishing Company: 1-77.

-ENGINEERING DESIGN AND DESIGN EDUCATION.

1967. "Toward a Science of Design". Fourth Conference ofn Engineering Design. Proceedings. July 17 and 18, 1967. Thayer School of Engineering. Darmouth College. Hanover, New Hampshire.

-ESTEVA FABREGAT, Claudio.

1973. Antropologia Industrial. Barcelona. Anthropos.

1975. "La antropología aplicada y su problemática". Primera Reunión de Antropólogos Españoles. Sevilla. Universidad de Sevilla. : 253-321.

1977. "Antropología Aplicada": ¿Comunidad o Sistema?." 1er Congreso Español de Antropología. Universidad de Barcelona: 231-281.

1978. Cultura, Sociedad y Personalidad. Barcelona. Promocion Cultural, S.A.

1984. "El concepto de cultura". En Berenguer A. et al. Sobre el concepto de cultura. Barcelona. Editorial Mitre.:61-80.

- FEIGENBAUM, Ed.

1989. "What Hath Simon Wrought?". EN Klahr, D. & Kotovsky, K. Complex Information Processing. Hillsdale, N.J. Lawrence Erlbaum Associates, Publishers: 165-182.

& McCORDUCK, Pamela.

1983. La Quinta Generación. Barcelona. Planeta.

- FLORES, Fernando & WINOGRAD, Terry.

1989. Hacia la comprensión de la informática y la cognición. Barcelona. Editorial Hispano Europea.

-FLORIDA, Richard & KENNEY, Martin.

1990. The Breakthrough Illusion. Basic Books.

-FLORMAN, Samuel C.

1976. The Existential Pleasures of Engineering. New York: St. Martin's Press.

-FORSYTHE, Diana

1990. "Engineering Knowledge: The Construction of 'Knowledge' in Artificial

Intelligence". University of Pittsburgh. Computer Science Department.
Intelligent Systems Laboratory.
1991. "The Construction of Work in Artificial Intelligence". University of
Pittsburgh. Intelligent Systems Laboratory. Report No. ISL-91-4WP.
-FORSYTHE, George.
1963. "Educational Implications of the computer revolution". En W. F.
Freiberger & W. Prager (ed.). Applications of Digital Computers. Ginn. Boston:
166-178.
-FOSTER, George M.
1974. Antropologia Aplicada. Mexico. Fondo de Cultura Economica.
-FUCHI, Kazuhiro.
1991. "Future Development of Parallel Inference Technology". ICOT Journal. no.
29. Tokyo. ICOT.
-FURST, Merrick.
1992. Entrevista personal. School of Computer Science. 16 abril.
-GARDNER, Howard.
1987. La nueva ciencia de la mente. Historia de la revolución cognitiva. Buenos
Aires: Paidós.
-GEERTZ, Clifford.
1973. The Interpretation of Cultures. New York. Basic Books Inc.
1983. Local Knowledge. New York. Basic Books, Inc. Publishers.
-GILL, Kay & TUFTS, S. (ed).
1992. Government Research Directory. Detroit. MI: Gale Research Inc.
-GINER, Salvador.
1984. "Tecnocultura". La Vanguardia. sábado, 4 febrero: 8.
1987. Ensayos civiles. Barcelona. Edicions 62 s/a.
-GIOVENGO, Annette Lois.
1986. The Role of the University in High-Technology Economic Development:
Pittsburgh as a Case Study. CMU. College of Humanities and Social Sciences.
Doctoral Dissertation. April 1986.
-GIBBS, Norman E.
1989. "The SEI Education Program: The Challenge of Teaching Future Software
Engineers". Communications of the ACM. May 1989. Vol. 32, No.5:594-605).
-GOODENOUGH, W.
1957. "Cultural Anthropology and Linguistics". En Report of the Seventh Annual

Round Table Meeting on Linguistics and Language Study. P. Garvin, ed. Georgetown University Monograph Series on Languages and Linguistics, No. 9.

-GOMORY, Ralph E.

1989. "From the 'Ladder of Science' to the Product Development Cycle". Harvard Business Review. November-December. 1989: 99-105.

-GRAMSCI, Antonio.

1978. Introducción a la filosofía de la praxis. 4a ed. Barcelona. Edicions 62 S/A.

-GRASSMUCK, K.

1990. "Some Research Universities Contemplate Sweeping Changes, Ranging From Management and Tenure to Teaching Methods". The Chronicle of Higher Education. Sept 12, vol XXXVII, No 2.

-HABERMANN, Nico.

1990. 25th Anniversary Symposium. SCS. CMU. September 23th-26th.: 36-37.

1990. "Computer Science comes of age". Carnegie Mellon Magazine.:56.

& GARLAN, David., NOTKIN, David.

1990. "Generation of Integrated Task-Specific Software Environments" . En

Rashid, Rich (ed.) CMU Computer Science. New York.ACM Press.

& McDERMOTT, John, NEWELL, Allen. REDDY,Raj.

1991. Discurso de despedida como Decano de la School of Computer Science. 27 de septiembre. Skibo. CMU.

1985. Computer Science and the Universe of Disciplines. 7 Abril 1985. Department of Computer Science. CMU. Draft Material: Limited Distribution for Comment. Not for Quotation.

- HAKKEN, David.

1991. "If Computing Is a Cultural Process, What Constructs Should Inform Its Practice?". Paper presented at the Session: Cultural Perspectives of Information System Development. 90° Congress of the AAA. Chicago. Nov. 20th-24th.

-HARRIS, Marvin.

1979. El desarrollo de la teoría antropológica. Una historia de las teorías de las culturas. Madrid. Alianza Editorial.

1983. Introducción a la Antropología General. Madrid. Alianza Editorial. 3a. ed. de Alianza Universidad Textos.

- HAMBLIN, J.W.

1984. Computer Manpower -Supply and Demand by States- 1984. Tallahassee, Fla.

Quad Data Corporation.

-HAYES, John R.

1989. The Complete Problem Solver. Hillsdale, N.J. Lawrence Erlbaum Associates, Publishers.

-HEART, F.E., KAHN,R., ORNSTEIN, S.M., CROWTHER, W., WALDEN, D.

1981. " The Interface Message Processor for the ARPA Computer Network". En Siewiorek, Bell, Newell, Computer Structures: Principles and Examples. New York. McGraw-Hill Book Company.: 402-418.

- HELMER, Olaf.

1966. Social Technology. New York. Basic Books.

- HERSKOVITS, Melville.

1981. El Hombre y sus Obras. Mexico. Fondo de Cultura Economica. (1a ed. 1948).

-HILLIS, W.D.

1985. The Connection Machine. An ACM Distinguished Dissertation. Cambridge, MA. The MIT Press.

- HOLMBERG, Allan R.

1955. "Participant Intervention in the Field". Human Organization. Vol 14. No1.: 23-26).

1956. "From Paternalism to Democracy: The Cornell-Peru Project" Human Organization. Vol. 15. No3.: 15-19.

1958, " The Research and Development Approach to the Study of Change". Human Organization. Vol 17. No 1. : 12-19.

-HOWARD, John.

1990. "Andrew at 10". Carnegie Mellon Magazine. Fall 1990: 40.

- HUGHES, Thomas P.

1989. American Genesis. A Century of Invention and Technological Enthusiam. 1870-1970. New York. Viking Penguin.

- HURLBUTT III, Robert H.

1965. Hume, Newton and the Design Argument. Lincoln. University of Nebraska Press.

- IBM ACADEMIC INFORMATION SYSTEMS.

1986. "CARNEGIE MELLON UNIVERSITY. Reaching for World Leadership in Educational Computing and Communications". IBM. Milford CT.

- JAMES, William.

1984. Pragmatismo. Barcelona. Sarpe. (1a ed.1907).

- JANTSCH, Eric.
1967. Technological Forecasting in Perspective, Paris. O.C.D.E. Paris.
- JORDAN, Angel.
1990. "Angel Jordan: Los abismos de la ciencia". Entrevista de Soledad Alameda.
EL PAIS dominical. 16 Diciembre.
- JORDAN, Angel & HABERMANN, Nico.
1988. The Formation fo the School of Computer Science. September 16. Carnegie Mellon.
1976. "Electronics and Computers: A Personal Experience". Focus vol. 6, No 2, oct. 13, 1976.
- JOSEPHSON, Matthew.
1959. Edison. New York. McGraw-Hill.
- KAHN, Robert E.
1988. "A 'National Network: Today's Reality, Tomorrow's Vision, Part 2". EDUCOM Bulletin, Summer/ Fall 1988. :14-21.
- KANNAN, Ravindran.
1991. "Random Walks, Parametric Integer Programming". Proposal to the National Science Foundation. NSF 90-77. CMU. 23 dic. 1991.
& FURST, Merrick.
1992. Entrevista. 21 abril. Skibo. CMU.
- KARATSU, Hajime.
1986. Tough Words for American Industry. Cambridge, MA. Productivity Press.
- KERNOFF, Alan.
1990. Who's Who in Artificial Intelligence. Published by WWAI, in association with Tom Schwartz Assoc.
- KIBBEY, Mark., & EVANS, Nancy, H.
1989. "The NETwork is the Library". EDUCOM Review. Fall 1989.:15-20.
- KIESLER, S. & SPROULL, L.
1987. Computing and Change on Campus. New York. Cambridge University Press.
- KLUCHOHN, Clyde.
1983. Antropología. México. Breviarios FCE. (1a ed. 1949. Mirror for Man).
- KONDRATIEFF, Nikolai.
1984. The Long Wave Cycle. Richardson & Schneider. (1a ed. 1925).
- KROEBER, A & KLUCHOHN, C.
1963. Culture. A Critical Review fo Concepts and Definitions. New York. Random

House.

-KNUTH, Donald.

1974. "Computer Science and Its Relation to Mathematics". American Mathematical Monthly. April 1974: 323-343.

- KUHN, Thomas S.

1987. La estructura de las revoluciones científicas. Mexico. Breviarios del Fondo de Cultura Económica. 12a ed.

-KUNG. H.T. et al.

1989. The Design of Nectar: A Network Backplane for Heterogeneous Multicomputers. CMU-CS-89-101.

1990. Gigabit Nectar Testbed. SCS. 15 February 1990.

1990. "Heterogeneous Multicomputers". En Rashid, Rich. (ed.). CMU Computer Science. A 25th Anniversary Commemorative. New York: ACM Press.:235-253.

1991. Entrevista. 15 de marzo. School of Computer Science.

&COOPER, Eric & LEVINE, Michael,

1989. Gigabit Nectar Testbed. Submitted to: Corporation for National Research Initiatives.

&LEISERSON, Charles E.

1979. "Systolic Arrays (for VLSI)". en Duff Ian S., & Stewart, G.W., Sparse Matrix Proceedings 1978. Philadelphia: SIAM, : 256-281.

-KUZNETS, Simon.

1953. Economic Change. Westport, CT. Greenwood Press.

- LAKATOS, Imre y MUSGRAVE, ALan

1975. La crítica y el desarrollo del conocimiento. Actas del Coloquio Internacional de Filosofía de la Ciencia celebrado en Londres en 1965. Barcelona. Ediciones Grijalbo.

- LANGLEY, P. & CARBONELL, J.

1984. " Approaches to Machine Learning". 16 Febrero. CMU-CS-84-108. Computer Science Department. Carnegie Mellon.

-LANGLEY, P. SIMON, H., BRADSHAW, G. ZYTKOW, J.

1987. Cambridge, M.A. The MIT Press.

- LASSWELL, Harold.

1971. El futuro de la ciencia política. Madrid. Tecnos.

- LATOUR, Bruno y WOLGAR, Steve.

1979. Laboratory Life. The Construction of Scientific Facts. Princeton, New

Jersey: Princeton University Press.

- LATOUR, Bruno.

1987. Science in Action. Cambridge. Massachusetts: Harvard University Press.

- LAYTON, Edwin.

1986. (1971). The Revolt of Engineers: Social Responsibility and the American Engineering Profession. Baltimore: The Johns Hopkins University Press.

-LEISERSON, Charles E.

1981. "Area-Efficient VLSI Computation". Doctoral Dissertation. October 1981. SCS. CMU.

1990. "VLSI Theory and Parallel Supercomputing". En Rashid, R. (ed.). CMU Computer Science. A 25th Anniversary Commemorative. New York: ACM Press.:29-44.

-LENAT, Douglas.

1989. " Very Large Knowledge Bases". en Association for Computing Machinery and the Computing Research Association, Strategic Direction in Computing Research. Oct. 11-13 -1989. ACM Press.

-LEVI-STRAUSS, C.

1987. El hombre desnudo. Mexico. Siglo XXI Ed. (1a ed. 1971. Mythologiques IV. L'homme nu.).

1984. El oensamiento salvaje. México. Breviarios del Fondo de Cultura Económica. & ERIBON, D.

1990. De pres et de loin. Paris. Editions Odile Jacob.

- LIEBLEIN, Edward.

1986. "The Department of Defense Software Initiative"--A Status Report." Communications of the ACM. August 1986. Vol. 29, No 8. :734-744.

-LIVESAY, Harold C.

1975. Andrew Carnegie and the Rise of Big Business. Boston. Little, Brown and Company.

-LONGO, Guiseppe.

1988. "Some Aspects of Impredicability. Notes on Weyl's philosophy of Mathematics and on todays Type Theory". Report CMU-CS-88-135. Carnegie Mellon.

-LOVE, Karl.

1991. Entrevista. 12 junio. School of Computer Science.

-MALINOWSKY, Bronislaw.

1982. Magia, Ciencia y Religión. Barcelona. Editorial Ariel, S.A. 2a ed.

-MARCH, James & Simo, Herbert,

1981. Teoría de la organización. Barcelona. Ariel Economía. Ed. Ariel.

-MARUYAMA M. & HARKINS, A.

1978. Cultures of the Future. Mouton Publishers. The Hague. Paris.

-MASUDA, Yoneji.

1980. The Information Society as Post-Industrial Society. Tokyo. Institute for Information Society.

-McCARTHY, J. & HAYES, P.J.

1969. "Some Philosophical Problems from the Standpoint of Artificial Intelligence". en Meltzer, B, Michie, D.(ed). Machine Intelligence. vol 4. New York. Elsevier Publishing Co. Inc.:463-502.

-McCORDUCK, Pamela.

1979. Machines Who Think. San Francisco. W.H. Freeman and Company.

-MEAD, Margaret.

1943. And Keep Your Powder Dry. New York. William Morrow Co.

1953. Cultural Patterns and Technical Change. Paris. UNESCO.

1964. Continuities in Cultural Evolution. New Haven. Yale Univ. Press.

1977. Cultura y Compromiso. Barcelona. Granica Editor.

1978. "The Contribution of Anthropology of the Science of the Future". En Maruyama, M & Harkins, A. The Hague. Mouton Publishers. :3-6.

1978. "Anthropology in Culture Building. Results and Problems of World Anthropology 1977". en SHIMKIN, Demetri , TAX, Sol y MORRISON, John. Anthropology for the Future. Department of Anthropology. University of Illinois. Urbana, Illinois. Research Report No 4.

1979. "The Uses of Anthropology in World War II and After". en Goldschmidt, W. The Uses of Anthropology. A special publication of the American Anthropological Association. No. 11.: 145-157.

-MEDINA, M.

1989. " Mito de la teoría y filosofía de la tecnología". Anthropos. No 94-95. :35-39.

-MEHRABIAN, Robert.

1991. Discurso de despedida de Nico Habermann como Decano de la School of Computer Science. 27 Sept. 1991. Skibo. CMU.

-MERTON, Robert. K.

1942. " Los imperativos institucionales de la ciencia". En Barry Barnes et al. Estudios sobre Sociología de la Ciencia. 1980 Madrid. Alianza Editorial.:64-78.

1977. Teoría y Estructura Sociales. México/Buenos Aires. Fondo de Cultura Económica.

-MICHALSKI, R.S., CARBONELL, J.G. , MITCHELL, T.M. (ed.)
1984. Machine Learning. An Artificial Intelligence Approach. Berlin Heidelberg. Springer-Verlag.

-MIDANI, Akram & TEARE, Dick.
1976. "‘Methods’ in Art and Engineering". Focus, vol 5. No4. January 9.

-MILLER, George A.
1989. "Scientists of the Artificial". En Klahr D. & Kotovsky, K. Complex Information Processing. Hillsdale, N.J. Lawrence Erlbaum Associates.:145-161.
& GALANTER, E. & PRIBRAM, K.H.

1960. Plans and the Structure of the Behavior. New York; Holt, Rinehart & Wiston.

- MILLER, Philip L. & MILLER, Lee W.
1987. Programming by Design. Belmont, CA. Wadsworth Publishing Co., & Pittsburgh, PA. Carnegie Publishing, Inc.

-MINSKY, M.
1967. Computation: Finite and Infinite Machines. Englewood Cliffs: Prentice-Hall, Inc.

- MITCHAM, Carl.
1989." Tres formas de ser-con la tecnología". Anthropos. No 94-95. 13-26.

-MITCHELL, J. Clyde Ed.
1969. Social Networks in Urban Situations: Analysis of Personal Relationships in Central Africa Towns. Manchester. Manchester Univ. Press.

- MOODY, Fred.
1991. "Mr. Software". The New York Times Magazine, August 25,

-MORRIS,J. & SATYANARAYANAN, M., CONNER, M.H., HOWARD, J.H., ROSENTHAL, D.S.H.,
SMITH, F. Donelson.
1986. " Andrew: A Distributed Personal Computing Environment". Communications of the ACM. March 1986, vol. 29, No. 3.:184-201.

-MUNFORD, Lewis.
1934. Technics and Civilization. New York. Harcourt, Brace and Company.

- NATIONAL ACADEMY OF SCIENCES, NATIONAL ACADEMY OF ENGINEERING, INSTITUTE OF

MEDICINE: THE GOVERNMENT-UNIVERSITY-INDUSTRY RESEARCH ROUNDTABLE.

1989. " Science and Technology in the Academic Enterprise: Status, Trends, and Issues". Washington D.C. National Academy Press.

-NADER, Laura.

1974. " Up the Anthropology-Perspectives from Studying Up". En Hymes, Dell (ed.). Reinventing Anthropology. New York: Vintage Books.284-311.

-NATIONAL COMMISSION ON EXCELLENCE IN EDUCATION.

1983. " A Nation at Risk". Communications of the ACM. July 1983, vol 26, No7.:468-478.

-NATIONAL SCIENCE BOARD.

1989. Science & Engineering Indicators-1989. National Science Board Subcommittee on Science and Engineering Indicators-1989.

- NSFNET (NATIONAL SCIENCE FOUNDATION NETWORK)

1990. The National Science Foundation Computer Network for Research and Education. Milford, CT: IBM, Academic Information Systems.

- NEWELL, Allen.

1980. "Physical Symbol Systems". March 1980. CMU-CS-80-110. Department of Computer Science. CMU.

1982. "The Future of Computing at Carnegie Mellon University". The Task Force for the Future of Computing. CMU. Pittsburgh. PA.

1982. "Intellectual Issues in the History of Artificial Intelligence". CMU-CS-82-142. Department of Computer Science. CMU.

1989, "Putting It All Together". en Klahr, D. & Kotovsky, Kenneth Ed., Complex Information Processing. The Impact of Herbert Simon. Hillsdale, New Jersey. Lawrence Erlbaum Associates, Inc.

1989. "Epilogue. How It All Got Put Together ,A Story". ibid. :443-445.

1990. Unified Theories of Cognition. Cambridge MA. Harvard University Press.

1990. Entrevista personal. School of Computer Science. 17 de diciembre 1990.

1990. "Revisiting 'What is Computer Science'". en School of Computer Science, CMU. 25th Anniversary Symposium. :41.

1991. "Desires and Diversions". Charla en la School of Computer Science de CMU. 4 de diciembre de 1991.

- & PERLIS, A, SIMON, H.

1967. "What is Computer Science". In 25th Anniversary Symposium. School of Computer Science. CMU.

& PERLIS, A, SCHATZ, E.

1964. Proposal for a Center for the Study of Information Processing. Carnegie Institute of Technology. October 1964.

& SHAW, J.C.

1957. Programming the Logic Theory Machine. P-954. Feb. 28, 1957. Revised. The Rand Corporation. Herbert Simon Personal Archives. Hunt Library. CMU.

&SHAW, J.C. & SIMON, H.

1957. "Elements of a Theory of Human Problem Solving". P-971. March 4, The Rand Corporation. (*) Allen Newell and J.C. Shaw fo the Rand Corporation an Herbert A. Simon of Carnegie Institute of Technology. (H.Simon. Personal Archives. Hunt Library. CMU).

& SHAW, J.C. , SIMON, H.

1963. "Empirical Explorations with the Logic Theory Machine: A Case Study in Heuristics". En Feigenbaum, Ed. & Feldmann, J, Computers and Thought. New York. McGraw Hill Book Co.

&H, SIMON.

1956, "The Logic Theory Machine, A Complex Information Processing System. July 12. Presented before the Professional Group on Information Theory of the Institute of Radio Engineers, Sept 10, 1956 at Cambridge, Massachusetts. Santa Monica. California. The Rand Corporation.

1956. "Currents Developments in Complex Information Processing". May 1, 1956. P-850. The Rand Corporation. Herbert Simon Personal Archives. Hunt Library. CMU. & H.SIMON.

1972, Human Problem Solving. Englewood Cliffs, New Jersey. Prentice-Hall, Inc.

& SIMON, H.

1976. "Computer Science as Empirical Inquiry: Symbols and Search".

Communications of the ACM. March 1976. Vol 19, No3. :113-126.

- NEWTON, Isaac.

1946. Mathematical Principles of Natural Phylosophy and his System of the World.

Berkeley, CA. University of California Press. (1a ed. 1686).

-NOBEL, David.

1977. America by Design. Oxford.Oxford University Press .

-OFFICE OF SCIENCE AND TECHNOLOGY POLICY.

1992.Grand Challenges: High Performance Computing and Communications. A Report by the Committee on Physical, Mathematical and Engineering Sciences. National

Science Foundation. Washington D.C.

- PAHNOS, David.

1991. Entrevista. 20 mayo. Field Robotics Center. CMU.

- PEARL, Judea.

1984. Heuristics. Intelligent Search Strategies for Computer Problem Solving.

Reading, MA. Addison-Wesley Publishing Company.

-PELAEZ, Eloína.

1990. "Parallelism and the Crisis of von Neumann Computing". Technology in Society. VOL. 12, : 65-77.

-PERLIS, Alan.

1972. Introduction to Computer Science. New York. Harper and Row, Publishers.

1977. "Computers in Science and Technology". In Science and Technology in

America, an assessment. Edward Teller et al. :105-114. U.S. Departmente of

Commerce.

1979. " Current Research Frontiers in Computer Science". En M. Dertouzos M. &

Moses J. (ed.). The Computer Age: A Twenty-Year View. Cambridge, MA. The MIT

Press.: 422-439.

- PITTSBURGH FACTS.

1990-91. Statistical Guide to the Pittsburgh Metropolitan Region.

- PHILLIPS, William D.

1991. Report on the National Critical Technology Panel. March 1991. U.S.

Departmmt of Commerce. PB91-156869.

- POLYA, George.

1973. How to solve it. A New Aspect of Mathematical Method. 2nd ed. Princeton,

N.J. Princeton University Press.

-POPPER, Karl.

1962. La lógica de la investigación científica. Madrid. Tecnos.

1987. La miseria del historicismo. 3a ed. Madrid. Alianza Ed. (1a ed. 1944).

- QUESADA, Daniel.

1985. La Lógica y su filosofía. Barcelona. Editorial Barcanova.

-RADIN, Paul.

1968. El hombre primitivo como filósofo. Buenos Aires: Editorial Eudeba. 2a ed.

- RAND CORPORATION, THE.

1963. The First Fifteen Years. Santa Mónica. California. Nov. 1963.

-RAPPAPORT A. & HALEVI, Samuel.

1991. "The Computerless Computer Company." Harvard Business Review. July-August. 1991: 69-80.
- RASHID, Rich.
1986. "Threads of a New System". Unix Review. Vol. 4. No 8. August 1986: 37-49.
1990. "Mach: a Case Study in Technology Transfer". en Rashid R. (ed.) CMU Computer Science. A 25th Anniversary Commemorative. New York. ACM Press: 395-421.
1990. CMU Computer Science. A 25th Anniversary Commemorative. New York. ACM Press.
1990. "Mach talk". Carnegie Mellon Magazine. fall 1990.: 42.
1991. CMU, Computer Science. A 25th Anniversary Commemorative. New York. Association for Computing Machinery Press.
1991. Entrevista personal. 12 marzo. School of Computer Science.
- REDDY, Raj.
1988. "Foundations and Grand Challenges of Artificial Intelligence". AI Magazine, Winter 1988. : 9-21.
1991. "Will 3G Supercomputers Also Gather Dust?". en Rashid, R. (ed.) CMU Computer Science. A 25th Anniversary Commemorative. New York. ACM Press.: 465-487.
- REIF, Frederick
1980. "Theoretical and Educational Concerns With Problem Solving: Bridging the Gaps With Human Cognitive Engineering", en Tuma,D.T. & Reif, F. Problem Solving and Education: Issues in Teaching and Research. Hillsdale, New Jersey. Lawrence Erlbaum Associates, Publishers.: 39-50.
1991. Clase en el Psychology Department de CMU. 6 septiembre.
- RIGGS, Henry.
1988. "Innovations: A United States-Japan Perspective". en Okimoto, D.I. y Rohlen, T.P. Inside the Japanese System. Stanford: Stanford University Press: 246-252.
- ROGERS, E. & LARSEN, J.
1986. La fiebre del "Silicon Valley". Barcelona. Editorial Reverté, S.A. (1982. Silicon Valley Fever. Growth of High-Technology Culture. Basic Books, Inc., New York.)
- ROSENBERG, Nathan & BIRDZELL, L.E. Jr.
1990. "Science, Technology and the Western Miracle". Scientific American.

November 1990. Vol. 263. No. 5.: 42-54.

-ROSENBLUTH, A., WIENER, N & BIGELOW, J.

1943. "Behavior, purpose and teleology". Philosophy of Science, 10: 18-24.

-RUSSELL BERNARD, H.

1988. Research Methods in Cultural Anthropology. Sage Publications. Newbury Park.

-SALOMON, Jean Jacques.

1989. "Les politiques d'innovation en Europe". Futuribles. No. 139. Mai 1989 :29-52.

-SAMBURSKY, S.

1956. The Physical World of The Greeks. New York. The Macmillan Co.

-SAMUELSON, Pamela.

1990. "Should Program Algorithms Be Patented?". Communications of the ACM. August 1990, Vol 33. No. 8: 23-27.

-SANMARTIN, José.

1989. "No toda producción es síntesis. Reflexiones en torno a las diferencias entre tecnología de control y tecnologías sintéticas". Anthropos. No 94-95: 39-44.

-SCOTT, Dana.

1977. " An Appreciation of Christopher Strachey and his work" Foreword. En Stoy Joseph E. Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory. Cambridge, MA. The MIT Press.

1977. "Logic and Programming Languages". Communications of the ACM. September 1977, vol 20. No 9. :634-641.

1989. The Search for the President. CMU. March 17. 1989.

-SCHUMPETER, J.

1939. Business Cycles. Vol 1. New York. McGraw-Hill Book, Co.

-SERRA, Arturo.

1988. "La posibilidad de un cambio cultural diseñado". en Problemas en torno a un cambio de civilización. Barcelona. El Laberinto. Ediciones de Nuevo Arte Thor.

& ROJO, Arcadio.

1990. " Devising a Designet. Plan of Action to set up an International Computer Network on Design in Culture, Society and Technology". Workshop held in the

Engineering Design Research Center CMU. March 17.

-SERVAN-SCHREIBER, Jean Jacques.

1986. La revolución del conocimiento. Nuevo desafío americano. Barcelona. Plaza & Janés Editores, S.A.

- SHAFER, Steve.

1990. "Why Can't We Model the Physical World?". En Rashid, R. (ed.). CMU, Computer Science. A 25th Anniversary Commemorative, ACM Press: 295-309.

-SHANNON, Claude & WEAVER, Warren

1949. The Mathematical Theory of Communication. Urbana, Illinois.

-SHAPIRO, Stuart.

1990. Computer Software as Technology. Doctoral Dissertation. Department of History. CMU.

-SCHERLIS, W. L.

1987. "Graduate Studies in Computer Science". CMU.: 41.

& SCOTT, Dana.

1983. "First Steps Towards Inferential Programming". July 1983. CMU-CS-83-142. Department of Computer Science. CMU.

-SHAW, Mary.

1989. "Workshop on Complex Software Systems". en Strategic Directions in Computing Research. October 11-13, Association for Computing Machinery & The Computing Research Association. New York. ACM Press.: 50-52

1990. "Informatics for a New Century: Computing Education for the 1990s and Beyond". CMU-CS-90-142. Computer Science. Carnegie Mellon.

1990. "Prospects for an Engineering Discipline of Software". CMU-CS-90-165. Computer Science. Carnegie Mellon.

1990. 25th Anniversary Symposium. SCS. CMU. September 23th-26th.

1991. "Great Ideas of Computer Science?". Mensaje electronico en ANDREW. 15 de abril. Mary Shaw@SHAW. VITRUVIUS.CS. CMU.EDU.

1991. Entrevista personal. 28 de mayo 1991. School of Computer.

-SIEWIOREK, Daniel P., BELL, C. Gordon, NEWELL, Allen.

1981. Computer Structures: Principles and Examples. New York. McGraw-Hill Company.

-SIMON, Herbert. A.

1957. "Somes Notes on the Early History of LT." (9 de julio 1957). Personal

Archives. Hunt Library. CMU.

1973. Las ciencias de lo artificial. A.T.E. Barcelona.

1975. Carta personal a Pamela McCorduck. 12 Agosto. Herb Simon Personal Archives. Hunt Library. CMU.

1977. "Choices and Paths: A Professional Autobiography". May 8. Personal Archives. Hunt Library. CMU.

1977. La nueva ciencia de la decisión gerencial. Buenos Aires. Librería "El Ateneo" Editorial.

1977. "Liberal Education in a Technological Society". Focus. vol 6. No.7. April 5, 1977.

1977. Carta personal a P. McCorduck. Sept. 27. Personal Archives. Hunt Library. CMU.

1980. "Computers Reveal Mind's Secrets" H.W. Pierce. Pittsburgh Post-Gazette. Mon., July 7, 1980.

1983. "What Computers Mean for Man and Society". en Simon H. Models of Bounded Rationality. vol 2. Cambridge, MA. The MIT Press.:186-201). 2nd printing.

1987. "Why Should Machines Learn?". En Michalski, R., Carbonell, J, Mitchell, T. Machine Learning. Los Altos, CA. Morgan Kaufmann Publishers, Inc. : 25-37.

1987. "Computers and Society". en Kiesler S. & Sproull, L. Computing and Change on Campus. Cambridge. Cambridge University Press.:4-15.

1987. "Ciencia Cognitiva: la más nueva ciencia de lo artificial". En Norman, Donald. Perspectivas de la ciencia cognitiva. Buenos Aires. Ed. Paidós, :13-39.

1987. "The Steam Engine and the Computer: What Makes Technology Revolutionary". Computers and People. Nov-Dec. 1987. vol 36, Nos 11-12 :7-11.

1989. "The Scientist as Problem Solver". en Klahr D. & Kotovsky, K. Complex Information Processing. Hillsdale, New Jersey. Lawrence Erlbaum Associates.: 375-398.

1990, Entrevista personal. Department of Psychology. 1 de mayo.

1990. Entrevista personal. Department of Psychology. 30 de noviembre.

1990. Entrevista personal . Department of Psychology. 17 de diciembre.

1990. "Problem Representation". En Rashid. (ed.). CMU Computer Science. New York. ACM Press. :449-463.

1981. The Sciences of the Artificial. Second Edit. Cambridge M.A. The MIT Press.

1987. "Guest Foreword". en Shapiro S. (ed.). Encyclopedia of Artificial Intelligence. Vol 1. New York. John Wiley & Sons.

1991. Models of My Life. BasicBooks. Harper Collins Publishers.

1990. Entrevistas personales. 1/5/90, 15/12/90. Dep. Psychology. CMU.

1991. "The Cat That Curiosity Couldn't Kill". Last Lecture Series. Tuesday, April 9. Charla en el Skibo de CMU.

& KAPLAN, Craig.

1988. "Foundations of Cognitive Science: Overview". Technical Report AIP-46. Department of Psychology. CMU.

-SMITH, Bruce, L.R.

1966. The Rand Corporation. Case Study of a Nonprofit Advisory Organization. Cambridge, MA. Harvard University Press.

-SMITH, Elliot Dunlap.

1948. "Professional Discipline Versus General Education As A Preparation for Citizenship and Cultivated Living". (16 de Marzo 1948). Papers of Carnegie Plan.

1946. "Professional Education in a Free Society" (February 14, 1946). Carnegie Institute of Technology: Educational Papers.

& SLACK, R.C., WARD, P.L.

1957. "The Role of Humanistic-Social Education in Making Professional Education Liberal at Carnegie Institute of Technology". Carnegie Institute of Technology: Educational Papers.

-SMITH, James Ward & JAMISON, A. Leland.

1961. The Shaping of American Religion. Princeton, N.J. Princeton University Press.

- SMITH, Richard Norton.

1986. The Harvard Century. The Making of a University to a Nation. New York. Simon and Schuster.

- SNOW, Charles P.

1978. "Las dos culturas", en Ensayos Científicos. México. Consejo Nacional de Ciencia y Tecnología.

- STAVENHAGEN, Rodolfo.

1971. "Decolonializing Applied Social Sciences". Human Organization. Vol 30. No4. :333-344.

- -STEEN, Lynn Arthur

1983. "Developping Mathematical Maturity". en Raston, A., Young, G.S. The Future of College of Mathematics". New York Heidelberg Berlin. Springer-Verlag.:99-107.

- STEFIK, M. & CONWAY, Lynn.

1982. "Towards the Principled Engineering of Knowledge". The AI MAGAZINE, Summer 1982: 4-16.

-STERN, Nancy

1981. From ENIAC to UNIVAC. Digital: Digital Press.

- SUCHMAN, Lucy.

1987. Plans and Situated Actions. Cambridge: Cambridge University Press.

- SWETNAM, G. & SMITH, H.

1989. The Carnegie Nobody Knows. McDonald/Sward Publishing Co.

-TARBELL, Arthur W.

1937. The Story of Carnegie Tech. Being a History of Carnegie Institute of Technology from 1900 to 1935. Pittsburgh. PA. Carnegie Institute of Technology.

-TEARE, Richard.

1980. "Teaching Engineers" Focus. Vol 9. No. 5, February 1980.

-TEVANIAN Jr., Avadis

1987. "Architecture-Independent Virtual Memory Management for Parallel and Distributed Environments: The Mach Approach". CMU-CS-88-106. Department of Computer Science. CMU.

-THOMPSON, Mary.

1991. Entrevista personal. 11 de junio. School of Computer Science.

-TRAUB, Joseph.

1981. "Quo Vadimus: Computer Science in a Decade". Communications of the ACM. June 1981. Vol 24. No6. : 351-358.

1990. "What Is Scientifically Knowable?". En Rashid. (Ed.). CMU Computer Science. A 25th Anniversary Commemorative. New York, ACM Press.: 489-503.

-TURING, Alan M.

1936. "On Computable Numbers, With An Application to The Entscheidungsproblem". Proceedings of the London Mathematical Society. vol 42, part 1. Nov. 1936. : 230-265.

1974. "¿Puede pensar una máquina?". Cuadernos Teorema. Departamento de Logica y Filosofia de la Ciencia. Universidad de Valencia.

1986. "A.M. Turing'S ACE Report of 1946 and other papers". Ed. by B.E. Carpenter and R.W. Doran. Cambridge, MA. The MIT Press.

-TURKLE, Sherry.

1984. The Second Self. Computers and Human Spirit. Simon and Schuster. New York.

-TYLOR, Edward B.

1977, Cultura Primitiva. Madrid. Editorial Ayuso.

- UPDEGROVE, Daniel A.

1986. "Computer-Intensive Campuses: Strategies, Plans, Implications". EDUCOM Bulletin, Spring 1986:11-14.

-VALDEZ-PEREZ, Raul E.

1990. "Machine Discovery of Chemical Reaction Pathways". CMU-CS-90-191. Computer Science. CMU.

-VEBLEN, Thorstein.

1965 (1921). The Engineers and the Price System. New York: August M. Kelley, Bookseller.

- VESEY, L.S.

1965. The Emergence of American University. Chicago: The University of Chicago Press.

-VOLKER, Paul A.

1978. The Rediscovery of the Business Cycle. New York. The Free Press.

-VON GLASERFELD, Ernst.

1988. "Introducción al constructivismo radical". En Watzlawick P. (comp.), La Realidad Inventada. Buenos Aires. Editorial Gedisa. : 20-38.

-VON NEUMANN, John.

1961. Collected Works. A.H.Taub (ed.). Vol V. Design of Computers, Theory of Automata and Numerical Analysis. : 288-328. New York. Pergamon Press.

-VON NEUMANN, John, &BURKS,A. , GOLDSTINE, H.

1946. "Preliminary Discussion of the Logical Design of An Electronic Computing Instrument". En A.H.Taub,(Ed). John Von Neumann, Collected Works. Vol V. New York. The MacMillan Company: 34-80.

-WALLACE, Anthony. F.C.

1972. The Death and Rebirth of the Seneca. New York. Vintage Books.

-VAN WILLIGEN, John.

1980. Anthropology in Use. Pleasantville, New York: Redgrave Publishing CO.

& FINAN, Timothy,

1991. "Sounding: Rapid and Realiabile Research Methods for Practicins Anthropologists". NAPA Bulletin. National Association for the Practice of Anthropology, a unit of the American Anthropological Association, 92 pp.

-WEBER,Max

1985. La ética protestante y el espíritu del capitalismo. Barcelona. Ediciones

Orbis, S.A. (1a ed. 1904-1905).

- WEINSTEIN, Alving S. & ANGRIST, Stanley W.

1970. An introduction to The Art of Engineering. Boston. Allyn and Bacon, Inc.

-WESTERBERG, Arthur.

1979. "A Review of Process Synthesis". Design Research Center. DRC-OV-14-79.

CMU.

1988. " Synthesis in Engineering Design". EDRC 06-47-88. Engineering Design Research Center. CMU.

-WERNER, Oswald & SCHOEPFLE G. Mark.

1987. Systematic Fieldwork. Sage Publications. Newbury Park. (2vol).

-WHITE, Leslie.

1949. The science of culture. New York. Grove Press.

-WIENER, Norbert.

1985. Cibernetica. O el control y comunicación en animales y máquinas.

Barcelona. Tusquets Editores. (1a ed. en 1948).

-WRIGHT, Conrad.

1955. The Beginnings of Unitarianism in America. Boston. Starr King Press.

Distributed by The Beacon Press. 305 pp.

-WULF, William A, & SHAW, Mary, HILFINGER, Paul, FLON, Lawrence.

1981. Fundamental Structures of Computer Science. Reading, MA. Addison-Wesley Publishing Company.

FIN.

NOTA: Esta tesis ha sido por primera vez publicada completa en el Web en el año 2001, respetando el texto tal como se presento en Tribunal de Tesis Doctoral, en la Universidad de Barcelona el año 1992.

Artur Serra

Centre d'Aplicacions d'Internet.

Universitat Politecnica de Catalunya.

03/08/01

