EmDeplo Morphogenesis

Ana Cocho Bermejo

Thesis submitted in the fulfilment of the requirements for the degree of Doctor .

Supervisor:
Ramón Sastre

PhD Program:
ARCHITECTURE TECHNOLOGY, CONSTRUCTION AND URBANISM. Barcelona Tech (UPC)

ETSAB-ETSAV
Departamento de Construcciones Arquitectónicas
Departamento de Estructuras
April 2012

To Pila, Jose, Juan, Ana B. My family.
To Ciril,my life mate.

Acknowledgements

This Thesis would not have been possible without the continuous support and encouragement of my family. To them, I would like to express my love and infinite gratitude.

I would like to express my sincere gratitude to my supervisor, Prof. Ramón Sastre for his support and for many good academical advises. This work has been extremely benefited by his rigorous reviews and comments.

Abstract

This thesis demonstrates the advantages of having a global+local behaviour versus a stimulus-reaction performance on the EmDeplo Emergency System ETFE façade.

The results obtained, after implementing the thermal relationship of the material in the façade performance, show, that a perfect optimization through a ANN and specific GA for each scenario is maximizing the insulation performance of the pillows of the system.

Using the Media-Tic building as the latests stimulus-reaction example built, the thesis argues that advantages of (environmental) adaptability are not currently being used by nowadays architects, forgetting to use "time" as an important parameter in Human Oriented Architecture, and, thus far not obtaining environmental efficiency when using these type of designs.

THE MANIFESTO OF COMPUTER ART

I.
Computer Art has not come into existence yet.
That's exactly why we have to write, talk, and think about it, to call it into being.
The computer was not invented for us, the artists. The computer was made for military purposes, it hasserved scientific purposes, and when a flicker of hopefor artistic use appeared for the first time, it fell prey topropaganda and commercial film making straight away. In order to create art with the computer, we will haveto cast off all clichès of present commercial forms.
The computer is just a means. We are wrong if we want to use it to conceal a lack of vigour in our message under a more fascinating guise.
If we just use it to make our work easier, we give in to our innate idleness.
We have to be fully aware of ourselves and of the world around us to be able to use it adequately.
People say the computer will transform the world. By this they mean the same as what they proclaimed about radio and television; that communicating even huger masses of information can only be a positive line of development.
In reality, though, the computer can be one of the most effective means to increase the danger of war, the stress on mankind. It is also a means for further manipulations in the mass media.
The computer is a typical example of an instrument Man has created and now does not know how to use.
We have to face up to the fact that the computer will by no means change Man, the decisive unit of our world.
We can make another go at the eternal subject, perhaps shooting the film from a slightly different angle.
Taking a picture of an object with this new medium might bring to light some details never seen before, and the ones known might be put into new perspective.
If we approach the computer with our old way of thinking, grounded on old means and devices, we will be knocking our heads against brick walls and miss a magnificent opportunity to create a new world.
(…)
Traditional art trade, criticism and art history have not yet built walls around this kind of art. It is in our hands how we will shape their future relations.
(…)
The computer can become a new means to understand the world. Long forgotten knowledge about geometry, mathematics, logic and about the thousand fold forms of reality might once again come to light.
Science and art can be joined again.
We may lose the unique irreproducible stroke of the artist, but we will gain a new way of thinking. Perhaps we will realize that the artist's stroke might not even be of such value, and that works of art have always shone through with their content.
We can become aware of the working of hidden mechanisms. Using a drawing programme, I can become conscious of the subconscious processes which direct my hand when painting.
(…)

III.
Working with the computer can be a pleasure. It is an extremely interesting invention, a source of experiences which could never be accomplished through any other means.
(…)
The new method of projection alters the dramatization.
(…)

IV.
Let us establish a much better relationship with programmers. They become our co-workers even when we buy programmes ready-made, and our work will bear their initials as well.
Programmers develop software for us to use. Unless we know the capacities of the computer, we cannot sufficiently formulate our needs. Unless we sufficiently formulate our needs, we will receive nothing but traditional replies, programmes following the usual logic.
Electronic brushes will then follow in the footsteps of traditional brushes, instead of finding their own special ways.
(…)

V.
Only in very few works made with computers can you be enriched and feel the artist's desire to conquer the world, the desire to create a new work of art with a new medium.
We could even take an example of computer drawings made for purely scientific purposes: the motive is obvious, the demand for clarity of thought compels the scientist to create a simple and precise formulation of his ideas. Our aim is not science but creation.
Artists' responsibility is the responsibility of those who create signs; the signs we leave behind will make people of the coming centuries know we have lived and thought.

Budapest, 15 January 1989
Tamás Waliczky

[The *"Manifesto of Computer Art"* was performed first time at the IMAGINA festival (Monte Carlo), and published in the catalogue of the DIGITART II. Exhibition (Budapest)].
(see # 18, Waliczky, 1989)

Contents

1. Introduction

These thesis will argue about the importance of Dynamic Parametric Architecture versus Static Parametric Architecture. Describing the architectonical and the algorithmic context within which the Emergency Deployable System emerged, it will be discussed the importance of adaptability as the missing concept of parametric architecture.

Developing the concept of Human Oriented Parametric architecture, it will be discussed the need of implementing time as the lost parameter in current design techniques.

Using the media-tic building, of the Spanish architect Enrique Ruiz Geli, as an example of a current design that tries to implement new technologies and parametric ideas in its design process, it will be explained the idea of the need of the building of working with the environment, not defending against it as the basis for a good adaptability. Geli's design is currently using 104 Arduino chips for an individual control and performance of the 104 ETFE pillows of the building façade.

Through the creation of a virtual stimulus-reaction model of the system façade versus a model in which machine learning have been implemented for a better environmental performance, the efficacy on insulation of both models will be compared.

Morphogenetic processes idea will be discussed through also the principle of an adaptable membrane, as the thought solution for future architecture design processes improvement.

A model implementing a unique Arduino on the façade, will control the performance of the façade patterns, through, an Artificial Neural Network that will decide the kind of scenario the building is in, activating a Genetic Algorithm that will optimize insulation performance of the ETFE pillows.

The final virtual model will be able to obtained the goal proposed, for this thesis, a homogeneous temperature in all the spaces of the building of 22ºC. The maximum thermal optimization obtained, nevertheless, appears if the opening of the pillows is free within and interval of 0 and 1 m thickness. The constrains of the opening of the ETFE pillows to three positions, will be demonstrated more effective than a just stimulus-reaction behaviour, but also, much less effective that an unconstrained façade system.

The EmDeplo System will work with a Global behaviour, pattern performance of the façade, but also with a local behaviour for each pillow, giving the option of individual sun shading control.

Machine learning implementation will give the façade the possibility to learn from the efficacy of its decisions through time, eliminating the need of an on-off behaviour for defending against the environment. Instead it will work with it, adapting to it, and evolving with its variabilities.

2. Context.

2.1 Algorithmic context. The scenario in which EmDeplo emerged.

Dynamic Parametric Architecture vs. Static Parametric Architecture

" The contemporary architectural style that has achieved pervasive hegemony within the contemporary architectural avant-garde can be best understood as a research programme based upon the parametric paradigm. We propose to call this style: Parametricism. *Parametricism is the great new style after modernism. Postmodernism and Deconstructivism have been transitional episodes that ushered in this new, long wave of research and innovation." (see # 148, Schumacher, 2008)*

The contemporary debate about what parametric architecture is currently clearly divided, in various trends. Some of them consider parametric architecture as a new style, as Patrick Schumacher defines it, and other trends just missunderstand the idea of the possibility of a complex system of relationships, (with which we are currently able to deal with due to the latests computation improvements), making an effort just in obtaining an optimum static 3d shape.

"*Parametric design is a method of intelligently designing architectural objects based on relationships and rules using the computer. These are defined in parametric software and are easily manipulated to quickly generate multiple iterations of the design in 3D."* Adel Zackout.

Generalizing the definition, it is possible to obtain a description about which, most of the trends, agree.

"*Parametric design is based on object relations not on fixed metric quantities."* Michael Meredith *(see # 60, Meredith, 2007)*

But in all of these definitions a basic quality of parametric architecture is missing.
What is missing in architecture is to have real-time movement and adaptability.

"A new trend is emerging in architecture today: dynamic and time-like architectures (a term derived from the language of Theory of Relativity) that are capable of moving, flexing and reconfiguring themselves throughglobally networked control mechanisms are emerging." (see # 149, Senagala)

There are two kinds of parametric architecture:
- Static
- Dynamic.

The first kind obtains its form and configuration from several parameters that are not going to vary during the time of the building existence.

The second kind, the one believed with a more enhanced future, is the architecture in which the basic inputs vary during the building life time.

Nowadays, this second kind of building is not currently buildable. Not only because its complex dynamic structures, but also because it would have a very expensive cost.

Through the next chapters this thesis develops the idea of *Dynamic Parametric Architecture*, (from now on mentioned as *DPA*).

Static Parametric Architecture (SPA) is nevertheless the kind of parametric architecture that is actually being built, considering SPA, an architecture that is just obtaining a parametrical shape from scripting.

This thesis, makes a first approach to DPA proposing:

"EmDep Morphogenesis. Membrane environmental learning & control"

When we speak about architecture, considering the algorithm is the design process, the parameters use to be considered the original inputs, and the outputs use to be,the final architectonical form: A parameter is basically a value which is going to be used as an input in the algorithm to obtain an output shape.

This thesis considers this approach highly inappropriate.

What is missing in architecture today is that architects are not really taking advantage of the adaptability possibilities that this new algorithmical approach offers.

Some critics like Sanford Kwinter really think that Architecture has started to disappear as a discipline as we are entering the Algorithm Era. A new kind of design has born, the one without distance between engineering and invention.

In this sense Kwinter remarks two kinds of historical changes. First, the ones that just are routinized forms of innovation marking the routes of technological, economic and scientific realities. And secondly, the important ones, "*those break of fully geological magnitude out of which systematic human transformations became genuine possible*".

When the idea of topological space is presented, we are basically thinking about natural processes with natural set of instructions.

Adaptability appears as the idea of responsiveness to a changing environment.

The main aim is, currently, not the basis of the natural processes, which is adaptability, but to be able to map natural processes just to repeat those behaviours at the computer to apply them and get a beautiful shape.

"It is erroneous to think that we can just copy biology patterns in architecture; we absolutely can't and it is dangerous."

Cecil Balmond.

Architectural scene is just creating biologically inspired forms, when instead, it will be much better to understand the behaviour of the system in terms of material and changing data from the environment.

In the current parametric scene we are just imitating natural processes to fill or create spaces, but, really, we are not defining the real new kind of space that we need. We are just using natural processes to obtain images, 2D or 3D, and to fill spaces making them to have the appearance of being parametrically human designed.

The copy of a biological pattern alone does not guarantee a goal of a successful design,so:

Why are architects just currently imitating natural processes of pattern generation?

The answer is quite easy: we have not really realized yet which should be the basic human and environmental related basic parameters of our living spaces.

The term that this thesis will like to introduce is: a *Human Oriented Parametricism (HOP).*

As the basis of this kind of architecture it will be placed the human. The human and all the parameters that can be related to him and the environment, and that can be base, of the Dynamic Parametric Architecture.

So that, the Emergency Deployable membrane will be an example of Dynamic Parametric Architecture.

"Design should attack the entire world: its task is to produce advantage over adversity or hazard".

B. *Fuller.*

(see # 3, Kwinter, 2008)

This Version 2.0 of the membrane will be supposed to be able to change configuration on site depending on the different environmental situations. Trying to apply some DPA basic

parameters. Sample of them will be, the temperature and sunlight control of the cellules, done through some model trials along the next chapters of this thesis.

Complex Systems. Morphogenetic Processes.

"Form-making is the action of a series of subsystems, interlined but free enough to adapt independently"
(see # 107, Alexander, 167) p.42

No complex adaptive system will succeed in adapting in a reasonable amount of time unless the adaptation can proceed subsystem by subsystem.

Nowadays we are able to record natural processes that we weren't able to record before. We are able to digitize and map, ability that has changed our way to measure and represent space. Today morphogenetic processes are being used as methodological models.

"*Morphogenesis (from the Greek morphê shape and genesis creation, literally, "beginning of the shape"), is the biological process that causes an organism to develop its shape."*
(see # 10, Dictionary., 2010; 68, Wikipedia, 2010)

Natural systems are complex and adaptive, and also, self assembled. Nevertheless it is being missed in architectonical use of its basis for design their adaptability property. Property that is their base of existence.

"Formalism demonstrates first and foremost that form is resonance and expression of embedded forces. (…) The manifest form is the result of a computational interaction between the internal rules and the external (morphogenetic) pressures, those themselves, originate in other adjacent forms (ecology). The internal rules comprise an embedded form what is today clearly understood and described for the term Algorithm".
(see # 3, Kwinter, 2008)

Historically, the concept used for design was top– down engineering. The material solutions just came after defining the shape. Now it is being proposed that materials and structure should not be treated as separate elements.

The idea is to leave behind form-vs.-function debates. Formation and materialization process are inherently and inseparably related, so they should not be separated.

Recently architecture has been presented as an ecology, involving dynamic and varied relations and mutual modulation between material systems, macro and micro environmental conditions, and individual and collective inhabitation.

Understanding then, Ecology, as *a part of biology that studies the distribution and abundance of living organism and the interaction between the organism and the environment."*

We are walking from drawing surfaces to setting rules of interdependency *–genotypes-* leading to a potential differentiation *–phenotypes-* .

In botanic, Phyllotaxis is the arrangement of the leaves, buds, thorns and soon of the plant, according to environmental influences. This kind of growing process that can help us to understand the material distribution of a form towards the environment.

The study of multiple natural processes and their behaviour has just started to help us to understand the new definition of architecture. Architecture as a complex process, a set of interdisciplinary relationships new media based.

Lately more examples are appearing. Douglis work along his teaching experience presents us the idea of autogenic vs. allogenic successions:

"In ecology, an allogenic succession describes a succession where the stimulus for change is an external one.(…) In ecology, an autogenic succession describes a succession where the stimulus for change is an internal one. For example gradual soil improvement could allow a new species to develop."

These kinds of processes are based in the four basic characteristics that a self organizing process possesses:

"- stimulus

- Sensibility

- Sensivity, to react to the stimulus

- Irritability"

(see # 8, Various, 2006)

Ways of understanding morphogenetic processes are really infinite, so, trying to explore them for understanding our architectural Organism is just a path to reach its behaviour comprehension.

"Self organizing processes underlie growth of living organism. The internal organization of the system adapts to the environment to promote a specific function without being guided or managed from the outside. (…)A systematic change is in the horizon where the boundaries between the natural and the manufactured no longer exists"

(see # 8, Various, 2006)

EmDeplo proposes an intelligent skin, locally and globally adaptable, a living building skin that helps the spaces to work with environment variances instead of defending against them.

Operative Diagram vs. Typological Model

We are building diagrams, not buildings. We are clearly not differentiating form and shape. Architects do not understand the difference between parametric architecture and parametric geometries.

Should the knowledge of the tools be the principal determining concept of the architectural practise?

Is this idea new along history in architectural design?

Should be these "new tools" the basis of all of our spatial ideas?

Since CAD architects concern about the loss of control over their own designs. Design is strongly dependent on the tool used. The problem arises when the tool is not totally under control of its users, so the results, might be unexpected.

"FORM IS NOT AN END IN ITSELF.(…) It is the distant expression of a complicated set of interacting dynamics whose roots are nowhere else built than in the social, historical and material world of the human.(…) The hunters for form: (…), young narcissists stricken with computer puppy

love, who believe in magic (and software faeries too no doubt), and are totally enthralled with whatever the beloved machine digitally excretes.(…) ARCHITECTURE AS AN ORGANON: A SYSTEM OF INVESTIGATION, INVENTION AND TECHNIQUE."

(see # 3, Kwinter, 2008, Page 27)

Architecture has become a set of multiple processes related to its social, economic and human parameters. And sometimes, we wrongly believe that architecture is just one of the processes inherent in it and not the full system.

We have Models (as representation of real) vs. diagrams (abstractions of the real). Due to the previously mentioned error, we are just currently building diagrams.

"Form is not shape: it is something dynamic almost intangible. Form is a tension between volume and surface: It is the rhythm of the conectivities."

(see # 51, "Cross-catalityc architectures: In conversation," 2007)

Building becomes a complex system and form is the hybrid result. We should understand a model as a representation of the real or a part of it in any scale. These new methods have destabilized the essential status of the architectural image-object.

Paul Virilio mentioned that technology derives a hybrid, so, nowadays, we are living the hybridization of the architectural image.

Today it has occurred a displacement of the traditional aesthetic principles to judge an architecture image. As Tierney indicates, we have not just one point of view or orthographic projections, we currently have a dynamic point of view.

Classical philosophical though understood architectural image as fixed, now, evolutionary form generating software transform architectural image into event and performance.

Designs are engendered by their imaging techniques.

In parametric design, architecture will be shown and mapped as a network of relations of events that occurs in different ways and places.

As the current belief that everything can be mapped and stored, we are living the birth of a new profession: the data indexing.

So it sounds quite logical that, if we are speaking about Dynamic Parametric Architecture, in that situation, we will understand mapping as the idea of a continuous function. But, to understand it better, let's make its definition just inside a topological space: so let's think about mapping as a family of sets with certain properties that are used to define a topological space.

"A Graph is an abstract representation of a set of objects where pairs of objects are connected by links. The interconnected objects are represented by mathematical abstractions called vertices, and the links that connect some pairs of vertices are called edges."

(see # 14, Dictionary, 2010)

Graphs are not interested in scale but in relations.

In this sense they represent important *shift: from the map (topographical) to the diagram (topological)*.

Topology is a philosophy of relationships. Architecture today has become hybridized with other disciplines so topology is useful as a tool to map previously invisible relationships across this multidisciplinary boundaries and also how this relations work abstractly.

We should understand that the diagrams are a method not a model. A model test a hypothesis, a diagram is not an end itself and can produce unexpected results.

A simulation understood as an anticipation of the future through known processes.

"Historical trajectory of architectural image is linked to vision developments. (…) Each new technology delivers its own mode of perception, which ultimately frames and constrains the visions that are produced."

(see # 9, Tierney, 2007)

A type of digital formalism has resulted from the use of modelling software.

The model problem-function-solution has been substituted for information-field- interaction model as Kwinter suggests.

The complexity of the design process

Is intention necessary in design? Who designs?

Design used to be a decision-making process, a combination of thoughts that lead to the inception of form.

What if unconscious agents take decisions?

This is the basic doubt, the non total control that adaptability proposes. On the other hand, if we do not know completely the tools of the process: are we completely deciding the final result?

"Architectural design is a much more complicated process than any other design process because it entails factors that cannot be codified or predicted".

(see # 65, Terzidis, 2006)

While complexity may be a characteristic of many natural systems or processes, within the field of design, the study of complexity is associated with artificial, synthetic, and human made systems. Those systems, despite being human creations consist on parts and relationships arranged in such complicated ways that often surpass a single designer's ability.

Because of its quantitative nature, the study of the complex, involves by necessity computational methods.

"To think is to create, when we reach thinkable processes they will create themselves apart from doing was they were created for doing. A potential for unpredictable randomness would also have to exist. (…) If a system is close the result is predictable. Virtuality is an open relational system. It is constitutive of form but not limited to form."

(see # 9, Tierney, 2007)

We have to consider a new kind relation decision-intention. Quantitative nature of computation vs. abstract holistic nature of human thinking. Design is then now virtuality, not actuality.

"Such theoretical design models negate computation as a possible means for design realization mainly because it is based on discrete processes that are finite and, as such, restrictive. In contrast, human thought appears to be continuous, infinite and holistic…"

(see # 65, Terzidis, 2006)

Greg Lynn explains the plasticity of computer forms linked to Deleuze smoothness and continuity, as if software is associated with softness.

In Terzidis opinion, no truth at a mathematical level exists, as polynomial based curves or surfaces, exhibit a continuous and smooth behaviour only when implemented in a computer system. It is the numerical representation, processing power and resolution what makes the plasticity possible.

Algorithms are not the end product but a vehicle for exploration. They are abstract and universal mathematical operations. Some relationships between numbers and concepts appear to be to some designers too deterministic.

The problem currently is that we are not able of adding the time parameter to the models, to the ones currently used for obtaining the parametric geometry, so, when we pass to the real, everything collapses: Adaptability disappears.

Emergency Deployable will have the opposite problem: the need of physical models because, it does not exist, any software to 3dmodel it and to calculate its real behaviour.

Nevertheless, it will be possible to propose an adaptable system for it, including ideas as timing and environmental characteristics.

Emergency Deployable Health System

In the middle of all of these ideas, the idea of a parametrically based heath system was born. As a mixture between parametric urban space and human enclosed space, the Emergency Deployable Health System is defined as an intermediate system, complex and alive which parameters bases are clearly human and environmentally linked.

When [An_D] started designing the system it was obvious that it was defined by its morphogenesis. Emergency deployable is a complex system defined for its process to obtain its adaptability to the environment and also the system final complete form.

For us the system shape was completely different from the system form. Form was the final complete definition from the factory to the end of scenario deployability and shape was, and just was, the shape the membrane finally adopted on site.

It was designed through a developing and controlling process. Thinking about the relationships between patient/doctor, human/environment, the functional needs, the timing of the actual situations and the technological problems, it was proposed a new concept of health resources distribution and a new idea of health unit.

2.2.A. The Media-TIC disaster .2009.



Figure 1

Architect: Enric Ruiz Geli – Cloud 9.
Structure: Boma S.L. Agustí Obiol.
Building Mechanics: PGI Grup, David Tusset.
Budget: 20.791.486 €.
Built: 2009.
Built Area: 23.104 sq m.

"Now, in the age of the Information, the architecture has to be a technological platform, being the important things, the bits, the connectivity, the new materials, the nanotechnology... The connexions are more important than the weight of the materials. It is a question of an electronic, immaterial world, in which what imports is the drawing of the network and not of the gravity"
Enrique Ruiz Geli, Cloud 9 website.

Based on Gaudi's method for finding proportions? The idea of the "Digital Pedrera".

One of the main aims that Ruiz Geli mentions in all presentations of Media-TIC building, is, the idea of building "the Digital Pedrera".

In his own words the dimensions of the superstructure of the building are based on Gaudi's work.

The "superstructure" of the building, a hanged portico of 9 floors, has theoretically its origins in Gaudi's method.

"Form finding as a design method uses self organization of material systems for developing structural form in response to gravity:

- Generation of the form to be build

- Full-scale construction of the desired form."

(see # 62, Various, 2004)

This theoretical relationship is quite unclear. Apart from a possible misunderstanding of Gaudi's hanging catenaries models, it can not be found a real consideration about his methodology.

The architect mentions the "Gaudi Scenario" vs. "Now Scenario". He defends the idea of the team working for an "Energetic Beauty" instead for the "Modernist Beauty" from Gaudi's years.

Physical form-finding experiment in architecture is thought to have begun with Gaudi and followed by Frei Otto. In the 60's as part of the group "*Biologie und Bauen*" (Biology and building), together with the biologist and anthropologist Johann-Gerhard Helmcke, Otto established the basis of adaptability as a form-finding result. Evolutionary algorithms were used for optimization; now, we should be really interested in them because their adaptive characteristics.

Relationship biology/maths was established at the beginning of the 20th century by Whitehead and D'Arcy Thomson. Whitehead considered the idea of process, instead of the idea of substance as the basis of the world.

Presenting in 2011 the idea of the human relationship with the environment as stimulus-response behaviour, instead of an adaptive learning process, is, not only a bad praxis, but also a misunderstanding of what morphogenetic behaviour is, what technology should drive us to, and about the resources used at the building.

On the other hand, the references used by Ruiz Geli, AEG Behrens Turbine Hall and Saulnier's Menier Factory, clearly define his understanding as the building as a "container", a concept that contradicts all the basis of Gaudi's theories.

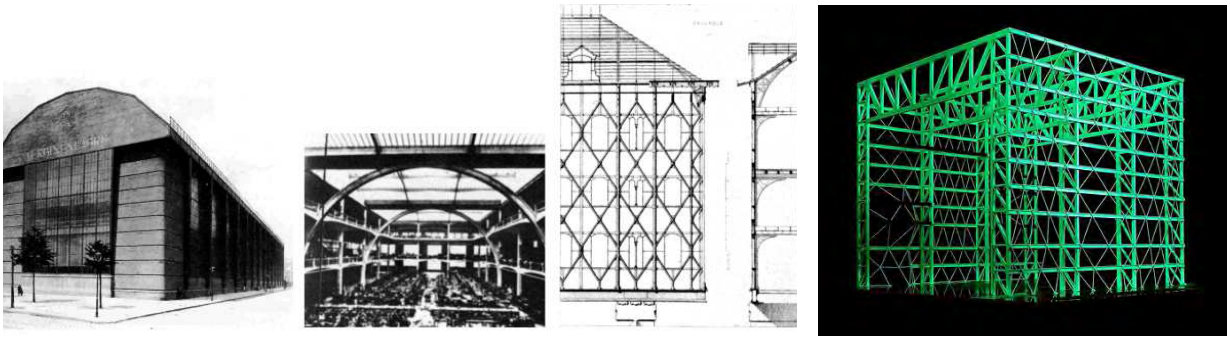


Figure 2 , AEG Behrens Turbine Hall , Saulnier's Menier Factory, and Media-tic structure.

Gaudi's principal inspiration was nature, understanding and conceiving building as global systems. Taking care of all its parts and interactions. He believes that structures as the bamboo, are the most perfect structures, because they are not only functional but also aesthetically beautiful. He conceived the "Sagrada Familia" as a forest, as a living organism of tress that has its branches and sub-branches to support their structure. He admired nature's adaptability.

Mentioning Gaudi as the basis for the monumentalized structure of this building, and, for the proportions election, seems not only quite inappropriate, but also shows a complete misunderstanding of Gaudi's thoughts and a superficial approach to his work.

Antoni Gaudi created an organic architecture working always in a 3d mind, trying to escape from the traditional vertical vs. horizontal flooring of the architecture till those days.

Media-TIC is a monumental container of a plain steel structure with perpendicular 2d flooring planes inserted into it. Obviously not conceived as organic but a rigid and static shape, the building is really far form Gaudi's ideas.

Gaudi looks for the equilibrium, dissolving structure with the rest of the building, creating a unique organism. The huge 14 trusses, clearly separated form the rest of the elements of the building of Cloud 9, are evidently not designed for dissolving with the other elements and parts of the building.

Media-TIC is superficially thinking about Gaudi's organicism & about what adaptability is.

Figure 3 , Media-tic façades.

Adaptability?

The media-TIC is supposed to "adapt" to the environment saving energy. I do consider that adaptation is wrongly defined by Geli when introducing the building to the audience.

When speaking of adaptability we are relating our ideas to the natural behaviour known in animals and plants to live and change together and depending on the environment around them. We are speaking about the property that natural complex systems on earth have, of being emergent, not only in form, but also in many other behavioural aspects

"Buildings not as fixed bodies but as complex energy and material systems that have alike span, and exist as part of the environment of other buildings, and as iteration of along series that proceeds by evolutionary development towards an intelligent ecosystem"
(see # 62, Various, 2004)

Natural systems are complex and adaptive. When we speak about adaptability, the relationship with the environment becomes fundamental.

Ecology defines the relationship between human groups and the environment.

Trying to define the adaptability of the building from its environmental performance, as the architect did when presented the building for the first time, is, not only, reducing, minimizing and wrongly defining the relationship Ecology-topology.

These terms are extremely linked and are the basis for the self-organization and self-adaptation of organic life.

The quality the architect mentions the building has *"to have the capacity of adaptation in the presence of an environmental change",* is completely inexistent.

The building has 104 Arduino chips to control 104 ETFE pillows with a total of 2500 square meters of façade. These chips behaviour can be changed via an IP address. Every pillow has a light sensor and a light meter that determine its behaviour. Behaviour that is not global, it is, in fact, local.

The façade is not adapting to the environment, it is just reacting to the sun opening the patterned membranes. Stimulus reaction. Not intelligent working or behaving. They are just working in a very static way and behaviour, to the environment and weather around the building.

They are working as a wall, a wall to defend, a wall in the antique way of understanding architecture, architecture as a castle to protect the human against the aggressive world.

The building is not working hand by hand with the environment, it is not understanding nature's logic, and it is not a system with the environment as a parameter in it. It is not learning from it, they are not working together,

It is not adapting, it is defending itself.

In 1926 Eveling defended that, up to then, the buildings have been a defence against weather, climate, and human failings as thieves....

He defined the living plant as the new primal symbol of architecture, a biological architecture. For him, the building is a growing organism whose base is planted on the ground.

The principle of membrane.

Eveling manifesto used to be considered within the Bauhaus movement, but, in fact, his writings never were part of the Bauhaus collection. He confronted the main strain theories and predicted that it would not be after 20 years when architecture scenario will understand his idea of membrane.

He was right. In fact, his membrane manifesto was this thesis main inspiration.

Eveling's membrane does not distinguish interior and exterior, it is "*a permeable threshold*".

"An inorganic type of membrane: an envelope that both protects a body from the outer world and structures its interior life and organization. (…) The space-membrane combines the plasticity of an animal organism with the metallic fortitude of architecture. "

(see # 138, Eveling, 1926)

Figure 4 , (see # 97, Eveling, 2010 (original 1926))

He ilustrated his membrane definition cleartly:

"The living cube: wohnkubus, is the centre: it represents an ideal living cell.
Four sides equal in size and in capacity to receive and transmit energy from/to the environment.
They are transparent but, there is one red that, maybe, indicates changes in temperature, humidity or light conditions.
The cube floats over the earth which is represented as a field of radiant forces rendered as red and black stripes, expanding like waves.
Diagonal lights signify the lights that connect the earth with the atmospheric envelope.
+ and – means de poles of the electromagnetic environment and its ionised ambience.
Inside the cube there is a dark schematic human figure and it is also, much bigger and transparent outside the cubic enclosure as a mere outline.

THE BODY IS ENVELOPED BY THE LIVING CUBE.

The true Eveling's living cube is the vegetable. "

(see # 97, Eveling, 2010 (original 1926))

Geli's media-TIC conception of what a permeable defensive façade is contradicts not only such an old manifesto, but also, all self-organizational processes, dynamic and adaptive, that have been lately demonstrated as a probable future of our conception of architecture.

He is using the technology of the future, but, implementing it as in the very ancient past.

Self organizing processes underlie growth of living organism.

The internal organization of the system adapts to the environment to promote a specific function without being guided or managed from the outside.

His concept is not efficient;

It is just a show off of technologies supposed knowledge.

And, what it is in fact, is a demonstration of the non comprehension of the origins of today technology possibilities and the extremely advantages that can be reached in architectural design through a good study of a façade system process and performance.

Energetically efficient?

The EFTE layer is presented as a second layer of the façade, separated two meters from the curtain wall.

Geli argues that he is obtaining a 20% of energetic saving.

Saving that, is basically obtained, by the decrease of use of air-conditioning systems. Decrease that is basically taking place, not for the insulation properties of the ETFE pillows, but for their positioning as a second layer separated form the main curtain wall, and, on the other façade, because of the implementation on the liquid nitrogen.

It is true nevertheless that the printed patterns on the membranes get a really good sun-shade light control. Controlled by a huge number of light meters, they "react" to the environment defending the interior of the building from the ultraviolet light.

One of the main qualities of the ETFE membranes is that it is possible to reduce UV-C in an 85% being UVA permeable. In my opinion this is the only part of the concept of the building that is conceptually well defined.

Media-TIC location in Barcelona makes the building receive direct sun-light. So that, there are two façades that are receiving more than 6 hours a day of sunlight.

These two façades are solved in two different ways. One with the ETFE pillows second layer, and, the other one, with a second layer on it that eventually creates a cloud of nitrogen.

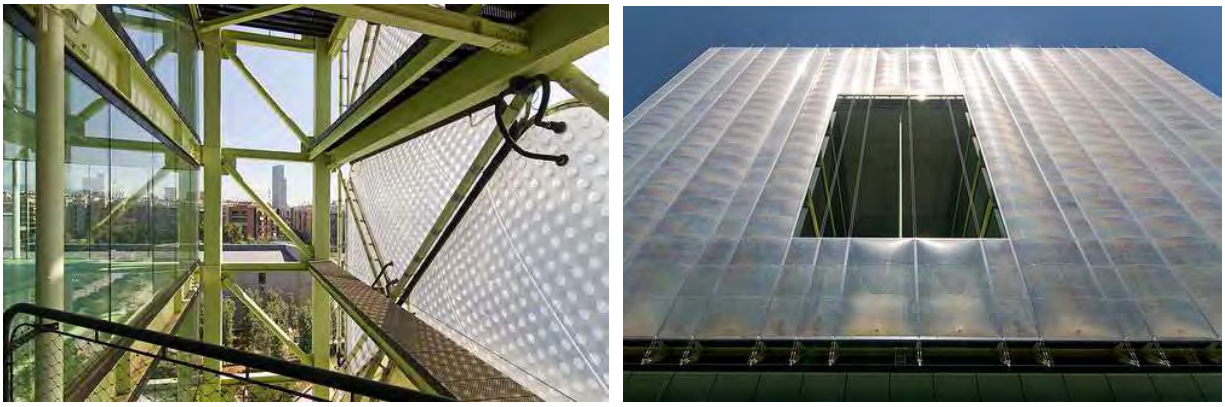


Figure 5, pillows and nitrogen layer façades.

Two aspects should be considered:

- Shading control
- Insulation control & environmental efficacy.

Sun-shade control of the façade is basically done by the nitrogen cloud and the pattern layer of the pillows. I consider it really effective.

Due to the characteristics of the climate, Mediterranean buildings use to include second layers on they façades for protecting from direct sunlight. We should remember Le Corbusier brise-soleil. It was so effective that has been reinterpreted in many different ways till nowadays.

Media-TIC is doing its own reinterpretation of the brise-soleil.

It can be discussed the kind of gas implemented in the cloud. Argon, for example, or any other inert gas will have performed better but, as the cost must have been an important argument, this discussion is quite subjective depending on the goals needed. In any case it is getting a reduction from solar factor 45 to solar factor 10.

On the other hand, the façade with the pillows performance is for me a completely different argument.

ETFE pillows are really effective in insulation and improving shading. Nevertheless, the only aim in this building was the shading effect as, the layer, is completely separated from the building.

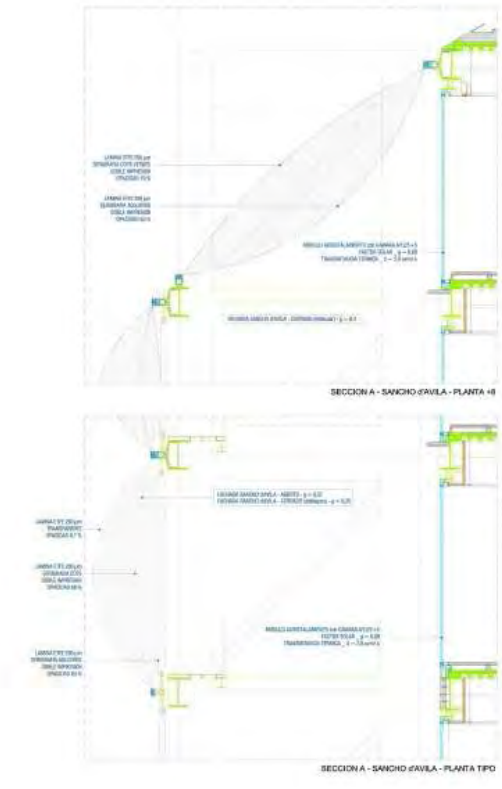


Figure 6 , *(see # 97, Eveling, 2010 (original 1926))*

So,

would it have been possible to obtain the same sun protection with only the patterned layer?
Why do we need to install the whole pillow mechanism if it is not being used properly?
Is it just a show off of technology because same goal could have been reached cheaply?

If the objective was just the shading, the ETFE pillows are inappropriate. They will not insulate, neither in summer nor in winter. They just will be used for giving shade in summer.

Shading in that kind of climates is fundamental for living. Nevertheless they are many other options, much more appropriate and cheap. Choosing these methods demonstrates just the need of trying to show some technology knowledge. But, in the end, the result is that the building is showing exactly the opposite:

A complete un-knowledge of the system they are implementing.

The global efficiency of the building is working under Kyoto's protocol of minimum CO_2 emissions. The building is saving 114 ton of CO_2 a year. Which is basically saved, by the other systems implemented.

Figure 7

The solar panels of the roof and the sea water used for refrigerating, as well as the vegetal roof, are the main help to improve the interior temperature of the building.

The nitrogen façade works quite well, but ,the results obtained with the pillows façade could have been obtained with any other shading system.

Figure 8, building diagram of energetic efficiency.

Media-TIC has been sold to the media as the newest technology implementation, and it is, just a bad example of how to implement it.

Arduino chips are being used just as on/off switches to turn on the shading. Any kind of simple mechanism, if that was the behaviour wanted, would have worked better to turn on the system. The stimulus-reaction performance on the façade does not need any chip implementation.

The inexistence of learning for advance performance makes the Arduinos not useful. The environmental performance could have been much better implemented with an intelligent behaviour of the façade system.

Just one chip will be needed for controlling the façade patterns and the learning procedure, and that are exactly the kind ob behavioural patterns that EmDeplo wants to implement.

Examples of real energetic efficiency. The bubble green house.

One very interesting example that really considers environment and is not really worried about technology show off is "The soap bubbles green house".

The initial idea was to try to improve the behaviour and efficacy of the green houses that are in some regions under -30ºC during winter or more that 40ºC during summer.

Solar Roof is a system invented by Richard Nelson, a researcher in solar building behaviour. The system will use a liquid composed by 1% of soap and 99% of water. The soap bubbles will be distributed with a pump between two transparent layers, each of them, with a a thickness about 6 mm, diameter of the bubbles.

This kind of study represents exactly the opposite behaviour in design. The real worry about efficacy without being worry about showing "new" technology that is being not effective.

Figure 9, bubble system, internal view.

The pumps only need for generating the bubbles a 2.5-5% of the usual climate control expense for heating and cooling.

The bubbles need to disappear at least two hours without pumping so, the worse part of the system are the problems to make the insulation disappear quickly. Nevertheless, Nelson indicates that, if that need is fundamental , it is possible to install some fans to help the bubbles to disappear quickly.

	Double Walled Greenhouse	**Soap Bubble- Insulated Greenhouse**
Package including frame, double poly roof cover, inflator fan, end covers, spring lock	$4,084	$4,084
Heater	$2,000	$1,800 (smaller size)
Poly Tank for soap solution		$200
Bubble Generators		$7,800
Soap solution		$200
Pump / pipes		$400
Ridge Beam		$600
Foundation		$1,000
End wall framing / Insulation	$600	$1,000
Total cost to build	$6,684	$17,084
Difference in cost to build		$10,400 more expensive
Cost to heat / year **PROPANE**	$1137.50	$146.25

Cost to heat / year **OIL**	$1206.85	$155.17
Cost to heat / year **WOOD**	$543	$70.50

Figure 10

Obviously, the price is a problem nowadays in every new system implemented. The building cost is highly expensive compared to the normal cost of a green house. The designer argues that this cost will be recovered in 10 years of building life due to the heating-cooling savings, which is true.

Nevertheless the real efficacy of the system will be reached when a serial factory fabrication can be implemented and, the now big initial investment, disappears.

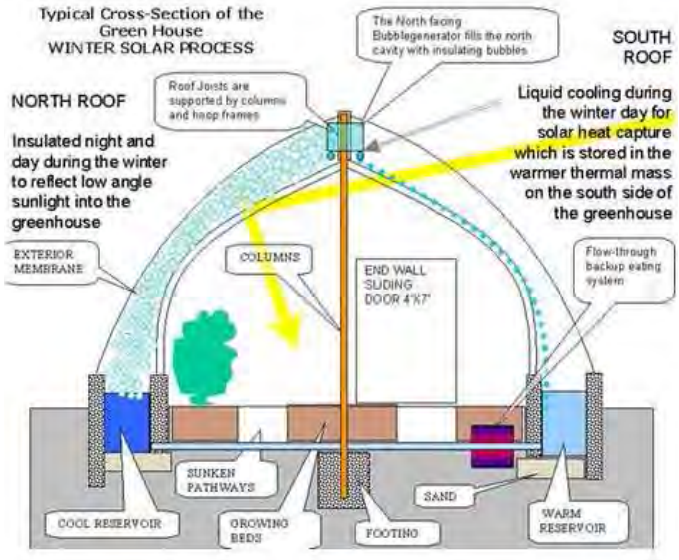


Figure 11

2.2.B. Arab Institute. Jean Nouvel 1981-87

As a iconic example of what this kind of intelligent adaptable to environment façades it should be mention one of the first examples of this architecture built.

In 1980 twenty two Arab countries agreed in the need of building a representative building for the Arab World as an Institution. The building will be a physical symbol of the friendship between France and the Arab world and, also, because of its special location, it will become the connection of old and new Paris.

In 1981, (more than 20 years before that Geli's Media-Tic Building), Jean Nouvel, together with Architecture-Studio, won design competition for building in Paris the Arab World Institute. Constructed between 1981 and 1987 the building has almost 17.000 m2 built next to the Seine river in the heart of the city of Paris. The team was given a complex program with a complex site.

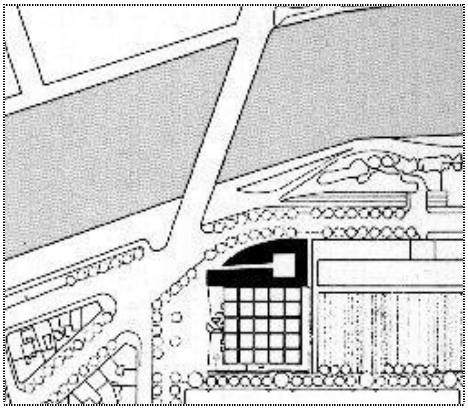


Figure 12A

Located on Rue des Fossés Saint Bernard the building acts as a buffer zone between the Jussieu Campus of Paris VI University and the Seine. The river facade follows the curve of the waterway, adapting itself to the view from the Sully Bridge. At the same time the building also appears to fold itself back towards Saint Germain des Prés neighbourhood.

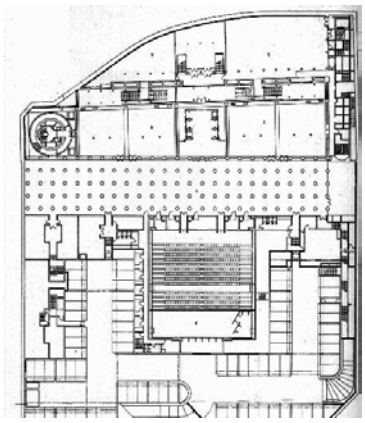


Figure 12B

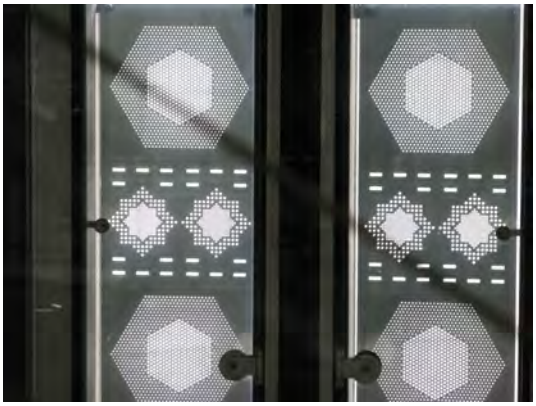
The front of the South Façade, more exposed to sunlight, is defined for consisting essentially of glazed panels of a homogeneous network of square leaves of glass. Every panel has been decorated by motives that remember to the ones used ones in the Arabic mosques. They are 250 metallic rosettes placed in the internal part of the crystal, that not are used as adornment, but also, are used as light filtering and parasols.

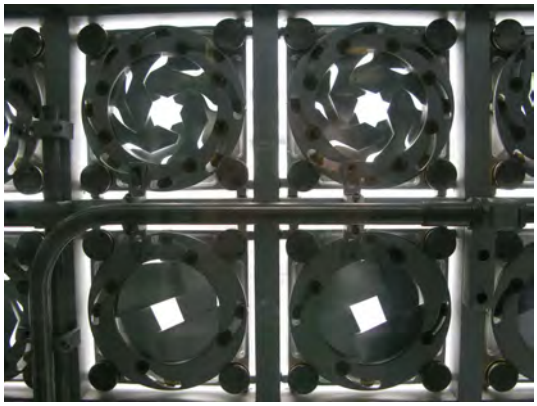

Pics, Ana Cocho, July 2011.

The openings of different size , combined, form every rosette, composed by small sheets of steel, some of which can be regulated by photoelectric cells. This makes possible that they work like diaphragms, being opened or being closed according to the needs of the diurnal light and creating surprising effects.


Pics, Ana Cocho, July 2011.

In the interior environments, the light, in this way leaked and modulated, does not invade, but wraps the spaces with an extreme smoothness and with an almost sacred silence perfectly adapted to the local functions: to read, to study, to visit.





Pics, Ana Cocho, July 2011.

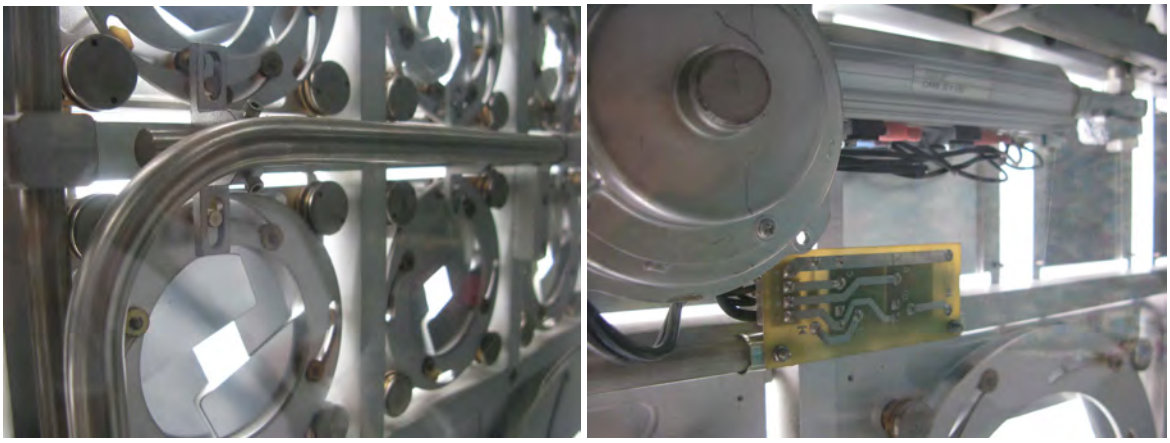
"It is, however, an ocular device of striking originality, made up of numerous and variously dimensioned metallic diaphragms set in pierced metal borders. These diaphragms operate like a camera lens to control the sun's penetration into the interior of the building. The changes to the irises are dramatically revealed internally while externally a subtle density pattern can be observed. Thus the whole effect is like a giant Islamic pierced screen, giving significance and an audacious brilliance to this remarkable building."

(see # 155, Sharp)

The unique use of high-tech photosensitive mechanical devices made this building famous in 1987. It uses high-tech photosensitive mechanical devices which control the light levels and transparency.

It interprets traditional wooden Arab latticework screens into a glass and steel construction with 27,000 light-sensitive diaphragms, organized in 113 panels, on 1600 elements, which operate like a lens of a camera. The changes to the irises are revealed internally far more than what can be observed from the exterior.

The motifs are actually 240 motor-controlled apertures, which open and close every hour. In this way, the building automatically controls its own light and creates a play of light and reflections on the inside.



Pics, Ana Cocho, July 2011.

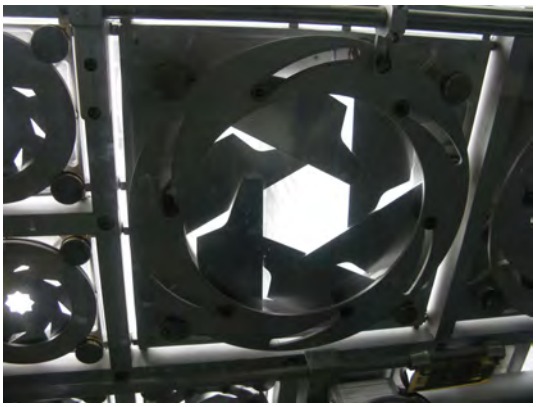
The lens' polygonal openings echo Arabian geometries. The huge 30 by 80 meter façade acts as a giant Mushrabiya. This subtle interpretation perfectly combined high - tech modernity with traditional Arabian architecture form. "Visitors should know that this building is Arabian," says Nouvel.

Figure 12C

Each diaphragm can be opened or closed by a computer, depending on the intensity of the sun's rays. Oriental ornaments, imitated with Western high-tech materials, create wonderful light reflections and shadows into the building's interior.

Mushrabiyas were the traditional lattice work that has been used for centuries in the Middle East to protect the occupants from the sun and provide privacy. During the various phases of the lens, a shifting geometric pattern is formed and showcased as both light and void. Squares, circles, and octagonal shapes are produced in a fluid motion as light is modulated in parallel. Interior spaces are dramatically modified, along with the exterior appearance.

While these ocular devices create an incredible aesthetic, they are functional from an environmental controls standpoint as well. Solar gain is easily mitigated by closing or reducing the aperture sizes.



Pics, Ana Cocho, July 2011.

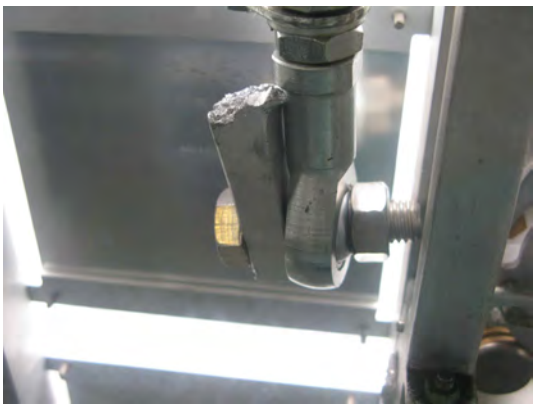
This 25,000 photoelectric cells similar to a camera lens, were designed to be controlled via a central computer to moderate light levels on the south facade originally with a stimulus-reaction behaviour.

As we can see, more that 20 years before of the media-tic building proposal, exactly the same behaviour was proposed for the Arab Institute Building. With one chip as we can see in the pictures per cell and with no global behaviour of the façade as adaptable to the environment pattern proposed.

This is another comparative that concludes that the technology proposed at the media-Tic is not correctly used, chosen or explained. Twenty years before Jean Nouvel already design this kind of façade.

On the other hand, due to over twenty years of daily use, only some of the Mashrabiyas still operate. The physical parts of some cells have collapsed, as we can see in the pictures below, showing that mechanical behaviour was the weakest part of the whole design after some time of intensive use.

Since quarrels between the sponsors consisting of the French Department of Foreign Affairs and 22 Arabian member states have brought financing and support to a standstill. The network control, that used to work as precisely as a clockwork with its two motors per square, is now out of order. The façade no longer works.



Pics, Ana Cocho, July 2011.

The cells that actually survive are now controlled by a central computer in an attempt on updating the system of the façade and it stimulus-reaction original behaviour.

2.2.C. What is EmDeplo?

0. Emergency Shelters & light structures

Buckminster Fuller

As a prologue to the presentation of EmDeplo system I would like to write some words about the author that is considered the father of emergency sheltering and light structures focuses on low cost production. I would like as well to mention some of the examples more remarkable on the research done, obviously, apart from the typical emergency tents used nowadays.

Buckminster Fuller was considered an outsider from architecture and also an eclectic inventor. He was basically obsessed with two basic principles.

- affordable housing
- light weight structures

Fuller considered that the building industry in his day was working with outdated processes. So, for the world fair in 1933 he proposed the *Dymaxion Dweller Machine*. *The housing example will have:*

-automatic ventilation
-air-filtering
-movable partitions
-plumbed in vacuum cleaner
-2 Dymaxion bathrooms

Unfortunately the budget was 100 million dolars. Fuller, nevertheless, stimated that after mass production the cost per house can be reduced to 1500 dollars.

The only house that was finally constructed was the Wichita house, in 1948, based in his 1928 proposal. With half of the price of normal housing in the market if was supposed to be able to be built by six men in one day or in six days by one man.

Finally the Dwelling Machine did not go to market because Fuller was obsessed on the supervision of all models, thing that was completely impossible in a mass production process.

But Fuller's work that was more influential for this thesis was his project for a military shelter in 1955. US Navy was ne of Fuller's big clients along the years. Able to protect , in his own words, the soldiers even in the case of an atomic bomb, the shelter easy really light, easy to assemble and able to be lifted by a helicopter. In fact, it was supposed to be so cheap that it even was possible to leave it behind at the battle field if necessary because its really reduced cost.

1954, B. Fuller Marine dome
(see # 150, Kronenburg, 1995) p.43

Military engineering has been always the principal vehicle for emergency sheltering innovations. This shelter was supposed to substitute the military tents used so far. Although, military tents had been improved along WWI and WWII its configuration and way of working was basically the same. And it is , in fact, the basic idea behind the tents being used currently in emergency hospitals.

Military tents for the World War I
(see # 150, Kronenburg, 1995) Chapter 5.

During 70's and 80's the World War experiences were used to try to develop some improvements for the shelters in the case of disaster. MUST Hospital was, for example, a self-contained medical unit, transportable and supported with double wall inflatable containers

MUST Hospital (see # 150, Kronenburg, 1995)

The first emergency shelter effort was done for the earthquake of 1960 in Concepcion, Chile. The first problems with these kind of shelter appeared:

- the problem of having to do the erection with non local personnel.

- the problem of flown in components.

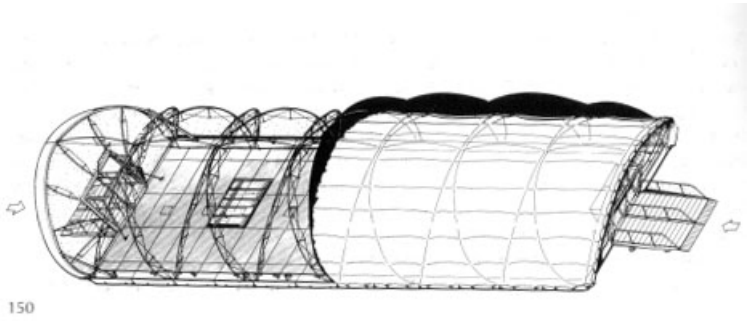
In this sense Future System proposed in 1993 the Hospitality Tent Museum of moving. In his discourse abut disaster they focus on the idea of a general misunderstanding:

What people after a disaster do need is not really what we think they need.

They differentiate for example, different options for different scenarios, dividing the situations basically in three types:

1-natural disasters
2-war, post war
3-refugees

148,149 Future Systems, Peanut House, 1984. The hydraulic arm can be manoeuvred from the surface of the lake to an elevated position and back to the land; **150** Future Systems, hospitality tent for the Museum of Moving Image, 1993



150

1993, hospitality tent Museum of moving: PTFE+inclined GRD wire, stiffed ribss+steel arches.
(see # 150, Kronenburg, 1995)

This was the first example of emergency sheltering that included PTFE in its design.

Artist+activist+designer.

On the other hand, the idea of the multidisciplinary designer was born in the 60's creating the relationship designer/activist/artist, what provoked the birth of the first examples not coming directly from military research.

Archigram, archi(tecture) gram, for example, were accused of doing art, no design processes, and so that, accused of no solving problems.

The Cushicle, presented by Michael Webb in 1966 was the first proposal of a human customizable environment. Able to take several forms it has a food storage and water supply. Air sustained it can be inflated creating a space for a human including a chair.

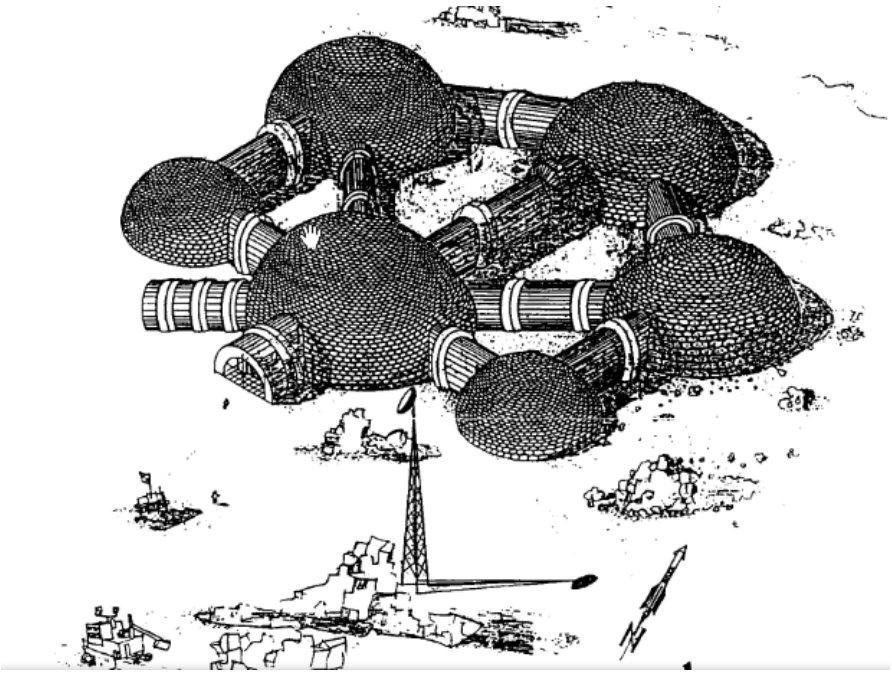
Also in those days, 1969,Cedric Prize, was proposing the Hair Tent where he defended that there was an inherent uncertainty in the validity of any permanent site. This basic design premise, thus, for Prize, activates the entire nature of the component parts.

Lavapolis

In 1991 Texas A&M university presented : Lavapolis, M.A.R.S. (Modular assembly reusable structure).

Texas university was collaborating with NASA since 1986 and they proposed through a study of the Department of Architecture students under the supervision of Professor Ikhlas Sabouni this Mars Habitat for future crews, of around 20 people, that will need to stay at the planet between 6 and 12 months.

Inflatable cylinders supported with light weight aluminium rings Lavapolis was considered to be the largest pneumatic structure until that time.



(see # 159, Sabouni, 1991)

1. EmDeplo Concept

The idea of an "Emergency Parametrically Customized Deployable System" was born with [An_D].

[An_D] was a work team, Design Research Laboratory Based, at the Architectural Association of London, which investigated about deployable parametrically customized membranes.

This chapter is a short brief of the work I did during that period which is my starting point for the next thesis chapters.

Since the beginning we were interested in body oriented architecture, a deployable fully integrated into the body architecture that deploys anywhere and anytime.

W.H.O., remarks natural disasters and other unpredictable events are so common today that urges to find out response for them. Architects are asked to invent new kind of high adaptable and rapidly deployed spaces for different emergencies.

The Emergency Intermediate Health Deployable System, Interface customized, will be able to satisfy most medical needs in the shortest time in any scenario. Factory connected and parametrically designed. A time-based system in two kind of unit:

- *Basic triage:*

Fast deployable pack ready to be sent immediately after the disaster. With a limited use in time and focus in the acute phase. Usable as an adaptable triage or first-aid unit working alone, or, with an existing health facility that is damaged or overcrowded.

- *Specific health Unit System:*

Different rapid deployable units customizable according to the kind of emergency through an interface-based design factory connected. Units response to the specific spaces and needs, so, it is a complete integrated system, able to adapt to specific diseases, spatial and technological needs, and able to form a field hospital.

Deployable 3D structure from a flat surface, able to arrive directly from the factory to site, is perfectly packed and ready for an easy and quick enablement. A Multilayered Membrane Intelligent System different for both packs but based in the same logic.

It will be designed through a 2D patterned deployable surface that expands into a complete 3D space: The idea of having a multilayered patterned membrane that expands 400%.

High adaptability and rapid deployability are highly requested in order to fulfill every kind of timely measures. Sometimes the answers that are able to achieved are not satisfactory in term of effectiveness and efficacy. Too often the provisory dwelling units used to resolve an immediate housing problem after an earthquake disaster tends to became permanent and rather frequently the first aids arrive too late on the place of the calamity.

It has always been difficult to predict the unpredictable but with nowadays knowledge and technologies it should be at least possible to forecast the predictable in order to be ready for providing the primary human requirements. Out-of-hospital emergency medical services (OHEMS) and hospital emergency departments form the two pillars of emergency medical services. Although better coordination could improve performance and reduce mortality and morbidity, it is not achieved in most cases. Out-of-hospital emergency services must cope with the fast development of communication systems and technology, and the population's demands for fast and efficient services.

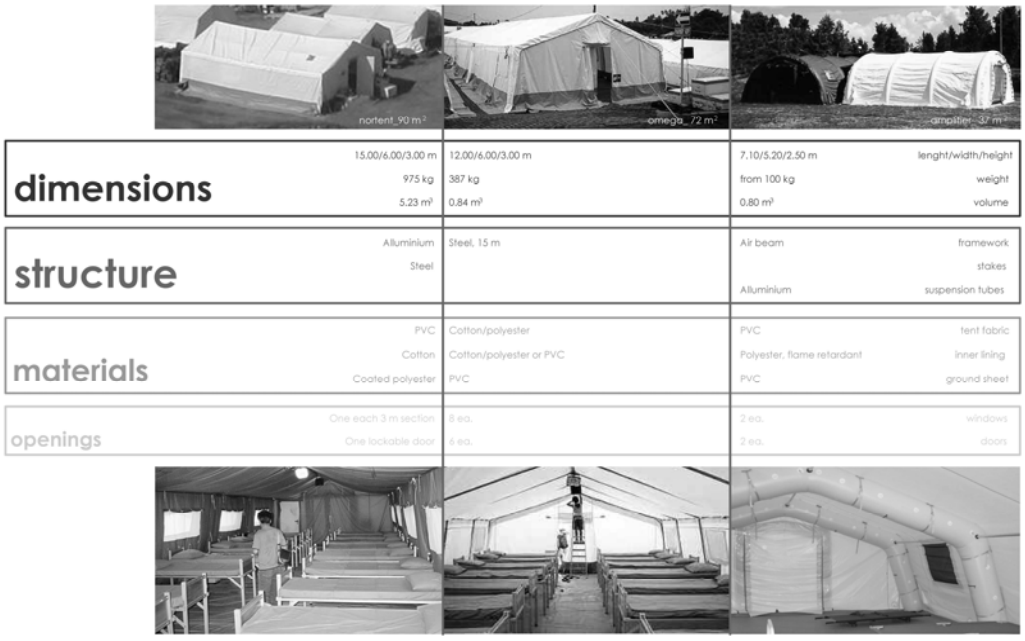


Figure 13

In an attempt to fulfill all these necessities health care delivery systems are now confronting important challenges posed by the rapid development of communication and biomedical technology, the need for cost-containment and the focus on effectiveness and efficiency, the increase of populations' demands for services and changes in demographic and epidemiological factors.

Even though all this growing interest about the increasing OHMS performances the WHO is still facing a problem with intermediate health services. At this stage they subdivide the emergency units in three categories, ascribing different aims at diverse duration in time.

- First 48 hours:

It is allowed to work with the energy of the location. In any case is recommended to be Self-Sufficient. It is needed to have operative capacity in site.

- From 3 days up to 15 days:
  Need of Secondary attention, traumatized continuous control.

- From 15 days up to 2 years:
  Need of not being tents or non self-sufficient structure mechanisms, having Closed air circuit, self energy and primary attention.

[An_D] members

Ana Cocho Bermejo Andrea Balducci Caste' Daode Li

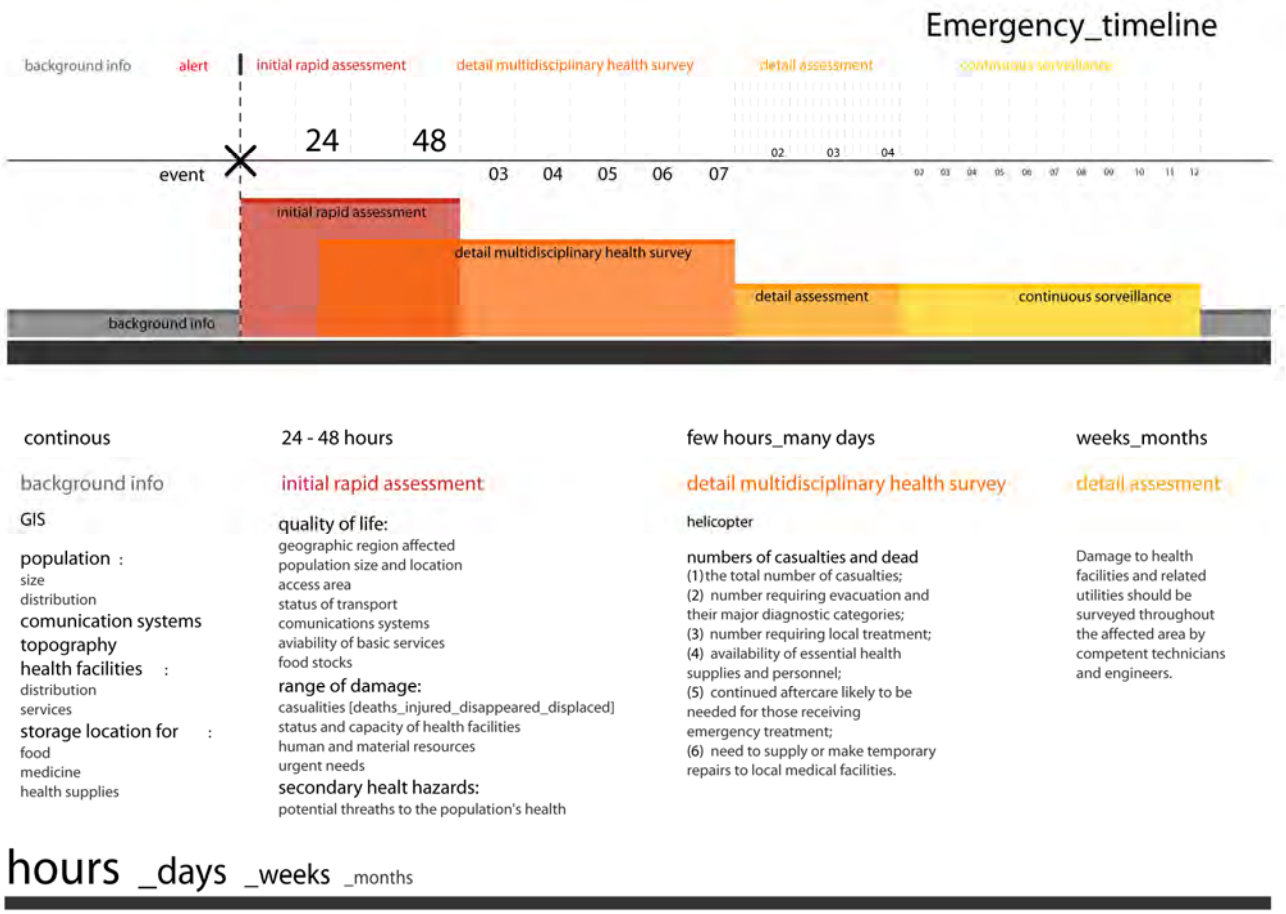
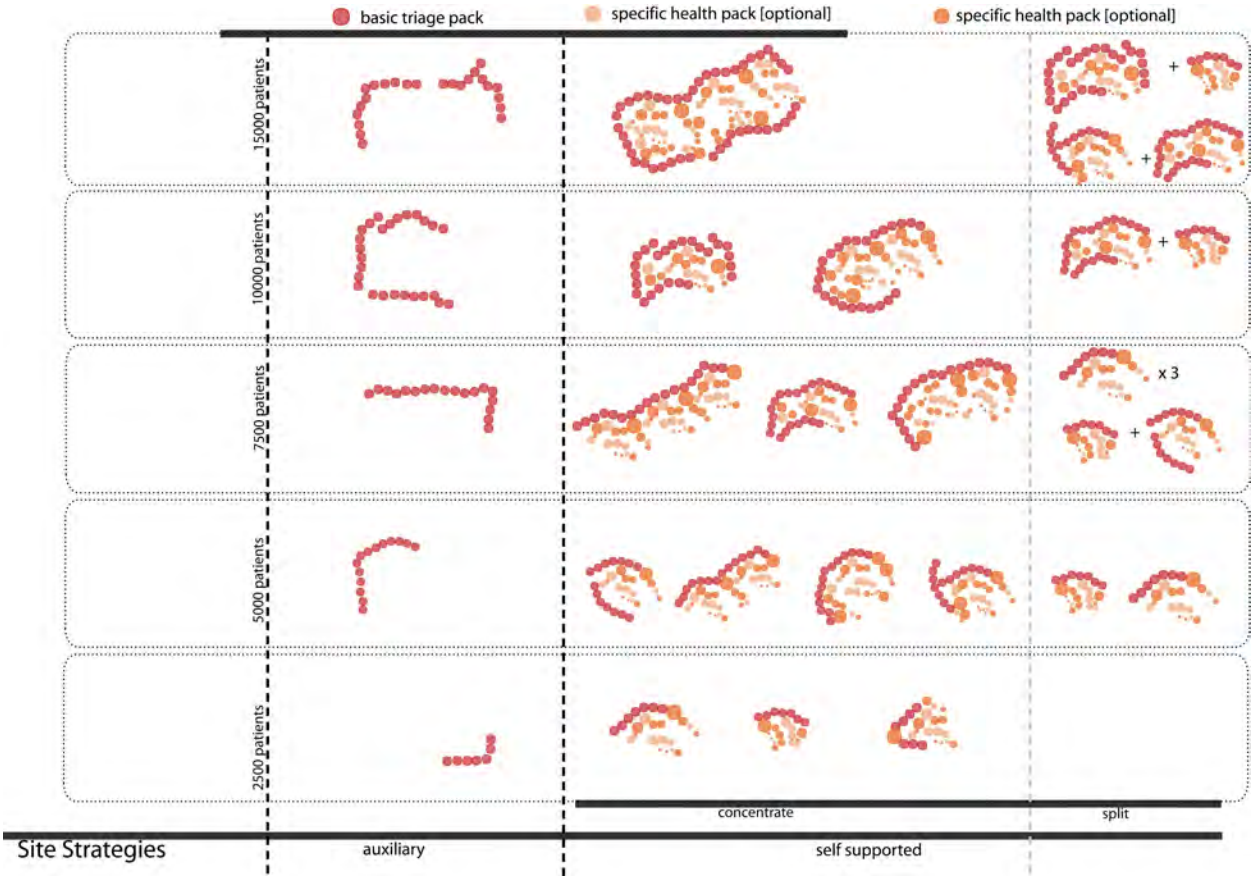


Figure 12

2. Proposal Basis



2. A The factory interface

In recent decades the notion of time-based design has increased the architectural practice's interest to explore new kinds of design processes more linked to biology, philosophy and other disciplines which main potential consider, a real application into contemporary process, to change the conventional architectural methods by diagramming, mapping and animation techniques. Since we design our interface, which, connected to the overall system generates the health system, we had realized that not only the common architectonical design techniques will be used.

The EMERGENCY INTERFACE will help us to decide quickly in critic situations how the Unit must be and will carry us through the design in real time. The Health Deployable System has an interface-based design. When an emergency, predictable or unpredictable occurs WHO will work through the Interface. Connected to the factory , the complete fabrication process of the membrane system with be perfectly customized for the emergency scenario. Making the client answer a serial of questions, the interface connected to the factory will fabricate the system,

customized for the situation, pack and folded, ready for deployability by truck, plain or boat trough several containers.

The interface will design some compulsory parts for the case, already customized for the situation, but also will offer to the client some non compulsory parts that the client will be able to choose to have at the unit system. Also customization of several compulsory and non compulsory parts is allowed. The questions the interface proposed are those which help the customization of the system and deployability. Sample questions will be where the emergency happens or it is going to happen, kind of emergency, expected number of days, expected number of patients, kind of diseases, etc…(See diagram below)

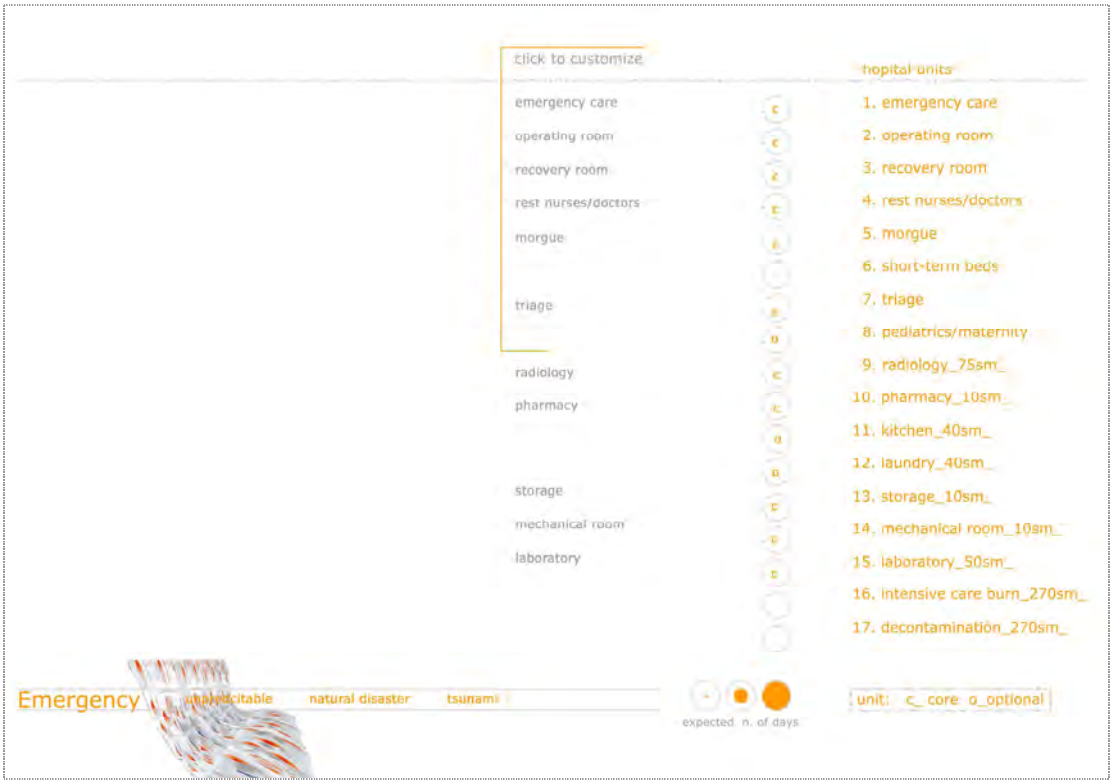
So, in the Emergency Health Deployable System, the parameters are the characteristics of the scenario and the kind of disaster. The parametric customization of the membrane system through the Interface factory connected is the basic idea of the efficiency of the Emergency Units.

1

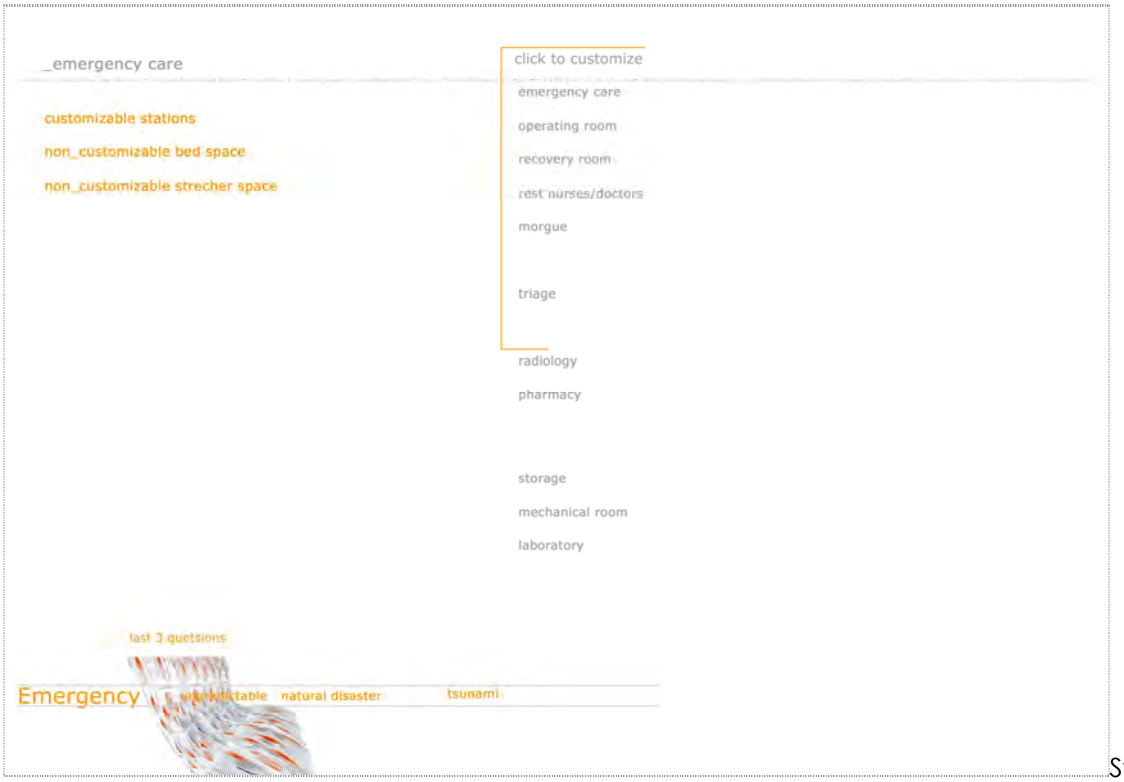
2

Once the interface work has finished, all the data will be receive at the factory. Then the proper pattern system will be automatically designed, creating the proper membrane sheet, and folded and packied, for a proper deployment.

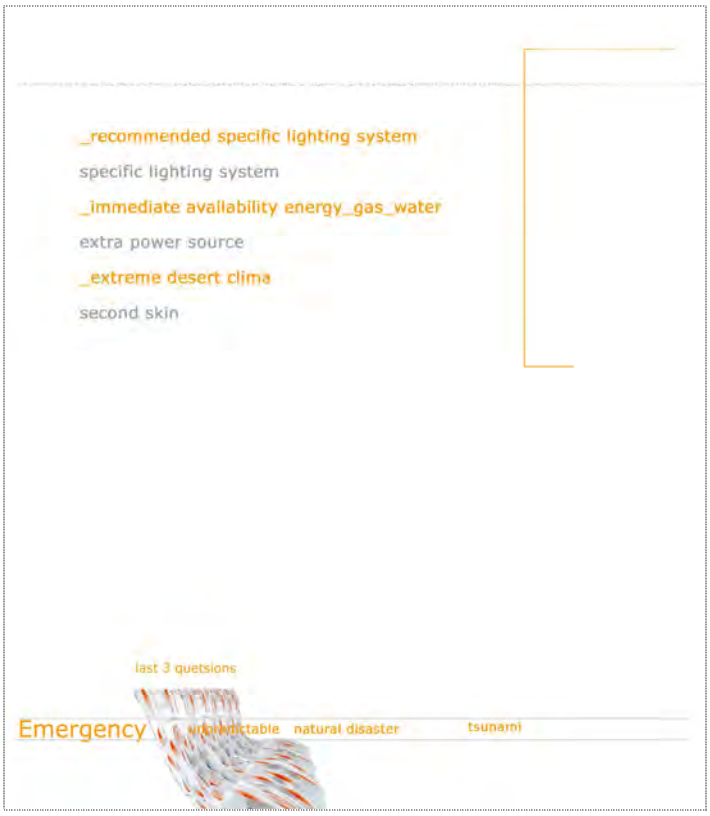
The clear properties of the material and some basic real controlled parameters will perform the unit, helping us to create a real transformable, transportable and customizable space. Developing the pattern in some different scales, it will arrive to its final development.


click to customize
emergency care
operating room
recovery room
rest nurses/doctors
morgue
triage
radiology
pharmacy
storage
mechanical room
laboratory
hopital units
1. emergency care
2. operating room
3. recovery room
4. rest nurses/doctors
5. morgue
6. short-term beds
7. triage
8. pediatrics/maternity
9. radiology_75sm_
10. pharmacy_10sm_
11. kitchen_40sm_
12. laundry_40sm_
13. storage_10sm_
14. mechanical room_10sm_
15. laboratory_50sm_
16. intensive care burn_270sm_
17. decontamination_270sm_
Emergency
unpredictable
natural disaster
tsunami
expected n. of days
unit: c_ core o_optional

Step 3


_emergency care
customizable stations
non_customizable bed space
non_customizable strecher space
click to customize
emergency care
operating room
recovery room
rest nurses/doctors
morgue
triage
radiology
pharmacy
storage
mechanical room
laboratory
last 3 quetsions
Emergency
unpredictable
natural disaster
tsunami

Step 4

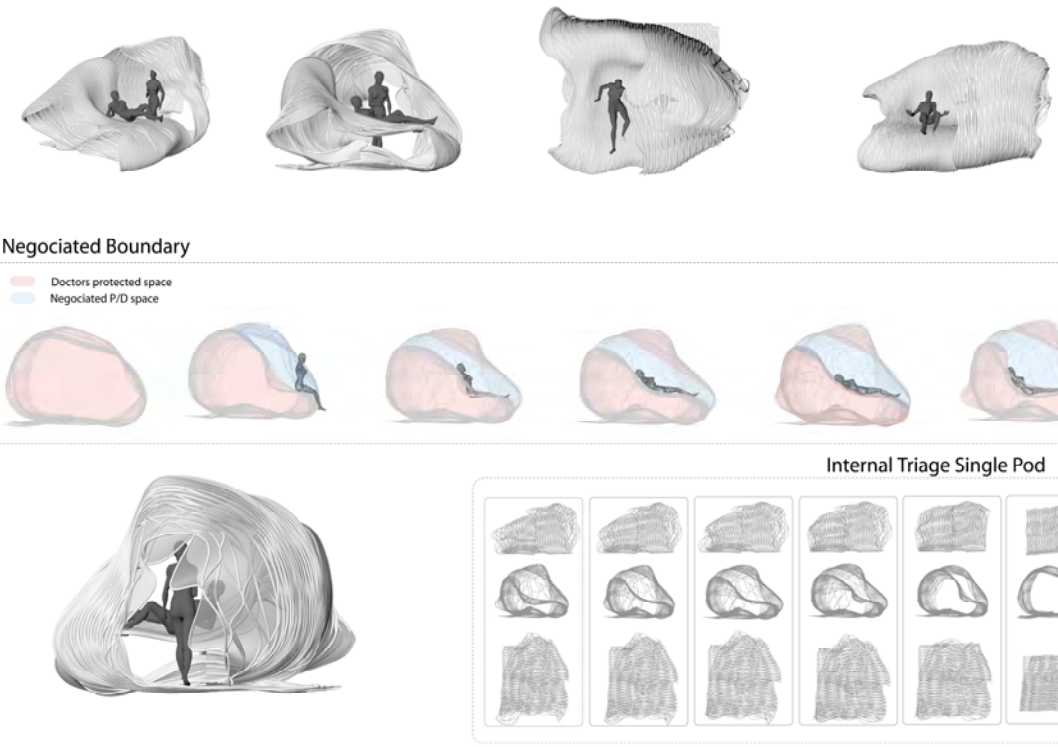


Step 5

2. B Basic Triage Pack

The triage proposed consist in a negotiated mixed space where the interaction between doctors, nurses and patients takes place only in the moments that is completely compulsory. The "sensor-ized" pod will react to the patient weight and movement making a complete adaptable cellule, but being always within the positions allowed for the doctor to the patient. Maximum number of patients will be attended with a minimum numbers of doctors and nurses. This new idea of triage eliminates, because is inherent in it, the need of another space for the admissions room.

Figure14



Four different types of basic triage packs are available since the first minute. They are predesigned and packed in order to solve 4 variations of most common different emergency situations.

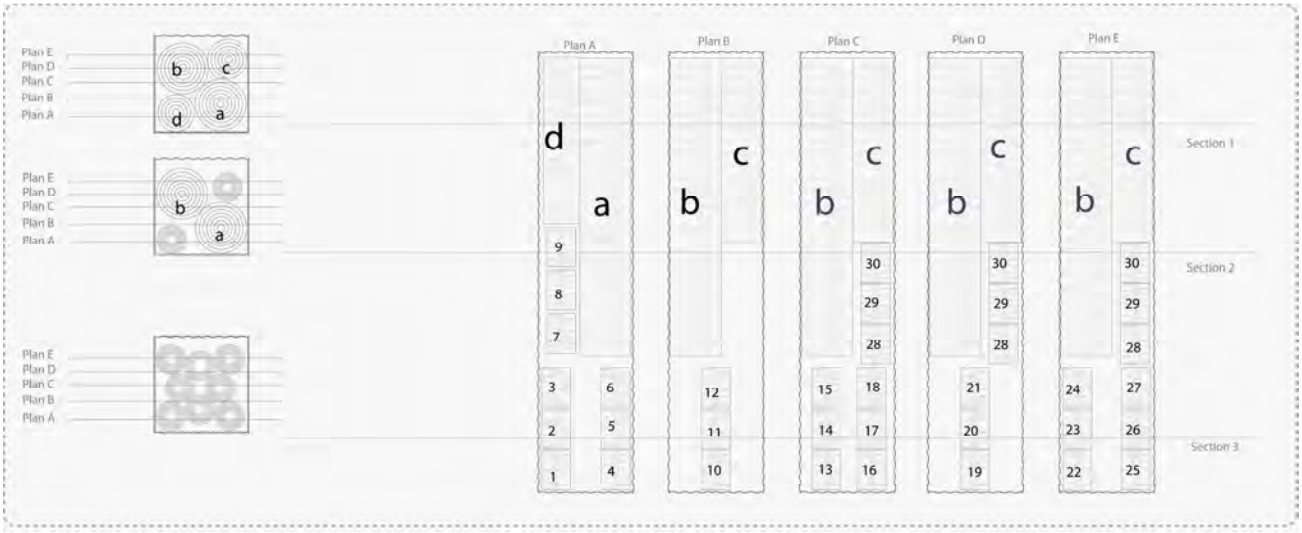
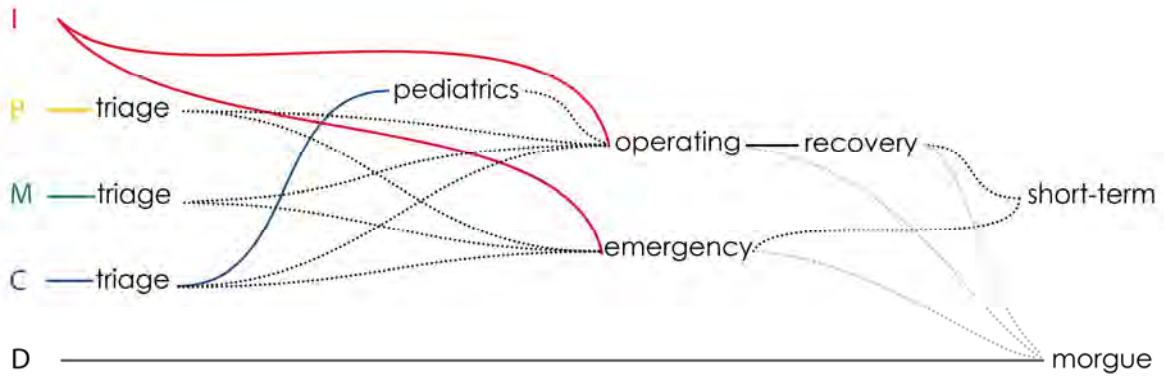


Figure 15



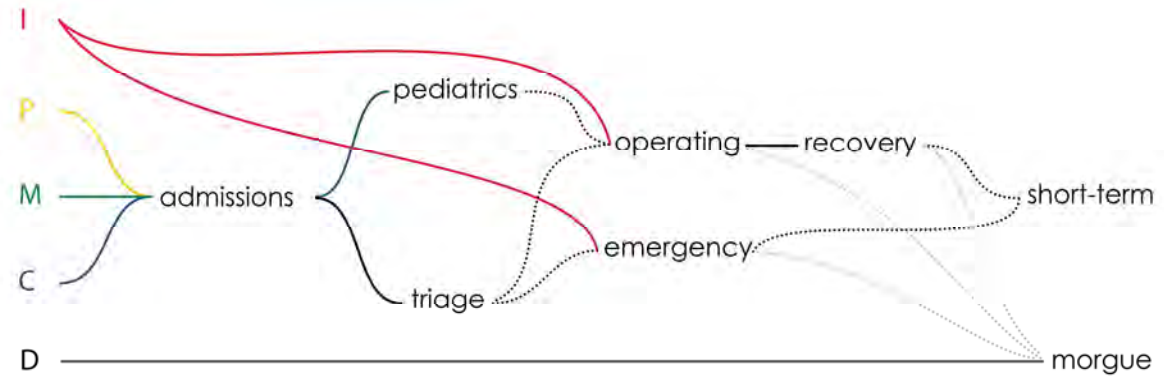
Immediate — They require immediate surgery or other life-saving intervention, first priority for surgical teams or other immediate treatment.

Priority — Their condition is stable for the moment but requires watching by trained persons and frequent re-triage. will need hospital care

Minor — These will not need advanced medical care for at least several hours. Continue to re-triage in case their condition worsens.

Child — They will be transported in the pediatrics ward in order to receive medical care

Deceased — These who are beyond help.



2.C Specific Units Pack

After the first 48 hours, with the background data already analyzed for the interface, it is able to be decided and sent a set of customize specific units.

In this way we obtain from the interface a customized program of needs. A program that is going to solve the real scenario problems and cases. The diagram above shows the customized needs programs for the case of study shown in the next pages: an earthquake scenario.

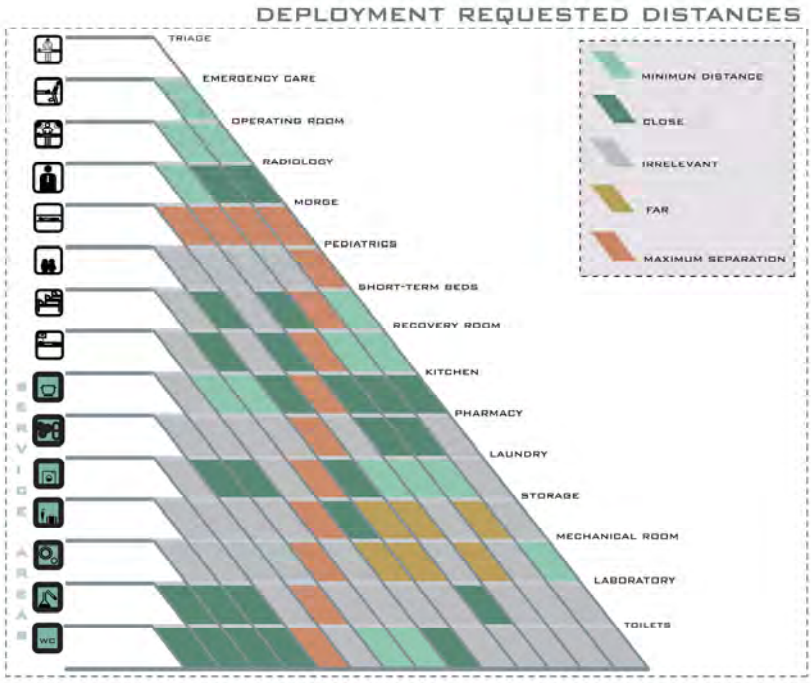


Figure 14

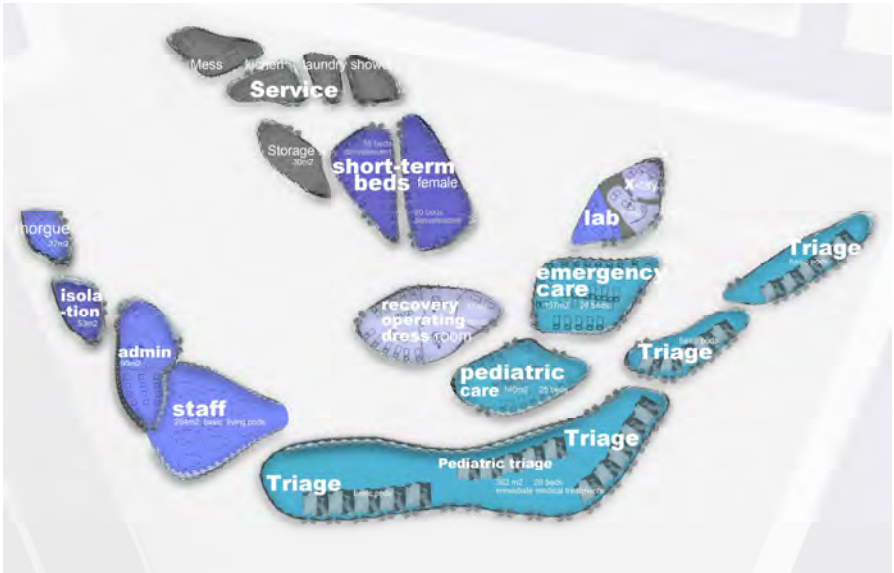
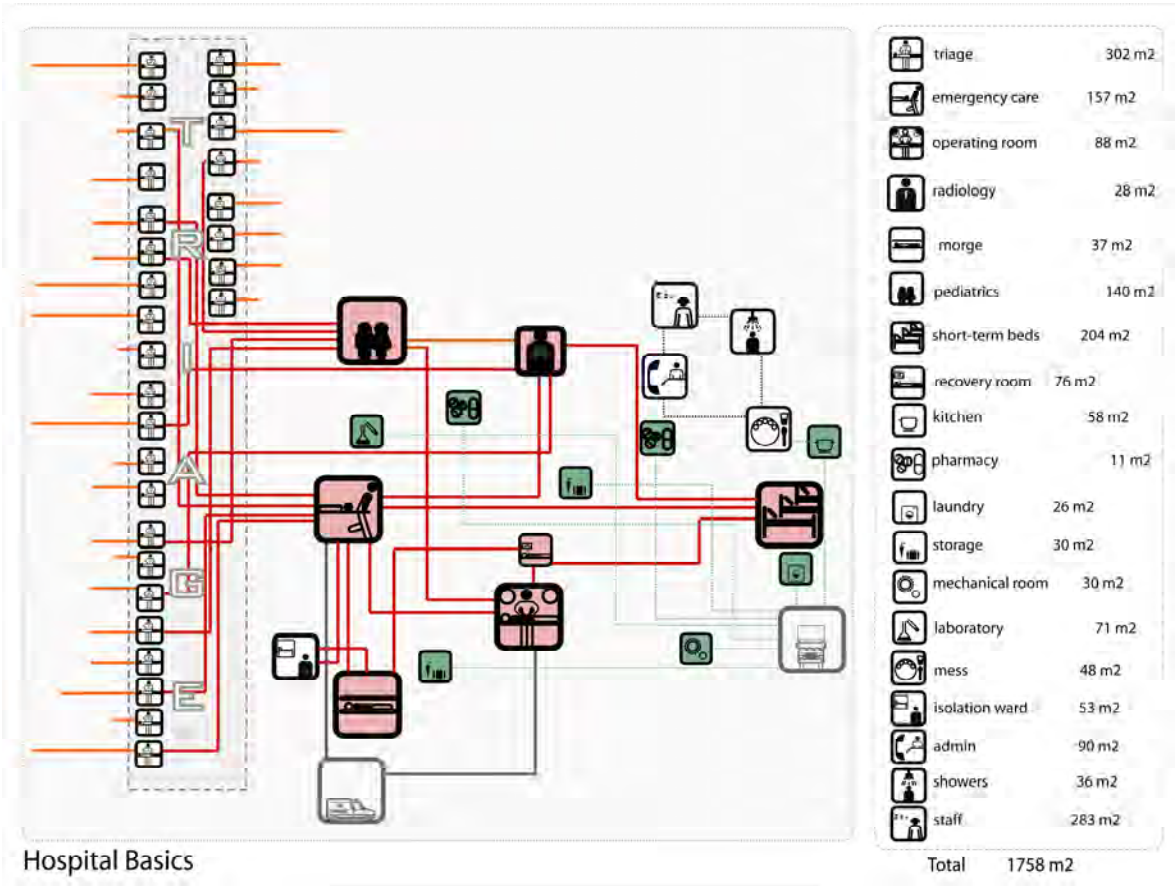
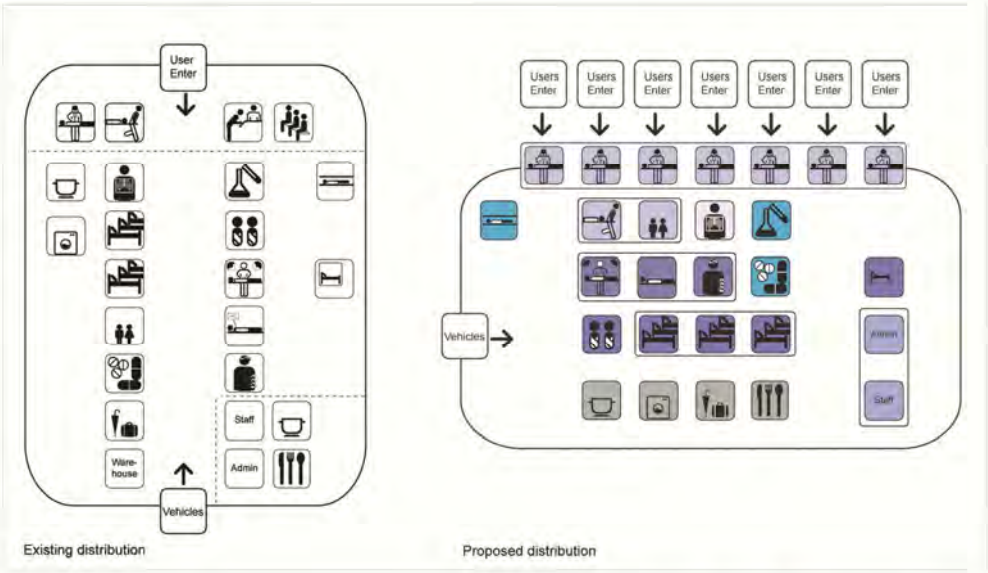
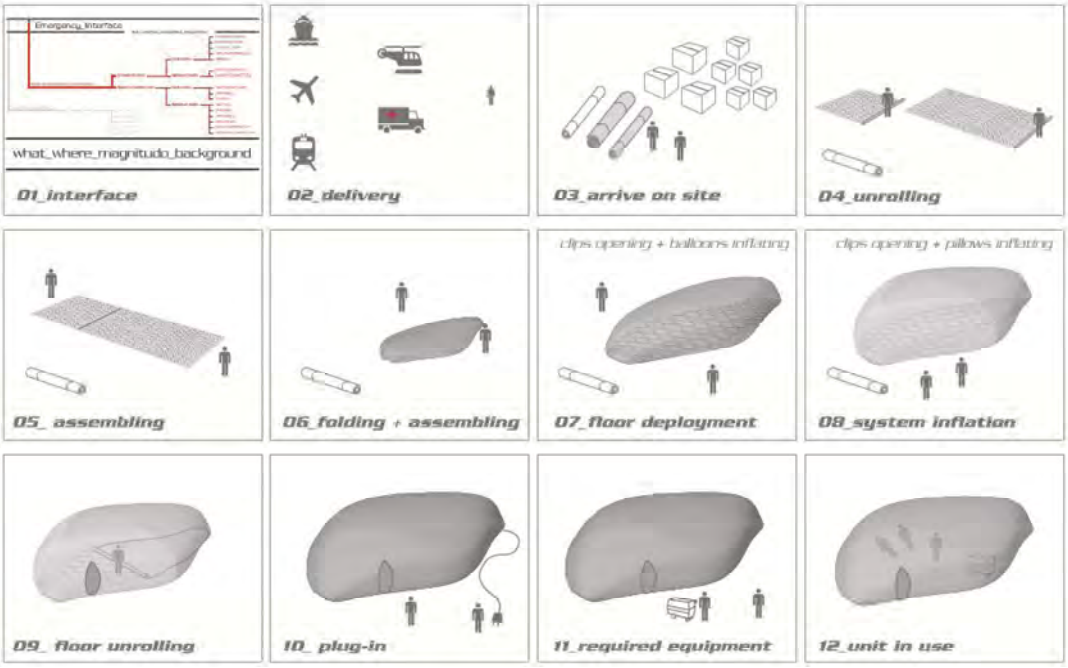


Figure 16


triage 302 m2
emergency care 157 m2
operating room 88 m2
radiology 28 m2
morge 37 m2
pediatrics 140 m2
short-term beds 204 m2
recovery room 76 m2
kitchen 58 m2
pharmacy 11 m2
laundry 26 m2
storage 30 m2
mechanical room 30 m2
laboratory 71 m2
mess 48 m2
isolation ward 53 m2
admin 90 m2
showers 36 m2
staff 283 m2
Total 1758 m2
Hospital Basics

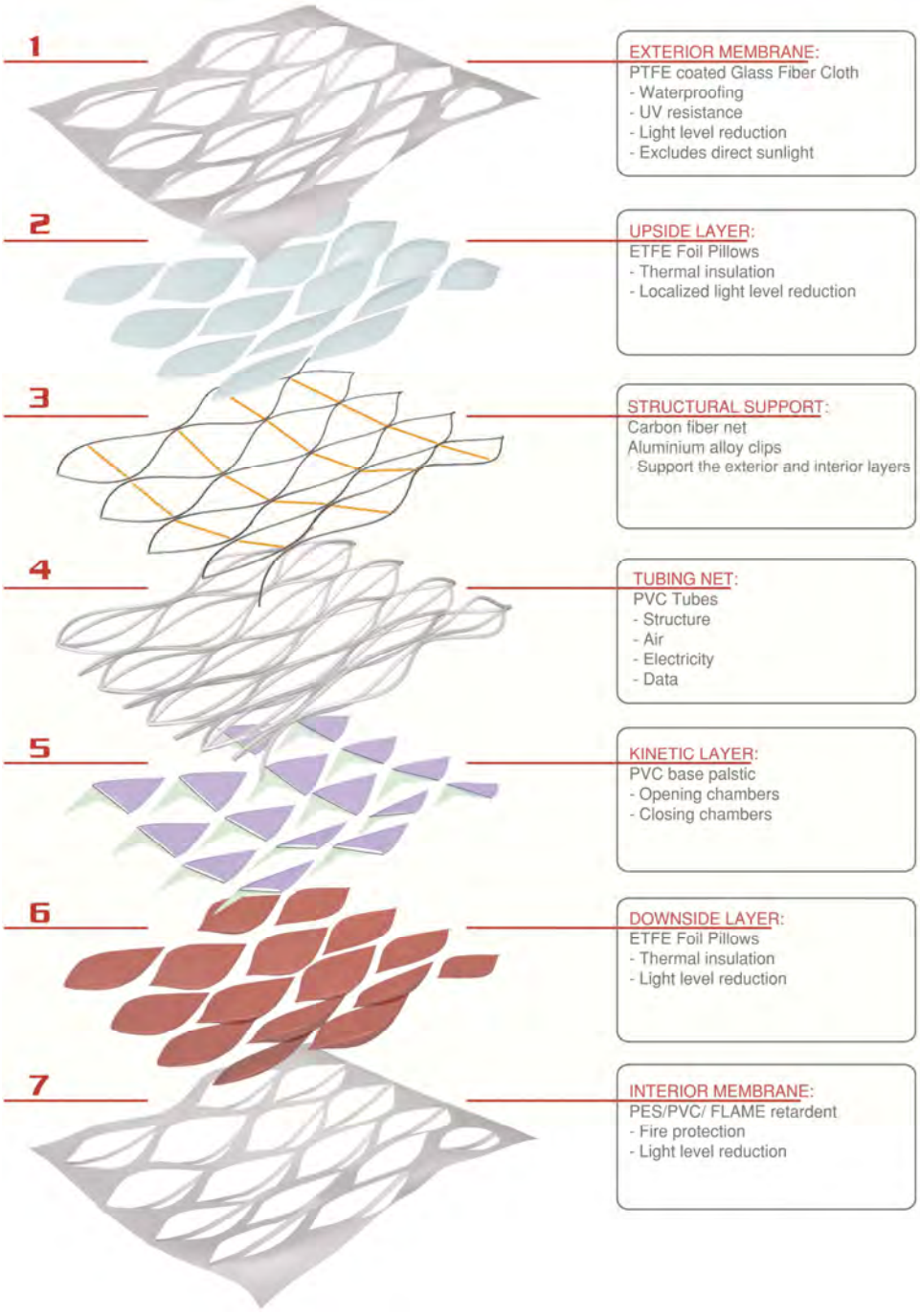

User Enter
Staff
Admin
Ware-house
Vehicles
Existing distribution
Users Enter
Vehicles
Proposed distribution


P/D Patient/Doctor boundary
P+D Patient+Doctor space
P Patient ward
D Doctor area
S Service area
Emergency care
Operating
Pediatric
Pharmacy
Laundry
Pediatric care
Recovery
Short-term beds
Laboratory
Storage
Triage
X-ray
Isolation
Morgue
Kitchen
Dress
Mess

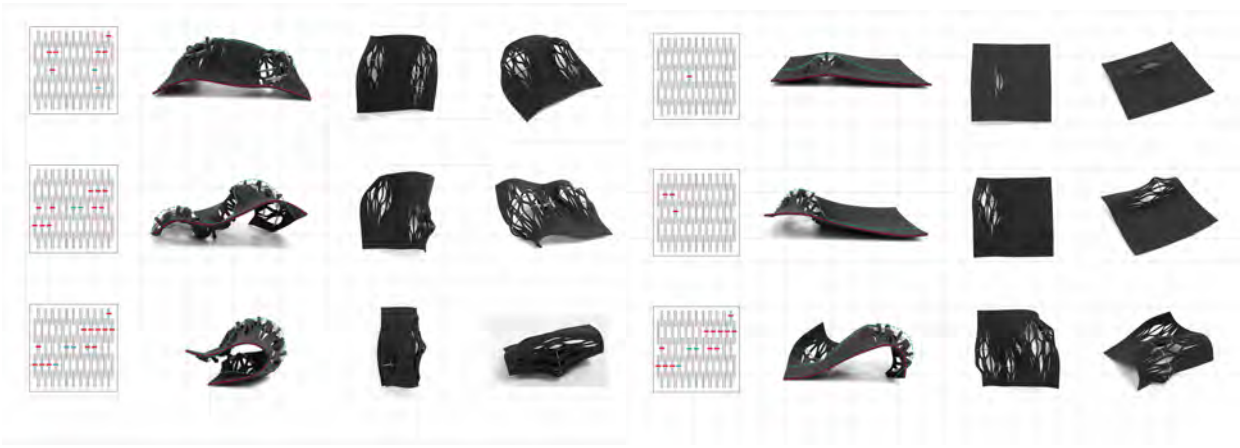

what_where_magnitude_background
01_interface
02_delivery
03_arrive on site
04_unrolling
05_ assembling
06_folding + assembling
clips opening + balloons inflating
07_floor deployment
clips opening + pillows inflating
08_system inflation
09_ floor unrolling
10_ plug-in
11_required equipment
12_unit in use

3. EmDeplo Material System

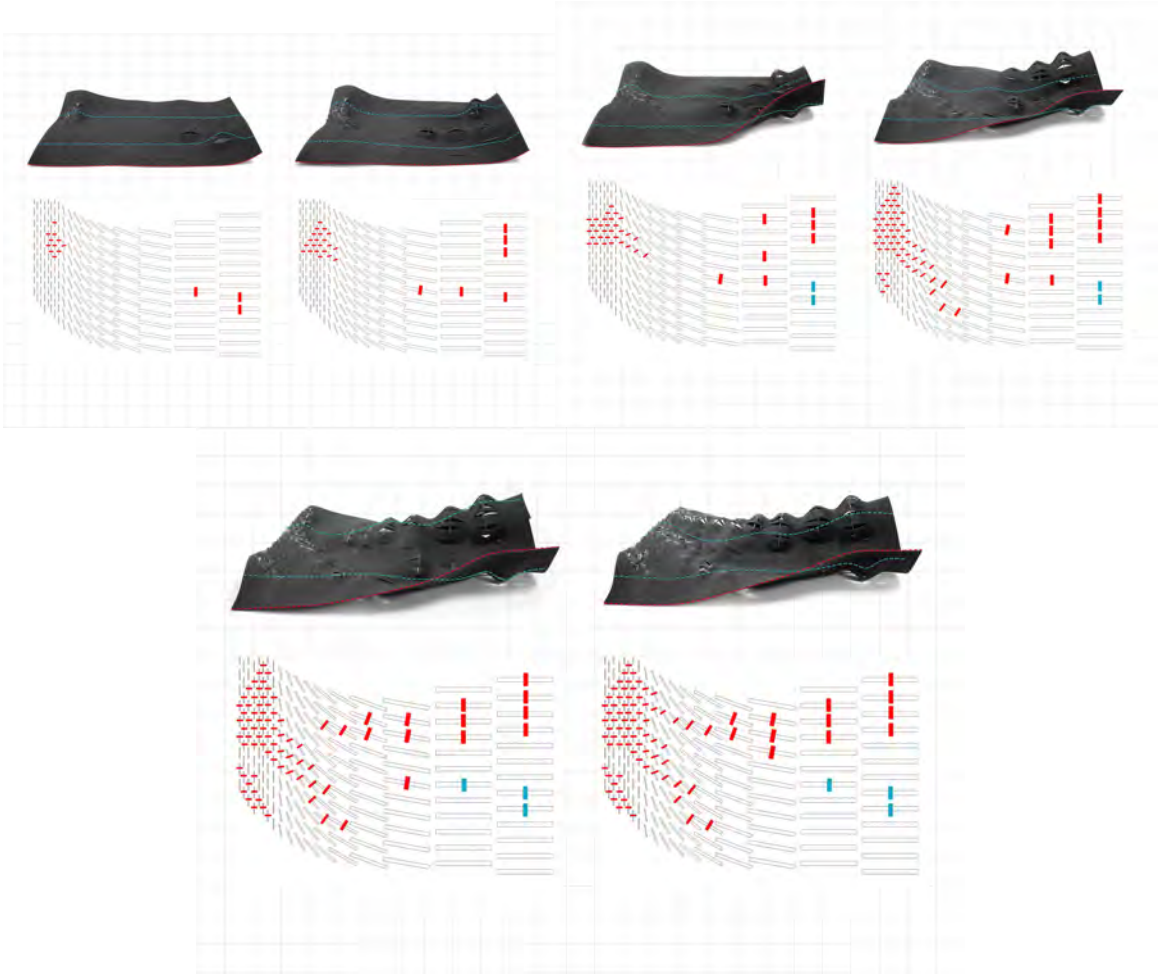
It is composed by:



The patterns, controlled in four different scales will hold all the design weight. Controlling some basic parameters, it is possible to order the material how to behave. Controlling some basic parameters is possible to tell the hospital what to do. This is the future adaptable customizable architecture.

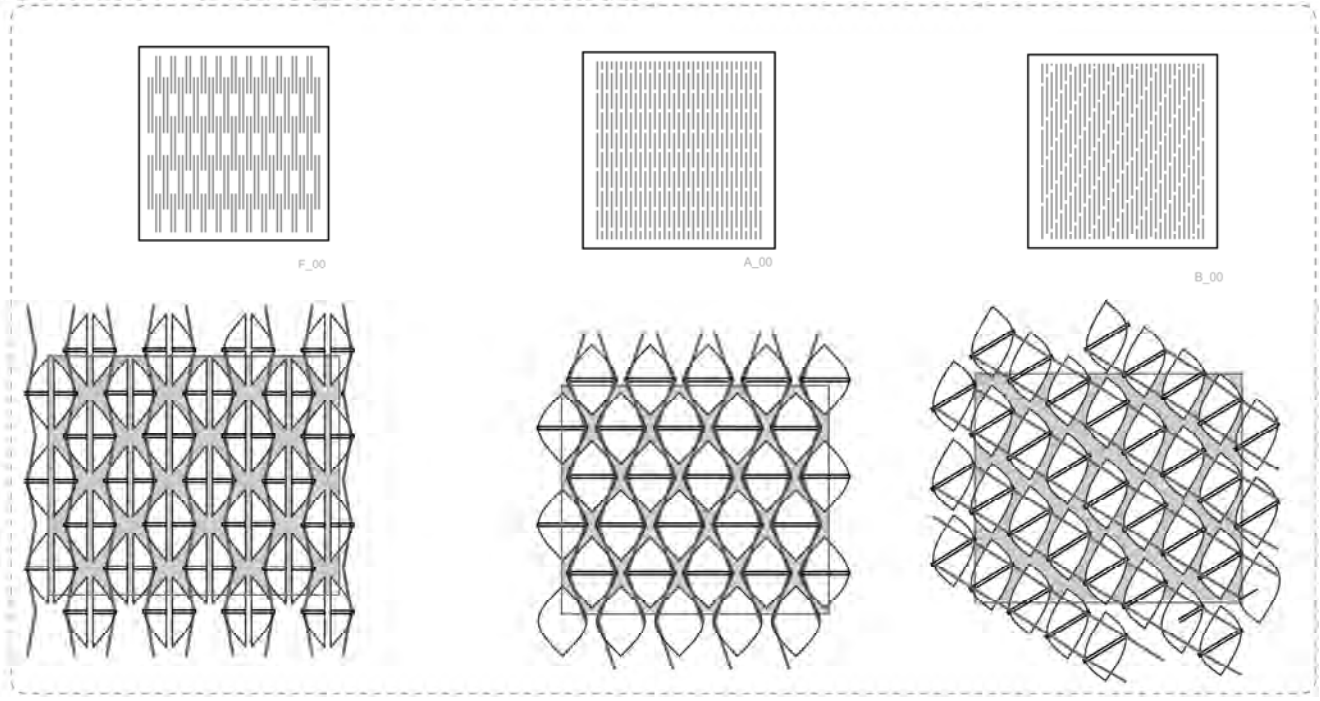


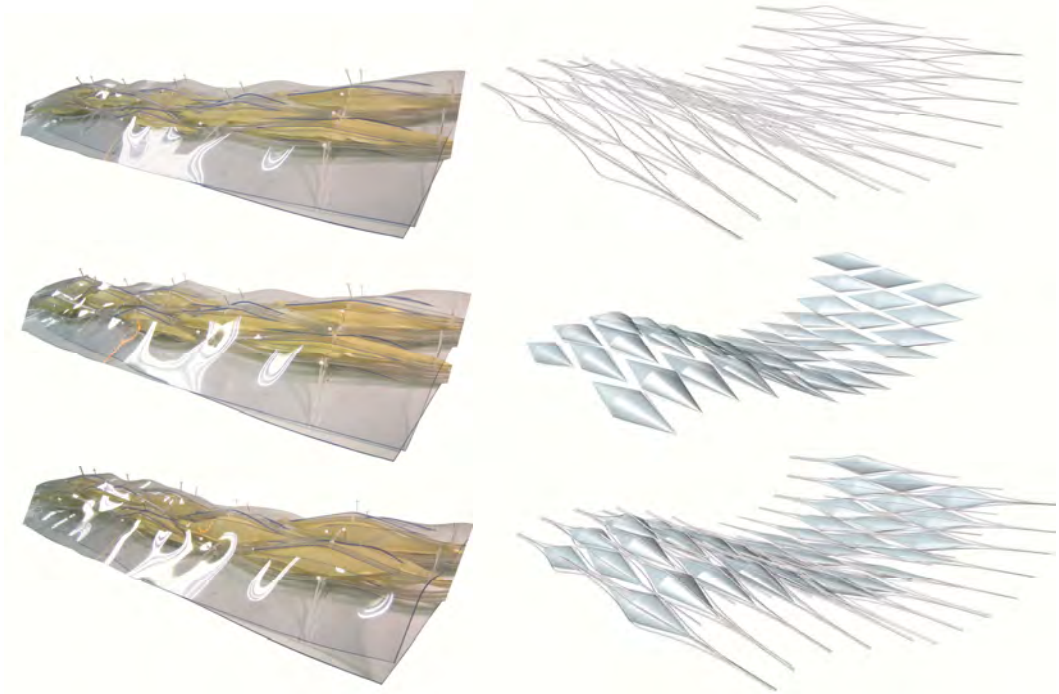
So that the system is not only an envelope for a space, it is not a space that after you can fix with the medical equipment. It is designed instead for each focus situation, patient and use.

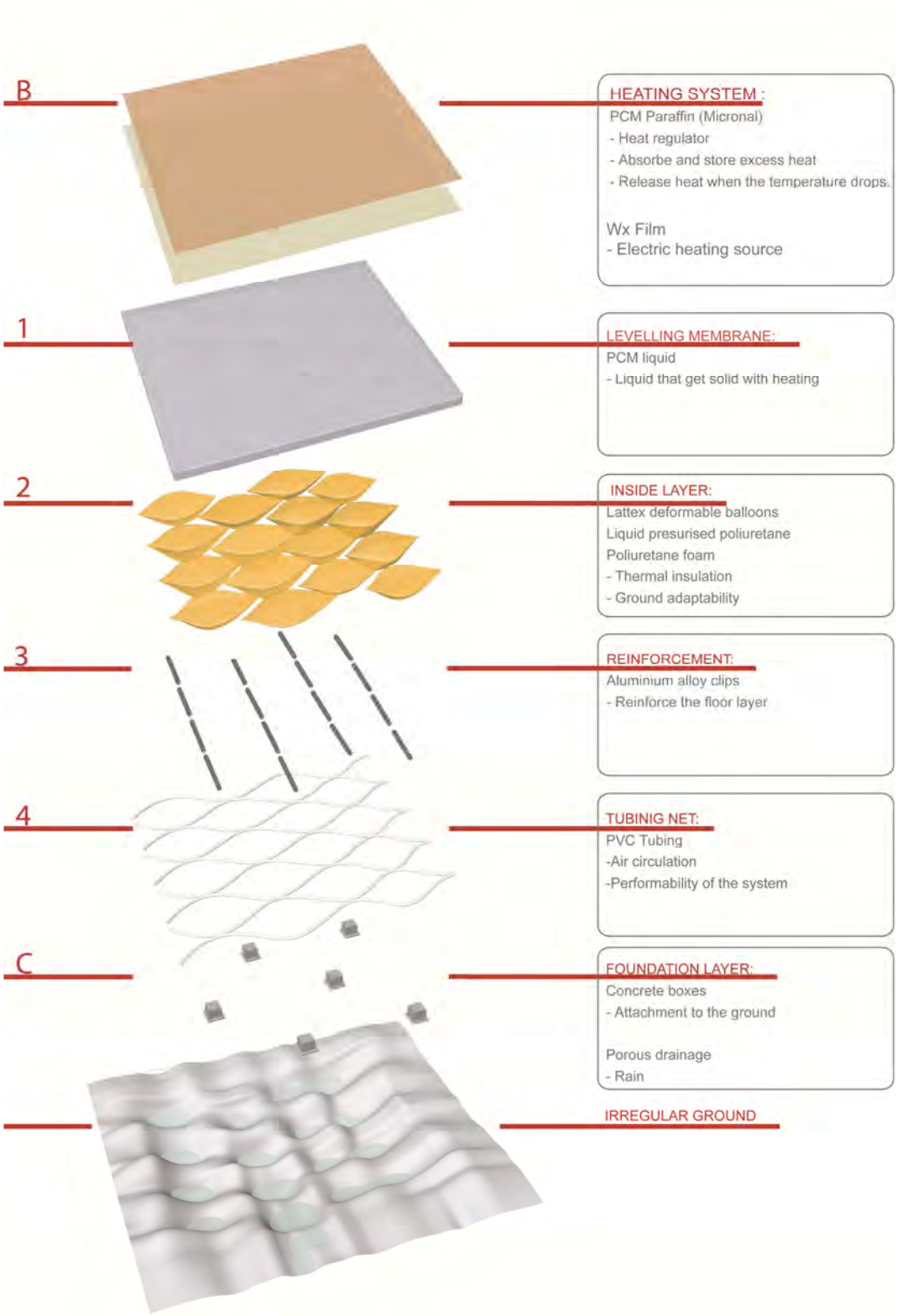


It is a fully integrated system. It will be design including water, electricity and oxygen supplies as well as the necessary medical equipment.

INITIAL PROPOSALS FOR THE SYSTEM

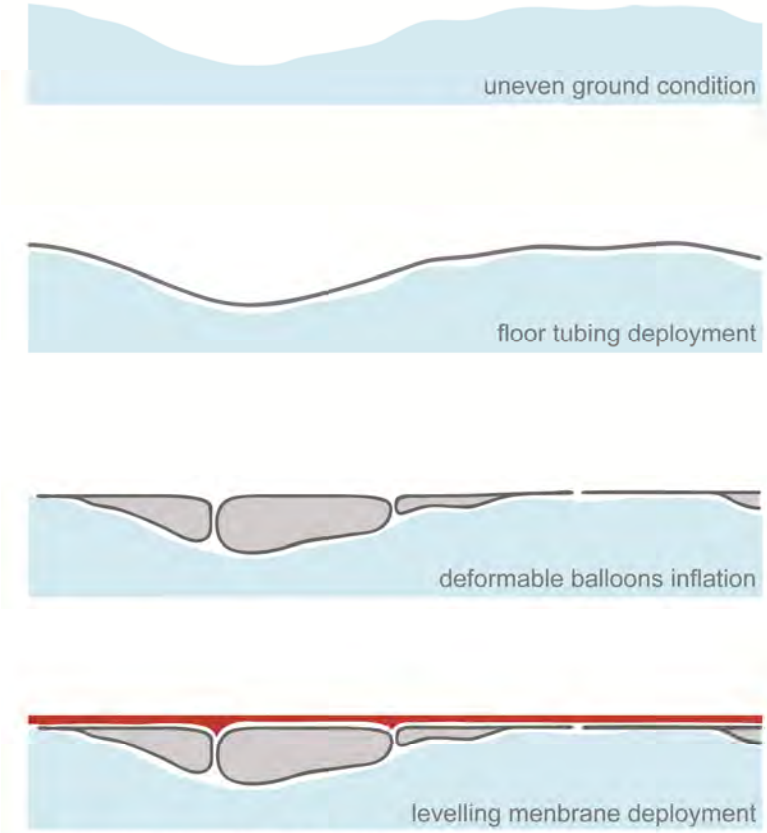




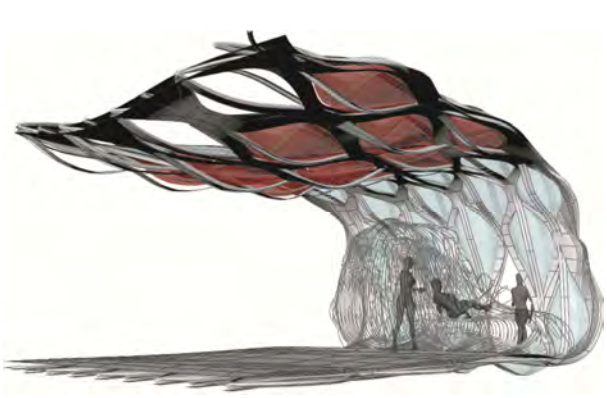


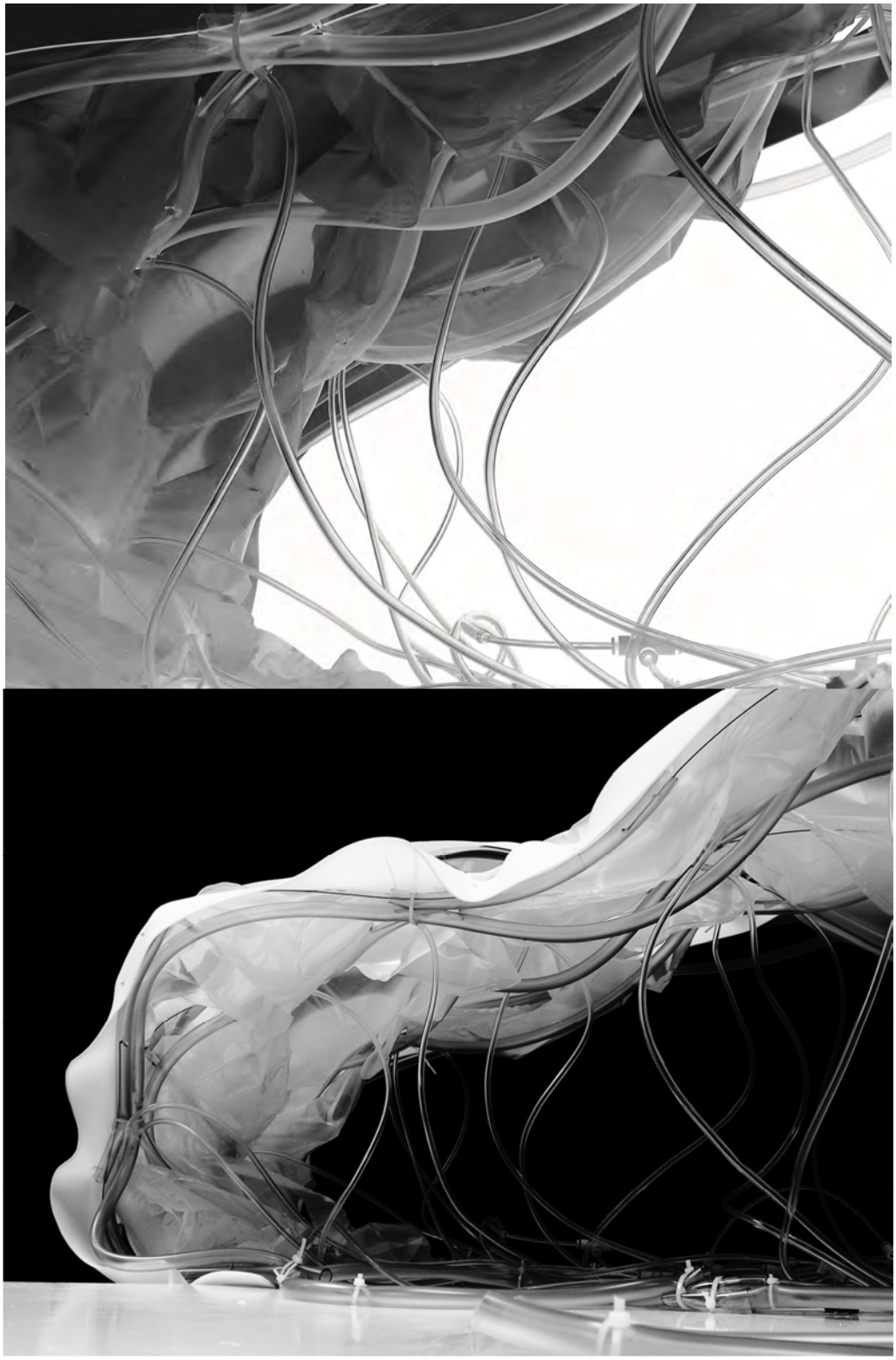
The adaptable floor system will be able to adapt in both systems to several range of different floor conditions. The floor of the triage is composed for a tubing system and latex deformable balloons. The balloons are full with polyurethane foam that when inflated, creates solid foam configuration a rigid base for the floor.

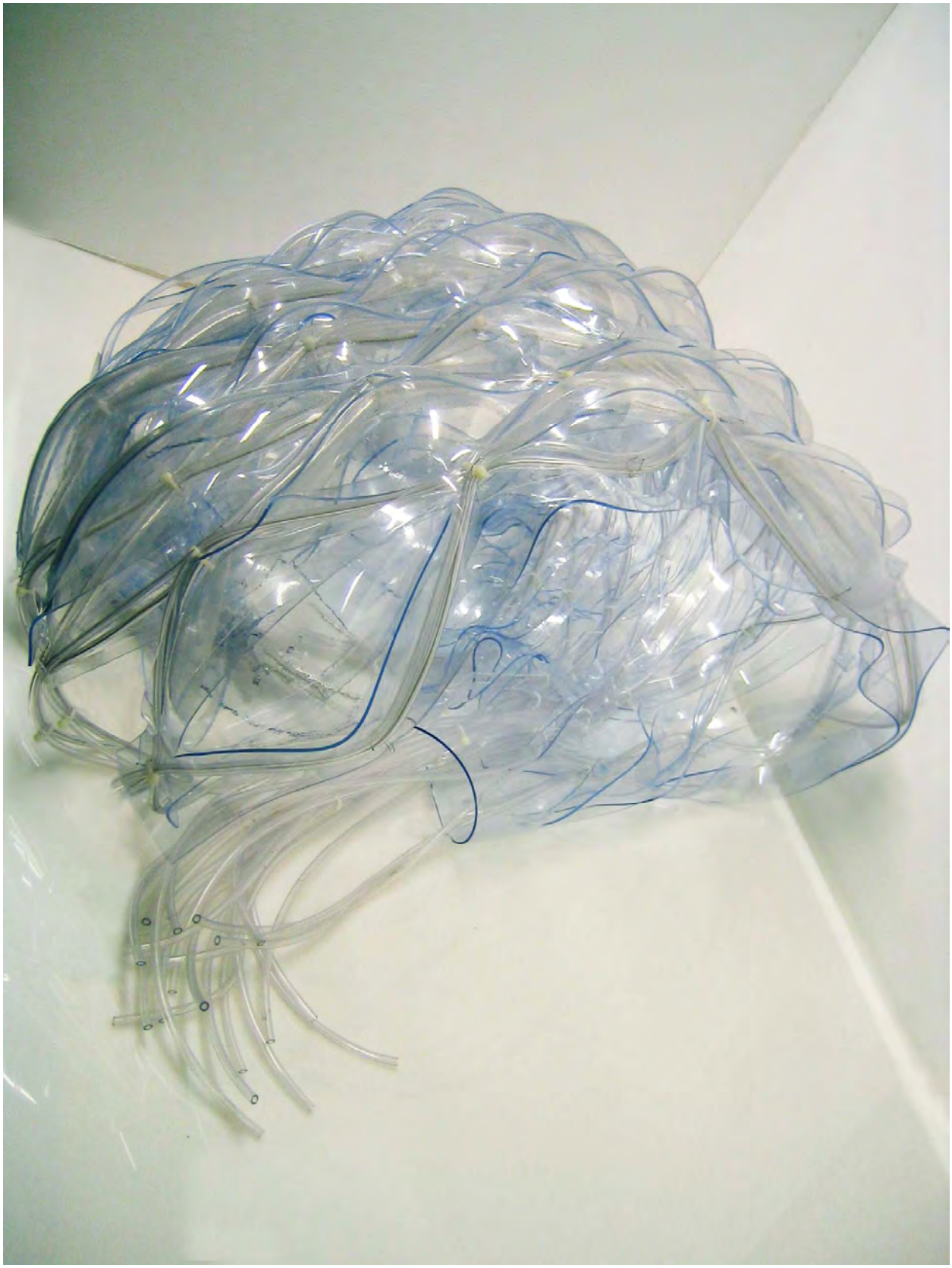
On it, a bag o Phase Change Material, will configure the horizontal surface provocating this phase change state when already deployed with an electric current trough it.

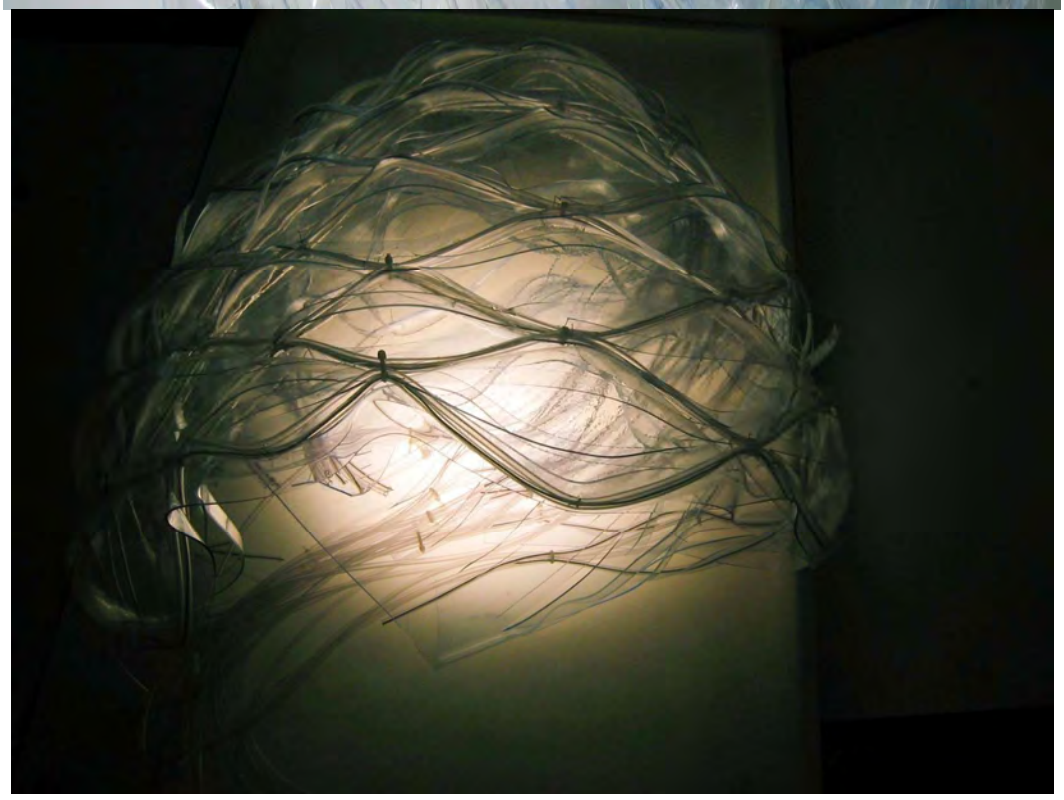


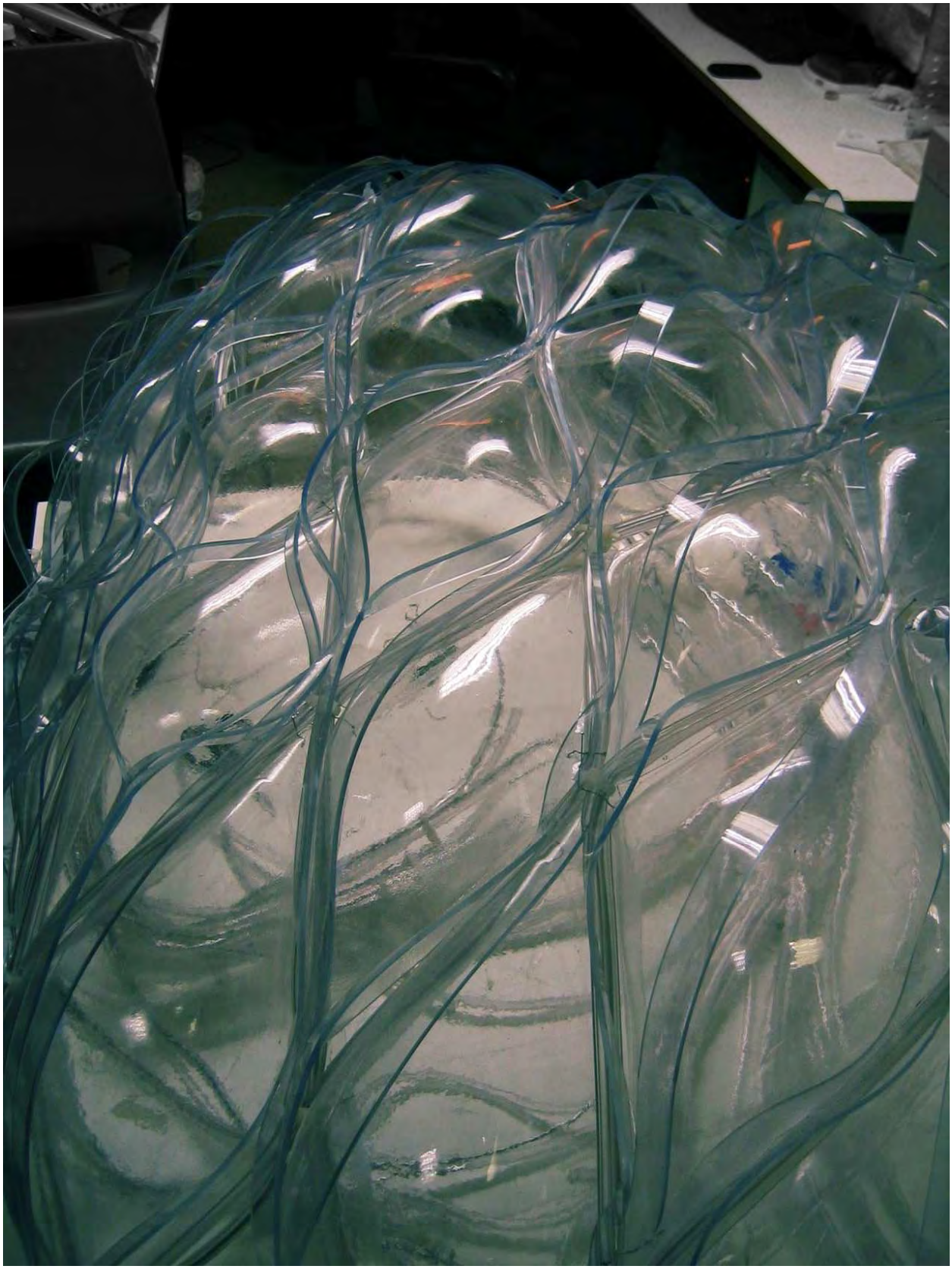
Floor system adaptability

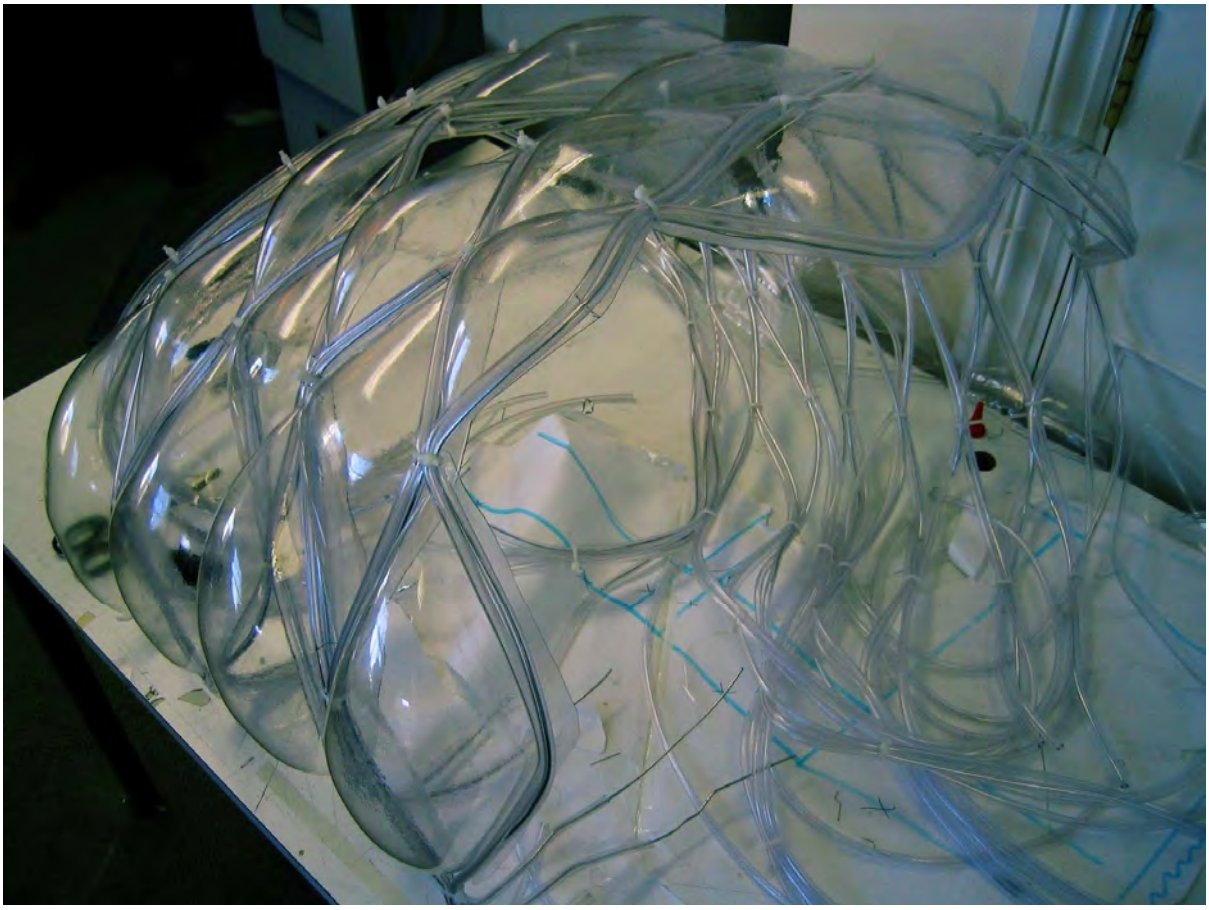












[Note of the author:

As part of my interest in improving my Artificial intelligence and Machine Learning knowledge for the software that will be programmed for EmDeplo learning, I did a research stay at the Bartlett School, at UCL, joining the program "Adaptive Architecture & Computation". Trying to develop my Computer Scientist skills, to be able to apply them to architecture, and, I also coursed at Stanford Engineering (on line), "Introduction to AI" with Professor Peter Norvig and "Machine Learning" with Professor Andrew Ng.

Both approaches were carefully decided. First , The Bartlett, as the main Research Institution in ML applied to architecture, and, second, Stanford Engineering, as the world's leader institution in robotics and AI, located at the heart of Silicon Valley, World Centre of Machine Learning development.

I specially want to thank Mr. Sean Hanna for his support and Mr. Ng for his enthusiastic approach to a discipline that is not always easy to develop.]

2.3 Machine Learning?

1. What is Machine Learning?

Arthur Samuel defined machine learning in 1959 as the field of study that gives computers the ability to learn without being explicitly programmed.

During the last years it has increased the idea of using Artificial Intelligence for giving computers new capabilities.

Tom Michell proposed it like a well-posed problem:

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

CPU of a computer of 2Gz can do 2 billions operations per second versus a brain that has 15 to 60Gz. 10^9 transistors in serial working versus 10^{14} working in parallel. So that, we just currently ca n simulate some structures of the brain to do certain operations.

From classifying your mail as spam or not, to self-driving cars, machine learning is considered to be the basis of most of the fields of human future development.

The approach that will be used of AI in this thesis will be the one in which, in a continuous loop, an intelligent agent will receive data from the environment through some sensors, will change or not its state, and will interact with the environment through some actuators.

The intelligent agent in this case will be the EmDeplo façade and its brain.

It will be a perception-action circle benefiting the adaptability property of the system.

Basic problems of AI can be,

1- Micro worlds. The sum of restricted domains will never be a real environment.

2- Lack of scalability.

3- Robustness. Use to fail in a novel situation.

4- Operating in real time.

As it was commented in previous chapters, also AI proposes the argument of biology as the solution for interacting with the environment. Proposing that neither a human nor a machine can think without the environment.

So that, we can establish as the principles of the new AI:

1- Embodiment. Importance of having a body.

2- Situationism, not micro-worlds any more.

3- Bottom-up design.

When speaking about Artificial Intelligence use to be really useful to define the terms and situations we will be speaking about.

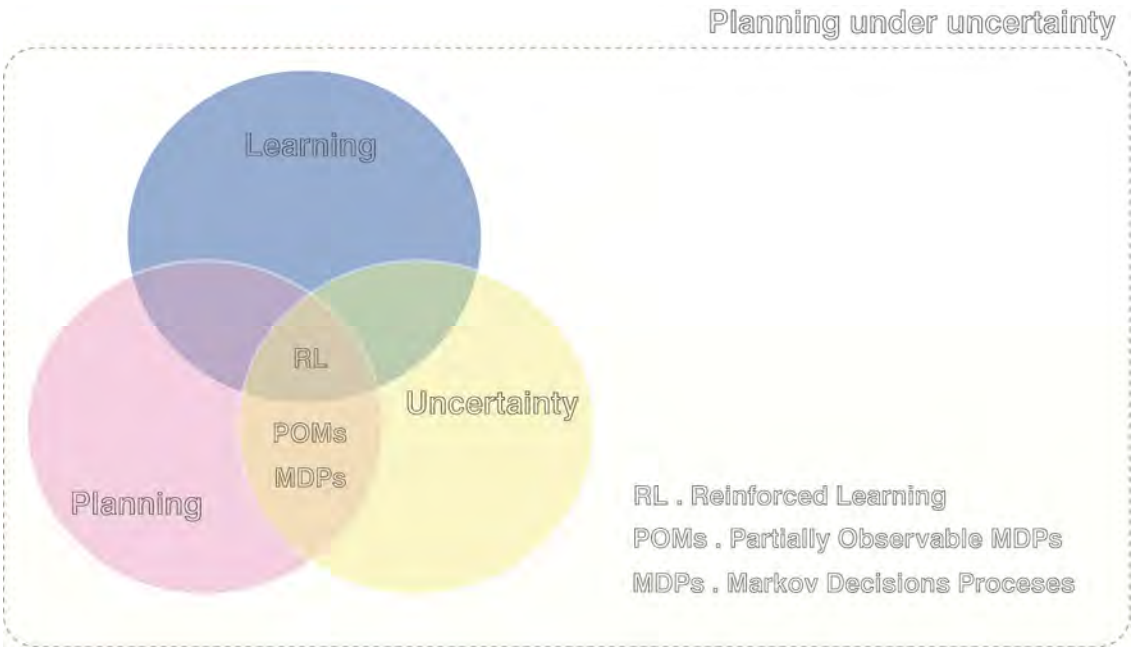
- We must differentiate between Fully Observable vs. Partially Observable Environments. While the first type is sufficient for an optimal decision, having the second kind implies that the agent is only able to observe part of the state and will probably need memory.

- We will distinguish Benign vs. Adversarial Environments, in which it will be difficult to find actions that do not contradict the goal that is trying to be achieved.

- We will work with deterministic(predictable) vs. Stochastic systems

- We should consider that the actions of the system can be discrete (a number) vs. a Continuous set of actions, meaning by that, infinite.

AI will be studied as a method for uncertainty management. AI aim is finding actions for an agent, so, planning is the core of Artificial Intelligence.



2. Problem Solving vs. Planning

The first approach to EmDeplo brain system design will be the simplest one: as a complex system but obtaining a goal through a sequences of actions.

In the very beginning this approach was thought to be appropriate as EmDeplo can be understood as a system which complexity comes from the existence of a multiplicity of states, maybe the climates.

Deciding an initial state, that means, the outside environment, in which the system is in, through several steps is maybe possible to obtain a goal, a second state of energetic efficiency.

The idea of a path of actions to the goal, considering always the path cost, did not seem inappropriate.

Nevertheless, Problem Solving as a method, has demonstrated its efficacy with a fully observable environment and with a discrete, deterministic and known domain. So that, it was considered that it will be a possible procedure for optimization of the façade.

On the contrary it was not considered as a method for developing the understanding and decisions related to the learning of the climate in which the system is in.

Nevertheless, Planning works very good within an environment with partial observablility, multi-agent based and stochastic. It is considered the agent's point of view and its *belief state*. A plan will be made without observing the whole world.

3. Finite State Machines. Memory-Less Planning under uncertainty.

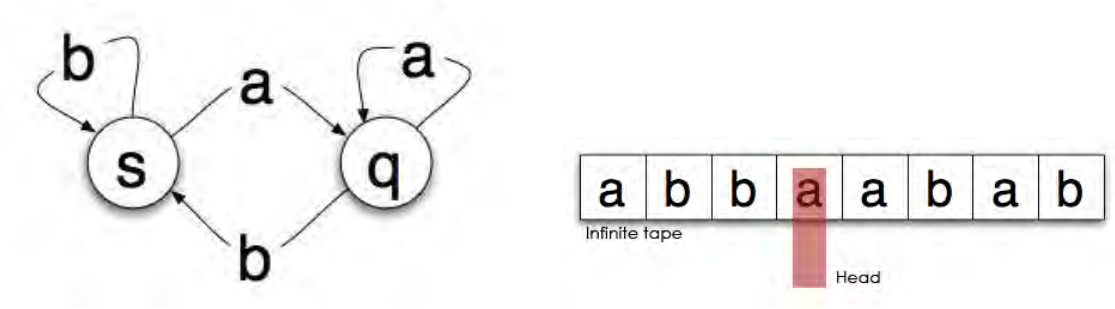
Alan Turing, who is considered the father of AI and Computer Science, proposed in 1936, the "*a*(utomatic)-machine". The Turing machine was able to manipulate symbols, through a series of rules, obtaining an output. This was the foundation of modern computers.

Composed by,

- Infinite length tape with cells that can be blank or a symbol in them.
- A head that can move left to right and read, delete or write.
- A state register, with just one state at a time.
- A transition table, (rules).

Its behaviour was dependant of the current status of the machine and the symbol read in the cell, deciding according to this parameters the next state:

A finite state machine where from each state and a given input symbol the automaton move into next states.

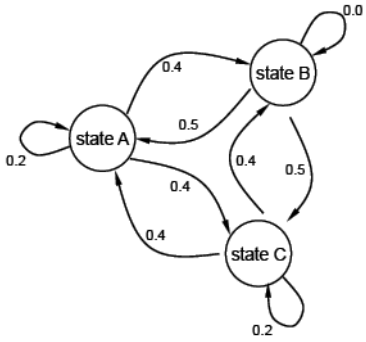


When the outcome, the next state becomes random, a Finite State Machine Becomes a Markov Model.

4. Markov Model

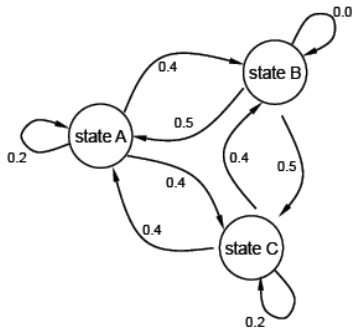
A Markov model is still memory less but provides more options in next state to the goal calculation. States in a Markov Model can be subdivided and increased and used, in the case for example of Hidden Markov Models, in several complex applications as Robotics.

	Autonomous System	Controlled System. Stochactic.
Fully Observable environment	*Markov chain*	*MDP. Markov decision process*
Partially Observable Environment	*hidden Markov model*	*POM . Partially observable Markov decision process*



	A	B	C
A	0.2	0.4	0.4
B	0.5	0.0	0.5
C	0.4	0.4	0.2

Markov models are not a good algorithm for training memory. They exist also Second Order Markov Models were the probability of the next state depends not only on the previous state, but also, to the previous of the previous state, meaning, not only dependency on state t but also on state (t-1).



	A	B	C
AA	.3	.5	.2
BA	.1	.3	.6
CA	0	.5	.5
AB	.5	0	.5
BB	-	-	-
CB	.3	0	.7
AC	.5	.3	.2
BC	.5	.4	.1
CC	.3	.6	.1

Nevertheless this kind of mathematical model results are quite restricted for the learning that EmDeplo system will be suppose to be able achieve.

5. Machine Learning

Machine Learning is a procedure to help the computers to learn models from data. ML seemed since the very beginning a good starting point for the configuration EmDeplo's brain.

Making the system learn from existing, artificial or new environmental models, is the main goal of the façade brain. Knowledge that will give the façade the possibility to adapt to the environment, learning from it and maximizing its efficacy.

Machine learning will use both CONEXIONISM & SYMBOL LEARNING, but always taking into account that a computer can manipulate symbols, but it is not able to understand them, (remember the Chinese Room experiment).

Machine Learning algorithms are basically divided in:

-Supervised Learning

-Unsupervised Learning

-Several others: Reinforced Learning, Recommender systems, etc...

Supervised Learning is commonly used for taking decisions or doing predictions based on the learning through known and labelled samples,

- *Classifying behaviour* will choose between two or more values for the outputs.

- *Regression behaviour* will be used to predict continuous valued outputs.

Finding an appropriate decision boundary will be the basic aim.

Unsupervised Learning is used to learn from unlabeled unknown samples. It main behaviour is as a *Clustering Algorithm*, meaning finding common patterns or parameters within masses of data.

A. Reinforced Learning.

Reinforced Learning is concerned about how an agent take decisions in an environment in order to obtain a maximum reward. The environment usually is modelled as a Markov Model, (mentioned before in this chapter). The actions taken are not right or wrong and the initial inputs are not defined, just the goal and not the reward function. In this way agents should find the best policy. This method allows the agents to obtain a maximum reward even though they do not know what the rewards are.

The learning will be based on what you, the agent, already know and what do you want to do.

Agents can be active or passive, and, depending on they having a fixed policy or they having to decide what to do next.

Fields of application of this kind of learning are multiple.

One of the most interesting is the one applied to urbanism at the Bartlett Space Syntax Department.

Space Syntax was developed at the Bartlett in the beginning of the 90's as a theory based in the idea of studying spaces as set of decision relations. In this way it was possible to study the social implications of a design but also its convenience and way of working.

Depth analysis was proved efficient by Alasdair Turner using agents to study different environment situations. The most famous one is shown above, and was, basically studying the depth and moving patterns in a museum.

He also develop a trial with a more evolved agent that he call the Automata Agent and compare it with the results of a basic Markov Model analysis

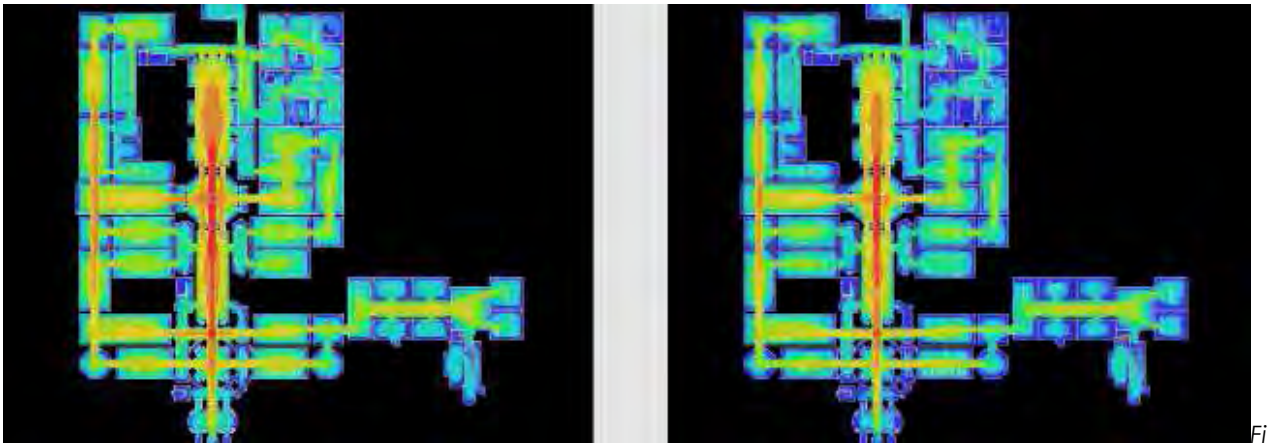


Figure M.
Alasdair Turner Automata. Left, result of Markov, right, results of Automata analysis.

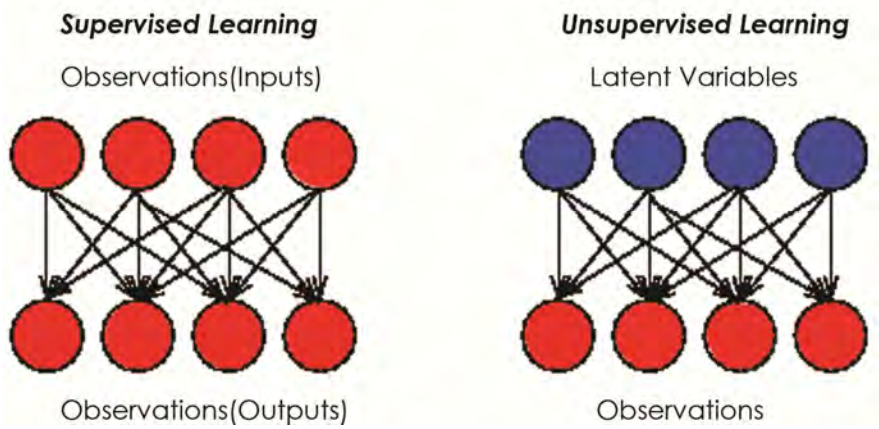
Even though agent analysis has been a very effective learning technique it does not fit in the main interest of this thesis.

The idea of using EmDeplo as an agent, inside in an unknown environment, that has to take decisions for a goal and a reward is clearly different to the learning process that our system must have, as the concept of reward function and goal might vary trough time during the existence of the building.

B. Unsupervised Learning algorithms

- Kohonen Network
- k-means
- Spectral cluster

Unsupervised Learning consists in Clustering Algorithms which labour is to find patterns in unlabelled data. Models of supervised and unsupervised learning differ basically in the observations locations. While in supervised learning the model is based on a set of observations used as inputs and classifies the outputs in another set of observations, in unsupervised learning the model starts from unknown latent variables, and, the set of observations is the end of the causal chain.



Unsupervised learning allows us to work and learn from more complex models than the ones used is supervised learning. While supervised learning is trying to find the connections between two sets of observations, unsupervised learning can work hierarchically into different levels of abstraction.

In this way the output can not be wrong or right, as the observations made will be pattern definitions.

Kohonen Networks

Unsupervised Learning is highly effective in Clustering and dimensionality reduction. Kohonen Networks, for example, imitate the hippocampus way of working creating a navigation map in 2D in a 3D environment preserving the topological properties of the input, this is why they are also called Self-Organizing Maps (SOM).

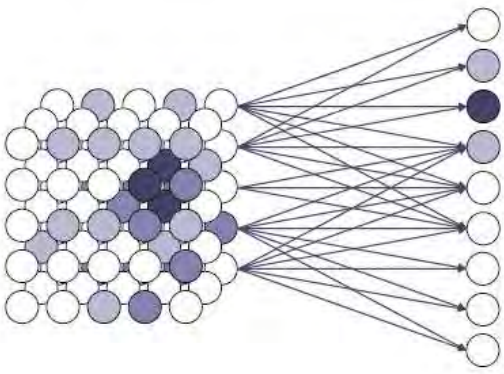


Figure A

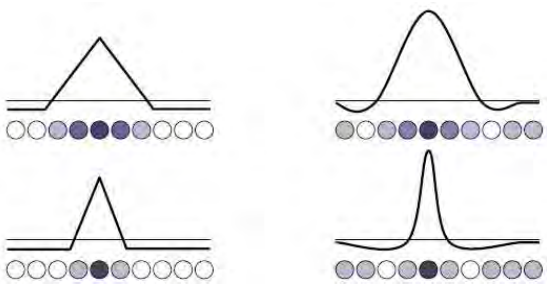


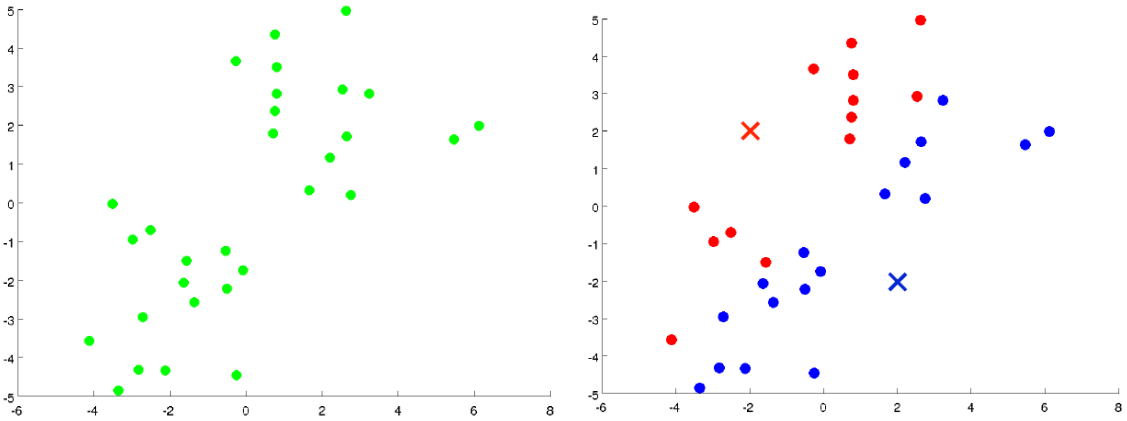
Figure B

Learning by competition, the neurons closest to the winner one have their connections strengthened and the losers, weakened. The method for strengthening are several as shown in the picture above, next to the typical composition of a Kohonen Network.

K-Means Algorithm

K-Means algorithm is the most popular clustering algorithm. It works through the different clusters centroids random initialization.

The inputs will be K, number of clusters, and the training set. For example, having a data set, and using random centroids of the number K of clusters, (K=2 in the example above),



we will do an assignation of the points to one of the centroids, the closest in distance.

Afterwards the algorithm will move the centroids to the average of the location of the points, and repeat the operation until is necessary.

The random initialization can take us to be stuck at a local optima, reason because the trial of lots of random initializations is required. A good recommendation will be, as the number of cluster K use to be unknown, to start with a K < m, being m the size of the training set. Nevertheless this election use to be done, normally, by hand.

Unsupervised Learning Conclusions

After a deep study of most common unsupervised learning algorithms, done through the Stanford Lectures and Adaptive Architecture & Computation Bartlett research stay, it is

concluded that unsupervised learning might be not the appropriate learning behaviour for EmDeplo's initial brain.

Nevertheless, as a basis for further research proposals, it can be considered for a future system behaviour more complex.

The ability of recognizing patterns from unlabelled data can be maybe used as the base of a more complex decision process. Receiving plenty of unlabelled data from the environment, and trying to find patterns in it, for proposing new scenarios for acting, can be an extremely advance adaptability behaviour of the membrane, that will be probably proposed as the basis of a Post-doc research stay.

C. Supervised Learning algorithms

- Linear Regression, Logistic Regression
- Multilayer Perceptron algorithm, ANN
- SVM- kernel

Linear Regression, Logistic Regression

Not having the intention of entering in a pure mathematical explanation of both methods, it will be just mentioned how both models are defined in statistics:

- *Linear Regression* models the relationship between a scalar variable *y* and one or more variables *X*.

- *Logistic Regression* predicts the probability of occurrence of an event by fitting data to a logistic function.

Both models use Gradient Descent algorithm to find local optima.

Linear Regression is a common method in Social Sciences to find relationships between important variables (ex. increase of housing prices with house size), both of them used to make predictions.

Logistic Regression is commonly used for multi-class classification problems. For establishing patterns of behaviour, relating for examples the size of a tumour with the probability of being malign (1) or benign (0), using o and 1 as positive and negative classes to distinguish.

The problem with these algorithms appears when the size of the features array is really big. The probability of over fitting increases and we will be dealing with an extraordinary number of parameters. For example for a medical prediction based in 100 parameters:

x_1= *size*

x_2= *age*

.....

x_{100}= *wealth*

We will be dealing with approximately 170.000 features. That makes this process clearly unreachable even if we are using just subsets of the training set.

$g(thetha_1*x_1 + thetha_2*x_2 + theta_3*x_1*x_2 + thetha_4*x_1^2*x_2 + theta_5*x_1^3*x_2 + theta_6*x_1*x_2^2+....)$

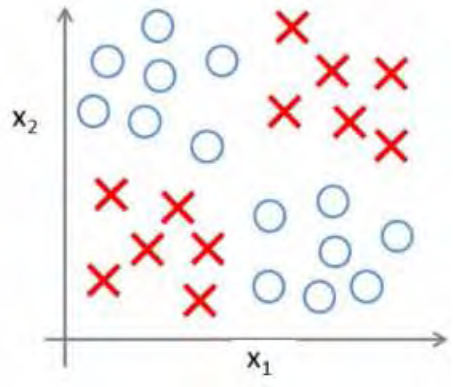
In this kind of situation, Linear Regression is highly not recommended.

Support Vector Machines

An alternative view of Logistic Regression are Support Vector Machines.

As a non-probabilistic linear classifier it is a kind of algorithm that can be taken into account for the decision of EmDeplo's brain configuration.

SVM work with the Kernel Models, representing the labelled samples in a n-dimensional space making them linearly separable. SVMs construct an hyper plane, or a set of them, in a higher dimensional space.



Mapping is done through Kernel functions, through the kernel trick:

for a matrix A,

$$A\varphi = \lambda\varphi$$

λ is an Eigen value
φ is an eigenvector

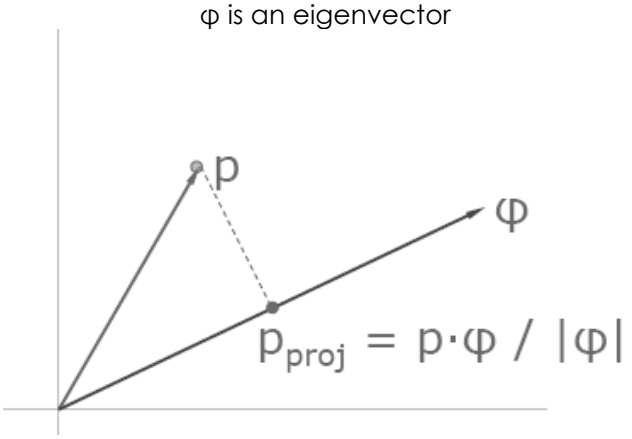
the dot product,

$$[A\ B\ C] \cdot [a\ b\ c] = Aa + Bb + Cc$$

The projection of a point p onto a new axis x will be:

$$p_{proj} = p \cdot \varphi$$

p is a point
φ is an eigenvector



For example, it will map a input space X to feature space F using mapping function Φ:

$$x = [x_1, x_2] \rightarrow \varphi = [\varphi_1(x), \varphi_2(x), \varphi_3(x)]$$

being

$$\varphi_1(x) = x_1^2, \varphi_2(x) = x_1x_2, \varphi_3(x) = x_2^2$$

going from a 2D space to a 6D space,

$$(x_1, x_2) \rightarrow (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2)$$

Not all similarity functions will be valid Kernels because they must satisfy *Mercer's Theorem*. There are multiple Kernel possibilities:

-Linear, Gaussian, Polynomial, String, Chi-square Kernel, etc.

For example, using a Gaussian Kernel.

$$f_i = \text{similarity}(x, l_i) = \exp(-|x - l_i|^2 / 2\sigma^2)$$

so for a training set of m examples in 2D:

$$(x_1,y_1), (x_2,y_2) \ldots\ldots\ldots (x_m,y_m)$$

$$l_1=x_1, l_2=x_2, \ldots\ldots\ldots l_m = x_m$$

For a training example (xi, yi) example x, $\mathbf{f \in R^{m+1}}$:

$$f_0=1$$

$$f_1 = \text{similarity}(x, l_1)$$

$$\ldots$$

$$f_m = \text{similarity}(x, l_m)$$

Considering for example $y \in \{0,1\}$

, the main property of Logistic Regression is $0 \leq h_\Theta(x) \leq 1$ and the basis of Logistic Regression,

$y = h_\Theta(x) = g(\Theta^T X) = g(\Theta_0 + \Theta_1 x_1 + \ldots\ldots + \Theta_m X_m)$

$g(z) = 1/1+e^{-z}$ (being z a real number)

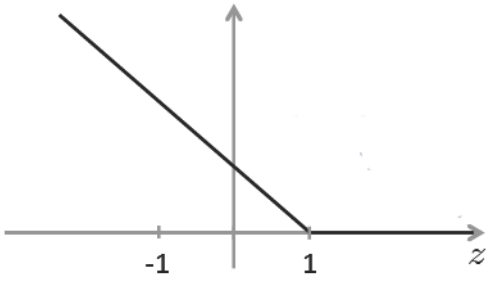
obtaining,

$h_\Theta(x) = 1/1+e^{-\Theta T * x}$

SVM will predict:

$y=1$ if $\Theta^T * f \geq 0$

$y = 0$ otherwise.

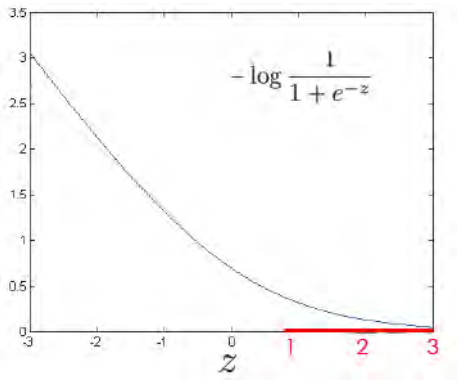


(being $\Theta^T = [\Theta_0 + \Theta_1, \ldots\ldots\ldots \Theta_m]$)

That supposes a huge different with a Logistic Regression Classification, where:

$y=1$ if $\Theta^T * x >>>>>>> 0$

$y=0$ if $\Theta^T * x <<<<<<< 0$

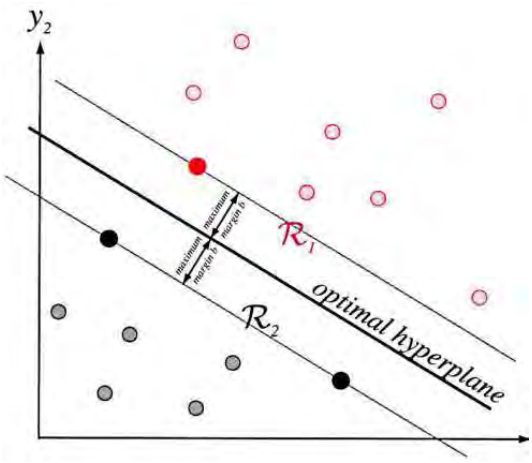


SVMs proposes a much better error minimization.

The best separation is the one of the hyper plane that has the largest distance to the nearest training data point of any class: functional margin.

SVMs are trained on the worst classified examples, known as *support vectors. This support vectors will be the samples on the margin.*

The lower the generalization error of the classifier is the one with the larger margin. This large margin around the decision boundary guarantees us an smaller error that in conventional Logistic Regression.



Logistic Regression vs. SVMs

Being:

n=number of features ($x \in R^{n+1}$)

m=number of training examples

- If n is large (relative to m):Use Logistic Regression, or SVM without a kernel ("linear kernel").

- If n is small and m is intermediate: Use SVM with Gaussian kernel.

- If is n small and m is large: Create/add more features, then use Logistic regression or SVM without a kernel.

A Neural Network, on the other hand, will be likely to work well for most of these settings, but may be slower to train. According to that it results, a priori the best algorithm to try for EmDeplo's brain.

Nevertheless it is considered that, for a real building implementation, the slow training speed can be a problem, but, for the development of this thesis this is not a basic disadvantage. In this way an ANN seem appropriate to start making the system working not having to be worried about number of features and training set sizes.

Artificial Neural Networks

An Artificial Neural Network is a mathematical model inspired biological neural networks. An ANN is a computational model of the brain trying to approach the capacity of learning from experience and taking decisions of the human brain. ANNs are normally adaptive to external environment learning from the data received from it.

So that an ANN approach is considered to be a great base system for EmDeplo brain.

ANN conexionism remarks parallelism between mind processes and brain processes: biology.

A Neural Network is formed by groups of artificial neurones configured to perform some calculation. Neurons will be Activation Units with inputs/outputs.

On the other hand they emit numerical signals while biological neurones emit pulses, and they are just one kind while brain has many different kinds of neurones. In this way ANNs do not store information in just one place, information will be stored everywhere, being able to learn association between patterns of experiences.

ANNs are holistic devices for solving a problem. All properties of the system will not be able to be determined or explained by the parts of it on their own.

Modern neural networks are non-linear statistical data modelling tools trying to simulate the brain parallelism way of working and capability of learn by training and pattern recognition, by feed forward and back propagation.

McCulloch-Pitt defined the basic unit in 1940's, trying to approach the basic way of working of a human neuron: *Threshold Logic Unit (TLU)*. Able to deal with logic operations its work was to do a weighted summation of the input variables for through a non-linear function to obtain an output. It was simulating a Finite Automaton.

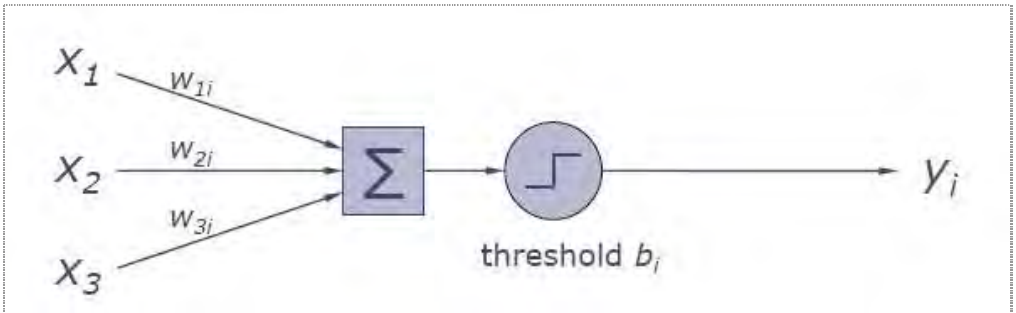


Figure A

$$y_i = \text{sign}\left(\sum w_{ij}\, x_j - b_i \right)$$

The representation of the threshold values as a bias term was introduced by Widrow in 1960.

Rosenblatt in the 60's developed the Perceptron, that uses, a linear threshold function, making the weights more flexible and improving the adaptability of the process. It is considered to be the simplest kind of feed forward Network. On the other hand, it was a binary classifier, not being able to deal with XOR functions.

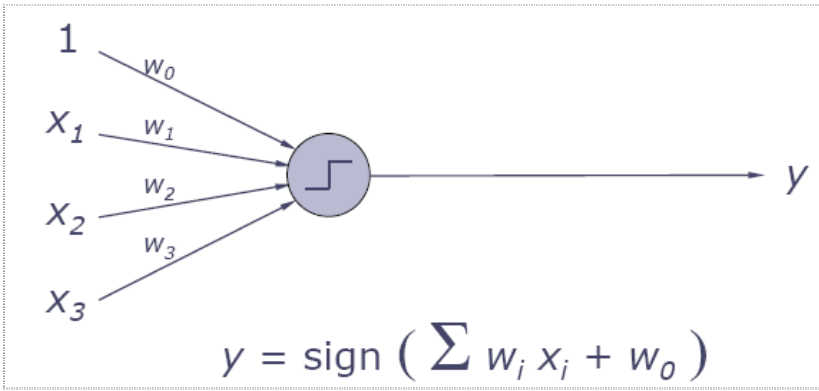


Figure B

During the 70's ANN research was stopped, occurring what was called the AI Winter. AI winter was caused in part by the publication of the book "Perceptron" by Minsky, book that was establishing some mathematical implications and impossibilities of ANN operations.

Some years later it was proposed that a perceptron with more than one layer, the Multilayer Perceptron, will be able to deal with non-linear separable operations.

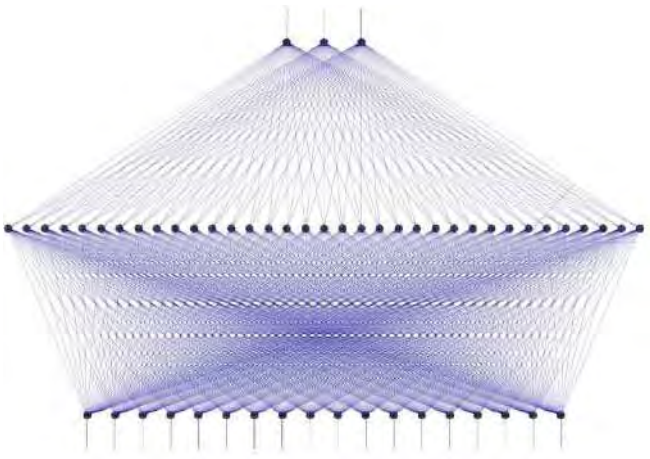


Figure C

In the 80's it was proposed the idea of *Back propagation*. This proposal broke the dead end path experienced by the ANN research the previous 10 years.

In this way inputs that were not linearly separable in the beginning, trough the new idea of having a n-dimensional intermediate layer of hidden neurons, became able to be mapped and classified.

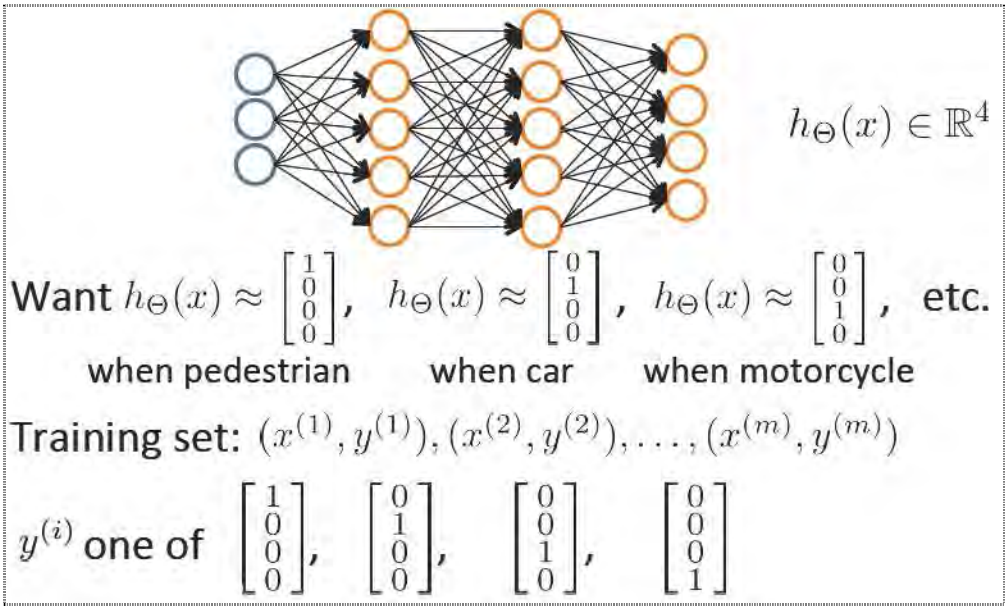
So that, loop networks with feedback were soon studied for define dynamic systems with memory.

Back propagation was the definitive alternative for adaptability. Using gradient descend and initialized with random weights and calculating the errors in each step, it allows the system to readjust the weights for the next sequence.

Next example is based on Andrew Ng basic Lectures example of a simple ANN classification problem. The system is trying to classify four different kinds of pictures, pictures from cars, pedestrians, trucks and motorcycles.

So:

number of neurons output layer = number of classes, (4 in this case)
number of neurons input layer = dimension of features x_i
hidden layer reasonable default:1 hidden layer, or
if >1 hidden layer, layers must have same no. of hidden units in every layer



This kind of Machine Learning algorithm seems, a priori, quite appropriate for EmDeplo's brain. The ability to distinguish between different kinds of environments and situations can probably be solved with an ANN.

3. Initial concerns & objectives about EmDeplo Morphogenesis.

Real vs. virtual imaginery

The main aim of many current practises and schools is the imaginery obtained through parametrical practise. There are two kinds of understanding parametrics. The first one, is the one that does not understand that parametrics, technology and new media have just changed not only our architectural design process, but also the definition of what architecture nowadays is. They used to just try to reach and image, a great image of a shape that obviously has to look parametrical but not necessarily has to be parametric:

The non-negotiable necessity of an image.

Why is so necessary to desperately create an image that look parametric if really the parametric scripting of the behaviour is not really helping us in the design, during the process or during its performance?

Same question can be applied to the technology concept and process of designing the media-Tic building.

Form is not shape, and parametric architecture does not mean parametric geometry. Emergency deployable is parametric architecture, but not just parametric shape. Architecture is not just form but a complex relation of forms.

Initial Objectives

The basic research that will have to be done for completing the viavility of the whole system are 3:

1 - Development of the software for the Interface at the factory that allows the cuztomization of EmDeplo when being fabricated and packed.

2 - Development of 1 to one model or proper software for demostrating Em Deplos structural behaviour accurately.

The difficulty of solving the simulation of the behaviour of the deployment and assembling of the EmDeplo membrane system of this thesis is, clearly, a goal. But it is also a problem. The kind

of simulation that it will be possibly reached will be so rude that will have for sure important gaps in the real physical membrane behaviours.

After working with Burohappold Engineering over the last years, it was decided that it will be necessary to design a new kind of software to simulate membrane structural behaviour with the 5 layers of the membrane working together. Thing that was not possible without external money investment. It was at that point where the consultancy with Burohppold Engineering ended.

So at that stage seemed that the only valid no-error path to demonstrate its behaviour should be a one to one model. A one to one model studying two different behaviours:

- Structural & deployable behaviour.
- ETFE pillows performance (shading & insulation).

So that, it was proposed the building of a one to five piece of the membrane. Piece to be built complete, with sensors and light meters, that will be connected in a controlled interior environment box, to try to demonstrate that the system will be able to adapt to the environment effectively, learning from it.

3 - Development of EmDeplo's brain.

This three initial objectives were demosntrated impossible to its cmplexity to develop on a unique PHD thesis, so it appeared the need of doing an election within this objective to develop.

The first point for development was soon discarded considering the design of the factory software not ppropiate for an Architectural Thesis. It is considered nevertheless appropiate for a Computer Engineering thesis or focused research.

The decision between the other two needs of development was at this stage strongly considered, being this decision a very important step at this point for the thesis structure.

Considering point 2: Physical System behaviour definition

As it was shown in previous chapters EmDeplo was born within emergency scenarios. The possibility to have emergency shelters without insulation needs or with insulation costs highly reduced was quite interesting since the very beginning.

ETFE and PTFE give us the possibility of obtaining high insulation qualities and also, if needed, high transparency, being also possible to control the shading. The pneumatic cushions are the base of an intelligent insulation system.

EmDeplo will work with a system of pillows of three layers, being the internal one patterned for controlling shade. Sending a pump for controlling the low pressure needed with the package units for the triage, we will have the possibility to control a whole shelter.

ETFE is really resistant to pollution, sunlight and does not degrade, resisting temperatures from -140ºC to 260ºC, which make it quite appropriate for a emergency shelter in an extreme climate and also make it resistant enough in the case of the shelter having to stay at the emergency scenario for a longer period.

The membrane will also be using this material because of its fire properties, as it does not generate flame or smoke just disappearing. On the other hand it remains clean with the rain and resists quite good wind. EFTE is recyclable and meltable , and it also allows a production line computer based.

"We need a worldwide technical organisation and new inventions to improve our buildings so that they can survive catastrophes. We need a system that allows us to react quickly (…) and we need instant buildings for those people who are affected by catastrophe".
(see # 62, Wiley-Academy, 2004)

The thermal conductivity of ETFE is quite high and does not vary with when temperature increases. Nevertheless λ in gases under low pressure tend to increase.

With a λ of 0,027 W/m*k ETFE is considered to be a thermal insulator. Nevertheless with a thickness of 3µm is obviously producing not a very high Thermal resistance, even having the three closed layers.

$R_{closed\ pillow} = e/\lambda$ $R_{closed\ pillow} = 3*10^{-6}/0{,}0027$ $R_{closed\ pillow} = 0{,}000125\ m^2*K/W$

With a normally inflation with a low pressure air, and not considering the minimal changes of a non-flat surface:

$R_{open\ pillow} = (3*10^{-6}/0{,}0027) + (0{,}35*2/0{,}02)$ $R_{open\ pillow} = 35{,}000125\ m^2*K/W$

So that, the insulation is highly increased when the pillows are open and inflated.

It is also possible to consider the use of other gases with better insulation properties than air. The gases considered should be safe and obviously non flammable, and, of course, non toxic for human.

In that sense sounds quite sensible to consider inert gases as a possibility for ETFE pillows inflation. With a λ even lower than the air, 0,017, Argon is a good election for these cases of insulation. Used in some window glasses for incrementing their insulating properties have never been implemented in a system of this kind.

$R_{open\ pillow} = (3*10^{-6}/0{,}0027) + (0{,}50*2/0{,}017)$ $R_{open\ pillow} = 58{,}82\ m^2*K/W$

Obviously two factors must be taken into account. The first one is the need of developing more exact calculus not underestimating the effects of the curvature of the pillows surfaces.

$$R'_{closed\ pillow} = 1/2\pi * \sum_{1...n} [(1/\lambda n) * \ln (\text{diameter}_{ext} / \text{diameter}_{int})]$$

The second one is the budget. For an emergency system, the cost of implementing low pressurized Argon may not be possible to assume. Nevertheless, in a building with such a high budget as the media-TIC budget, this kind of considerations should have been taken into account.

From digital to physical process

"Membrane is a thin, synthetic or natural, pliable material that constitutes the lightest material means for spatial organization and environmental modulation"
Michael Hensel & Achim Menges. (see # 63, Wiley-Academy, 2008)

Air, as an architecture material is a quite recent idea. Except from the uses of the roman of animal skins inflated for snorkelling or for air mattresses, it was not until the 19th century when, the glass houses appeared, when it was really taken into account.

Joseph Paxton Crystal Palace was the design that, make the idea of a really light and transparent material with a structural frame, to explode and to start being considered in architects minds. This lightness obsession will dominate afterwards the 20th century.

Vector-Foiltec was the company that introduced and generalized the use of PTFE and ETFE.

ETFE is known since the 40's when DuPont registered a US patent, nevertheless, it was not commercialized until the 70's.

ETFE was the result of a DuPont research for an industrial material resistant to abrasion, friction, radiation and to really high and low temperatures.

Architectural interest in ETFE was highly influenced by the 70's oil crisis, when the need of findings about new materials for replacing oil with solar energy.

Fullers Dymaxion House incorporated in 1929 some pneumatically stressed structural components but it most influential work for pneumatics will be his pneumatic geodesic dome.

Fuller's pneumatic geodesic dome
(see # 154, LeCuyer, 2008)p.18

During the 50's some examples of air supported structures appeared, mostly cover for swimming pools and small temporary designs.

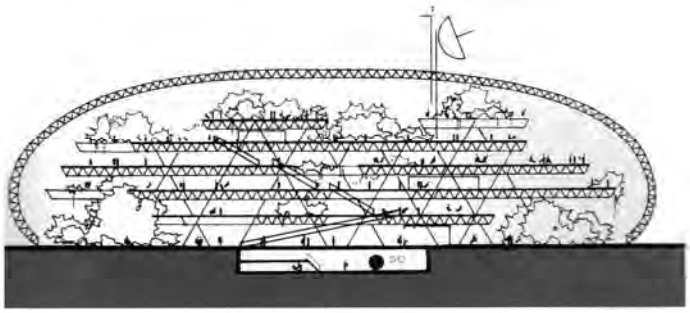
Nevertheless, in the 60's it started a huge research in pneumatic ideas. The US Atomic Energy Pavilion of Victor Lundy was a very good example of the possibilities in the future of pressurized structures, using two layers for the bubble, divided and pressurized.

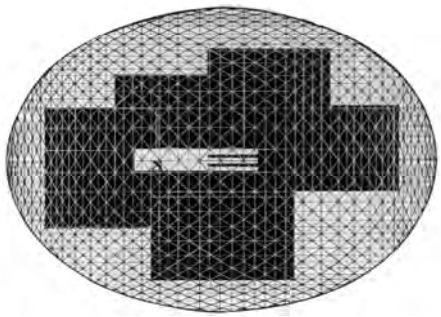
(see # 154, LeCuyer, 2008)p.32

The 70's lived an explosion of pneumatic structures with a massive culmination at the Osaka's Expo 70, where the largest collection of pneumatic structures until that moment where showed.

This decade left us very interesting collaborations as the one between foster and Fuller creating the *Climatroffice*.

The Climatroffice was a mixed structure for several used created within a climatic envelope.

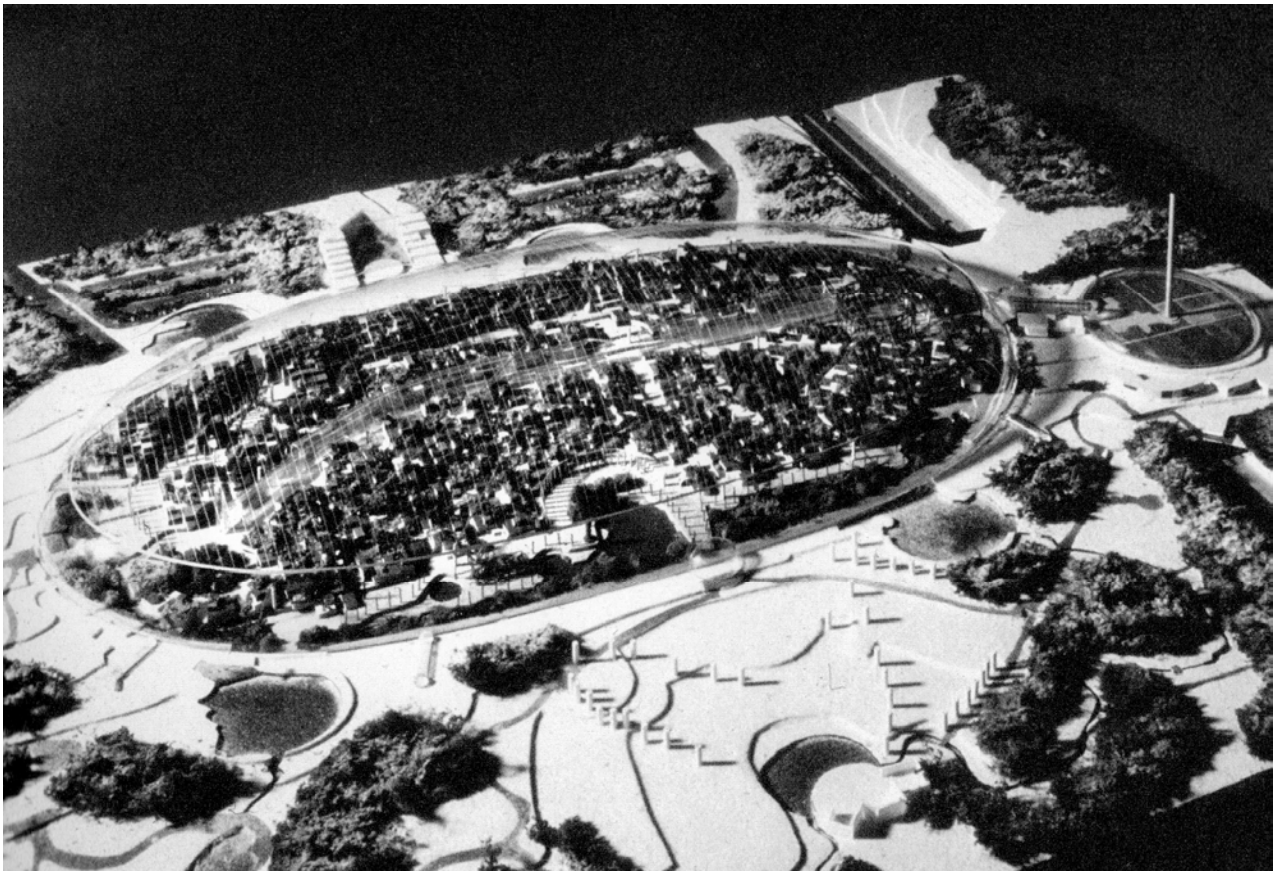




(see # 154, LeCuyer, 2008)

In 1984 after a decade of testing showing no loss of its properties during time, ETFE was ready to be used in the architectonical field.

In 1980 Burohappold Engineering was working in the coverage of a city in the artic, "58 degrees North", with Frei Otto and the University of Bath. at the first stage the industry proposed the PTFE as the material for the covering, but, after, considering the need of more transparency that the 10% the PTFE offered, FED and ETFE appeared as the new materials proposed. By that time, the FED cover of the Burger's Zoo in the Netherlands collapsed, leaving ETFE as the material that looked more appropriate for the project.



58 Degrees North, Artic City.1980.
(see # 154, LeCuyer, 2008)p.7

The structure was finally configured with cables and stainless steel frames.

About then, Vector Foiltec was contracted by Burgers' Zoo in the Netherlands for a series of improvements in their installations.

The first of them, Mangrove Hall, needed an urgent replacement of the FED (fluorinated ethylene-propylene) cable structure that had collapsed. It was substituted by 45 ETFE cushions reusing the previous cable and steel structure.

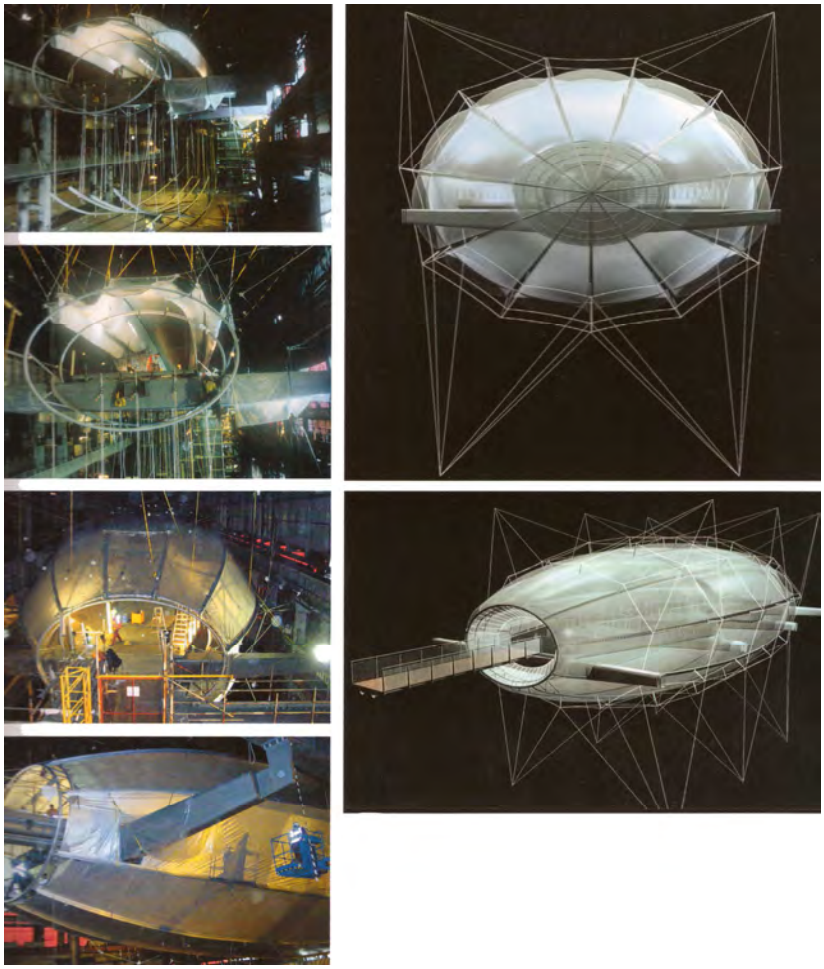
The success of this hall was followed by the Tropical Hall in 1988 and the Desert Hall in 1993.

From left to right, Mangrove Hall, Desert Hall and Tropical Hall. Burgers' Zoo.
(see # 154, LeCuyer, 2008) p.34

With a covered area of 6000 square meters, the Hampshire Tennis & Health Club, is the first example of ETFE use innovations and research using for a first time an uniaxial cable net.

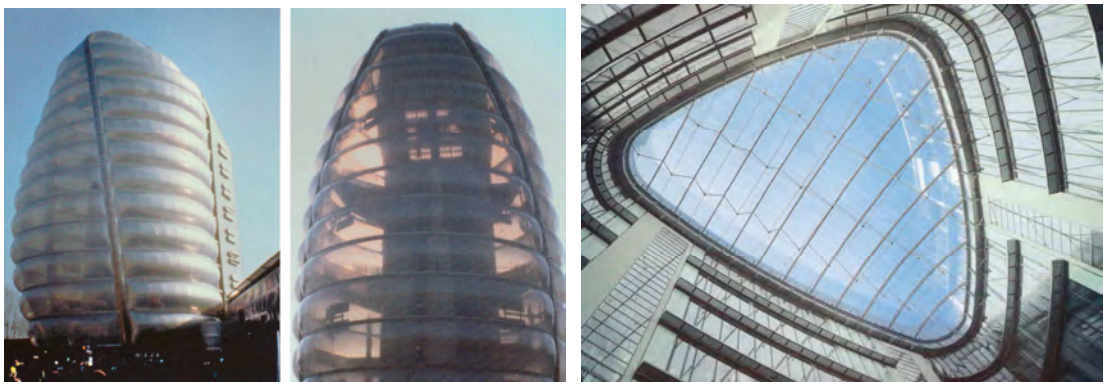
Hampshire Tennis & Health Club. 1995
(see # 154, LeCuyer, 2008) p.50

Sample of this increasing interest in research in new methodologies for ETFE use is also the gorgeous Air Pavilion at Magma. The structure is based in two compression rings at the end assembled with a diagonal structure



Pavilion at Magma.2000.
(see # 154, LeCuyer, 2008) p. 57

Other well know examples can be the Grimshaw's National Space Centre at Leicester finished in 2001, as the first completed ETFE façade and the cover of the Baseler Plaz in Frankfurt in 2004, first example of use of ETFE with a perfect equilibrium tension-compression.



(see # 154, LeCuyer, 2008) p.51

First steps in fabrication

The election of ETFE for the EmDeplo system was based in part for its extremely good sustainability. A single inflation units, that can control more that 1000 m2 of façade, operating by average half of the day time, uses the amount of power similar to the one a light bulb needs.

Problems?

EFTE is a material that required specialized handling and designers. Problems shown along its history as an architecture material can be dealing with heavy snow loads, as it has recently happened with the Allianz Arena badly design drainage details.

A common problem that can contribute a ETFE system to collapse, is the over-design. For example, if foils are to thick, the can become fragile and tend to fail.

It is also important to consider that ETFE suffers loss of strength at extremely warm temperatures, 70ºC, reason why, it is recommended to use smaller sizes of cushions in warm situations.

ETFE is also acustically soft offering almost an inexistent resintance to sound energy, allowing to be crossed for all external noises; nevertheless, on interior, ETFE cushions abosorb more that reflect sound what makes them a much more better than glass in that sense.

Advantages?

Apart from its already mentioned durability and sustainability characteristics, the great advantage of this kind of structure is that the inflation of it starts a load reduction cycle.

Also in ETFE cuts does not propagate and can be easily fixed with ETFE tape.

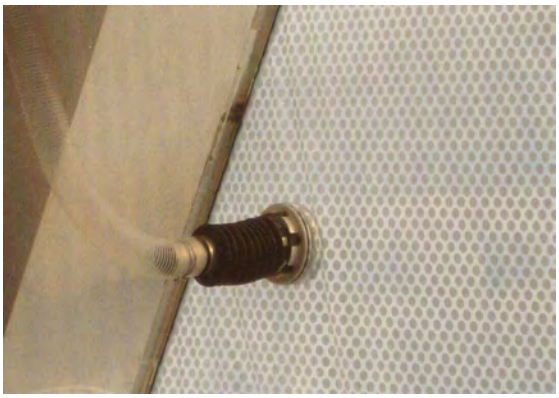
EFTE success is also based on its high transparency, a rate between 90 to 95 percentage transmision, and on the load reduction compared with glass load for the same suare meters.

Also the pnematic ocntrol software can be understood as the next step in engineering forgetting the old fashion mechanical systems.

On the other hand it has also a huge advantage of safety. In a fire, EFTE is self-extinguisable and does not product toxic smoke unless the temperature is higher than 800ºC, with is a very big advantage when compare to other plastics as for example PVC. the material disaoears not falling down onto the building ocupants.

Safety under other hazards has been studied lately concuding that ETFE cushions work well for eartquakes and explosions, based on its resistance and the property of not cleating sharp pieces in case of collapsing, being a much more safe material than glass.

Using ETFE the reduction of maintenance and price cost, its long and environmental performance and its recyclability are clear advantages in its use in architecture.



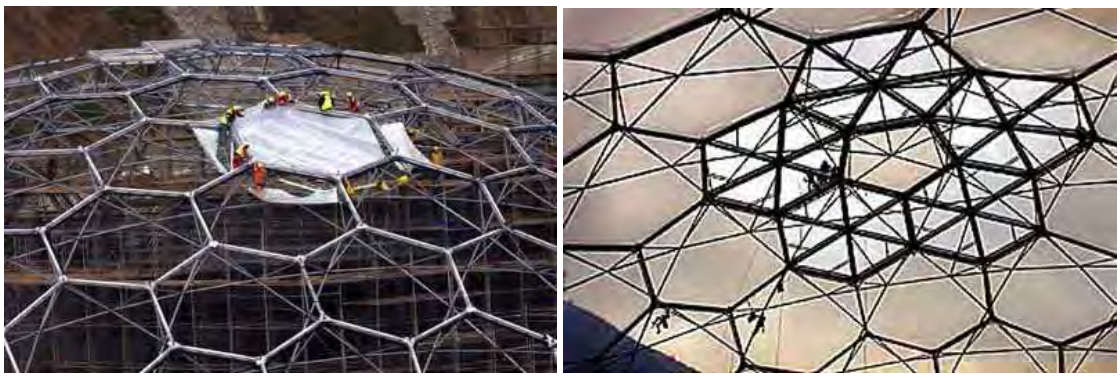
Typical ETFE pressurized system
(see # 154, LeCuyer, 2008) p. 37

One of the first steps of this research was to contact the company and establish a consultancy. The consultancy will be carried out by Iaso, the distributor of the membranes and pillows in Spain. Consultancy will be done by Mr. Patrick Vaillant.

Starting to work with the consultant, it was concluded that the possibility of creating a membrane of 100 pillows was real. The only problem was the fixing procedure.

EFTE pillows façades have been done till today as fixed rigid structures with really strong aluminium joints. ETFE fabrication process is a combination of handcrafted and non-linear analysis software packages. A typical design will include from two to five layers of different foil thicknesses

Since the very first implementation of ETFE at the Eden Project, this rigid metallic structures have been the basis of their structure. Remeber for example the very similar method used with ETFE cushion more recentl for the Water Cube For Beijin Olympics.



Figures 16. The Eden Project.

So, proposing to the factory the idea of a new flexible membrane of pillows, with a light join was an important aim. Obviously, Iaso studied the possibility, physically speaking, and not considering, for the moment, the high increase on a building budget due to the implementation of a non-standard method.

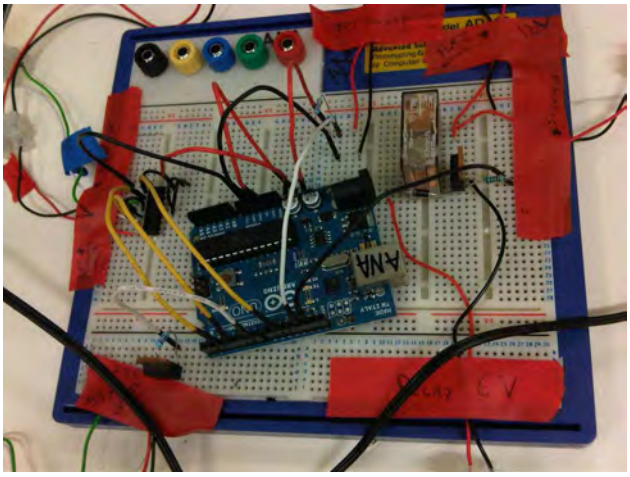
Mr. Vaillant concluded that it will be possible to develop a trial model only if industrially sealed. ETFE has Fluor within its composition, (*Copolymer Ethylene Tetra Fluoro Ethylene*), what makes completely impossible for any kind of glue working properly. For the same reason, also non-industrial sealing was eliminated as an option.

After going through some ETFE layer samples, it was decided to reduce the possibilities to three of them. Thickness was an important characteristic and the minimum thickness reachable was 100 micras, which will be a perfect thickness in a real size model.

Figure 17

The three types chosen where the white-transparent one and two different densities of patterned types, both of which were going to be implemented in the sample model, so that, it will be able to be decided which of them will be more appropriate.

The main idea was to build four pillows, one to five scale, implementing one Arduino and four temperature sensors and light meters to generate the inputs for the learning.



Iaso was decided to help in the building of this industrial model, showing real interest in the research. Nevertheless, the factory budget for this kind of research make in the end this objective

completely impossible. So, unluckily, without the factory help was not possible to test the material design and environmental performance during the length of the thesis research.

A different point of view was proposed then,

The realization of a completely virtual model simulating the reading of the sensors and light meters depending on the possible climates scenarios.

In this way it became possible to start simulating the learning and the performance of the patterns of the façade. A simulation of realistic inputs, that will allow, going ahead with the study of the proposed behaviour for learning, that in the future will be implemented in the system.

Considering point 3: Behavioural Learning Procedure

Cybernetics comes from the Greek form Plato used *kybernete*s.

Cybernetics as the study of Regulation Systems influenced the birth of Computer Science, ANN and AI in the 60's.

After the Shannon's "*Mathematical theory of Communications*" it was higly incremented the importance of "*Control + Communications*" study in Systems.

The study of AI Systems as systems with quasi-intelligent behaviour, introduced, the reactive research and bottom-up robotics as well as the study of evolutionary processes.

We started to simulate the emergent and generative properties of the nature processes, which use to result in well-adapted and efficient forms, starting to understand the difference, as Joslyn mentioned, between *intelligence-by-adapting* and *intelligence-by-reasoning*.

"*Systems that solve problems in surprising ways, in many cases structurally incompressible to programmers and resistant to reverse engineering*"
(see # 135, Joslyn, 2001)

The inventor of the Artificial Neuron, McCulloch opened the field of systems in which the properties of them were not dependant on the materials of the were constructed.

It was the birth of the abstract model of the system, and of the importance of the presence and absence of behaviours, not of the characteristics of those behaviours as before, starting a new world with the binary variables and the dependency between them.

Feed-forward and feedback systems with a not Newtonian Learning will be the base discussed in this thesis. How a stimulus-reaction behaviour on the EmDeplo Membrane can be improved enormously by making the membrane learn from its own behaviour and the environment.

Working with iteration or recursion and the self-organizing property of the system, *it will be demonstrated that, with a Goal State Simulation,* (which in nature is survival), *and the local goals of the system interaction, the membrane will definitely improve its insulation and sun-shading performance.*

A Hierarchical Controlled System.

A system that will work in two different scales:

[Global behaviour (t° control)] vs. [Local behaviour (shading & privacy)]

A local behaviour but also, a global behaviour from the rules of the individual: an emergent behaviour.

Patterns of behavioural adaptability without external control.

It will be learning to survive, will a goal of maintaining the proper temperature and shading, behaving like an organism that have to find out how to survive, and, trying to anticipate to what will happen.

"Autopoiesis is the condition necessary and sufficient for the organization of living systems."
(see # 108, Varela;, 1980) *page 8.*

It will be a natural generative, emergent, process with well-adapted and efficient forms, solving problems maybe in ways, in some cases, resistant to reverse engineering.

One of the first discussion and decisions to take will be about choosing a controlled System or an Autonomous System, thinking for example, if the learning should be externally supervised or not, and discussing about second order cybernetics, and, in Umpleby terms, the next step: the social cybernetics.

What kind of system should we design?

An ANN with an emergent behaviour?

A GA using natural selection to achieve the "[survival range of temperatures]?

Or an agent-based simulation?

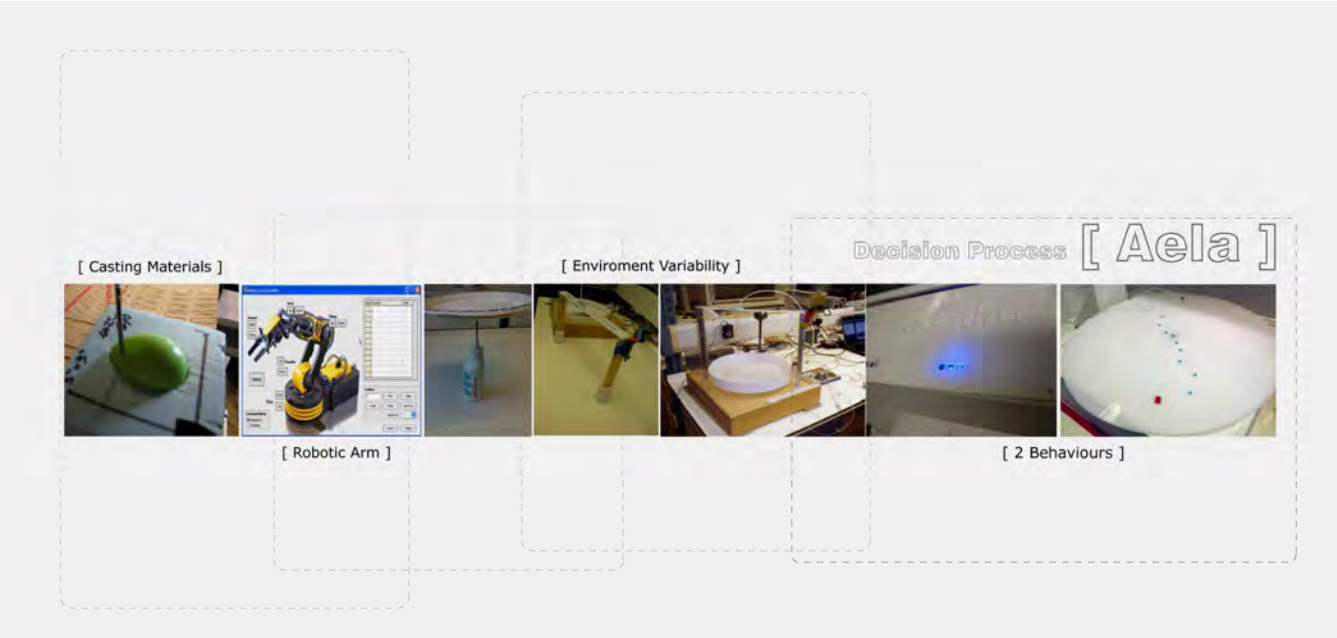
4. Initial experiments with Machine Laerning at the Bartlett.

[Aela] © Supervised Machine Learning within Variable Environment

Within a context of new digital fabrication techniques and new paths for AI implementation in Architectural Design, it is presented [_Aela] , a *Machine for digital fabrication with final product interaction.*

The main interest was to implement an Artificial Neural Network within a system for digital fabrication. The other main aim was to make the system to learn how to interact with its own previous creations.

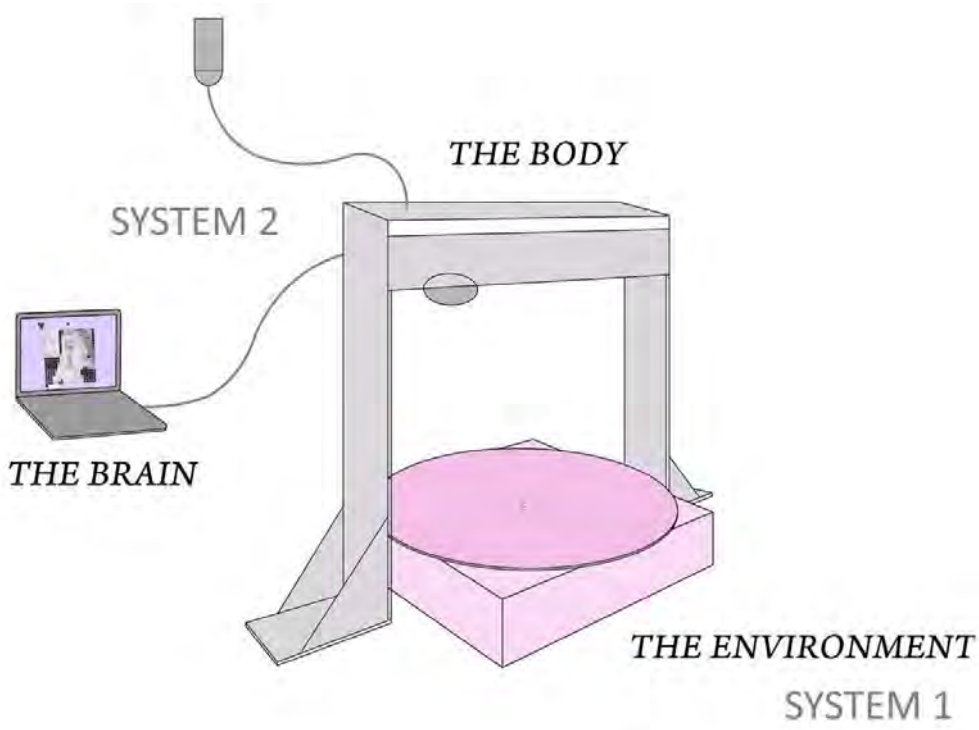
The idea of implementing algorithms for machine pattern recognition was also a basis for the code design. An important decision taken was to implement a supervised ANN and not an unsupervised one. Due to the kind of behaviour required for the machine, timing and procedure, an ANN was meant to be a much more efficient system for the research.



In this way, one of the first decisions was to implement a multilayer perceptron. As a basis for it, it will be used a set of 900 and 405 neurons for each of the behaviours studied.

[_Aela], through a series of observations used as Inputs for the ANN, learns and creates other observations, Outputs, which number and type depend on the behaviour studied.

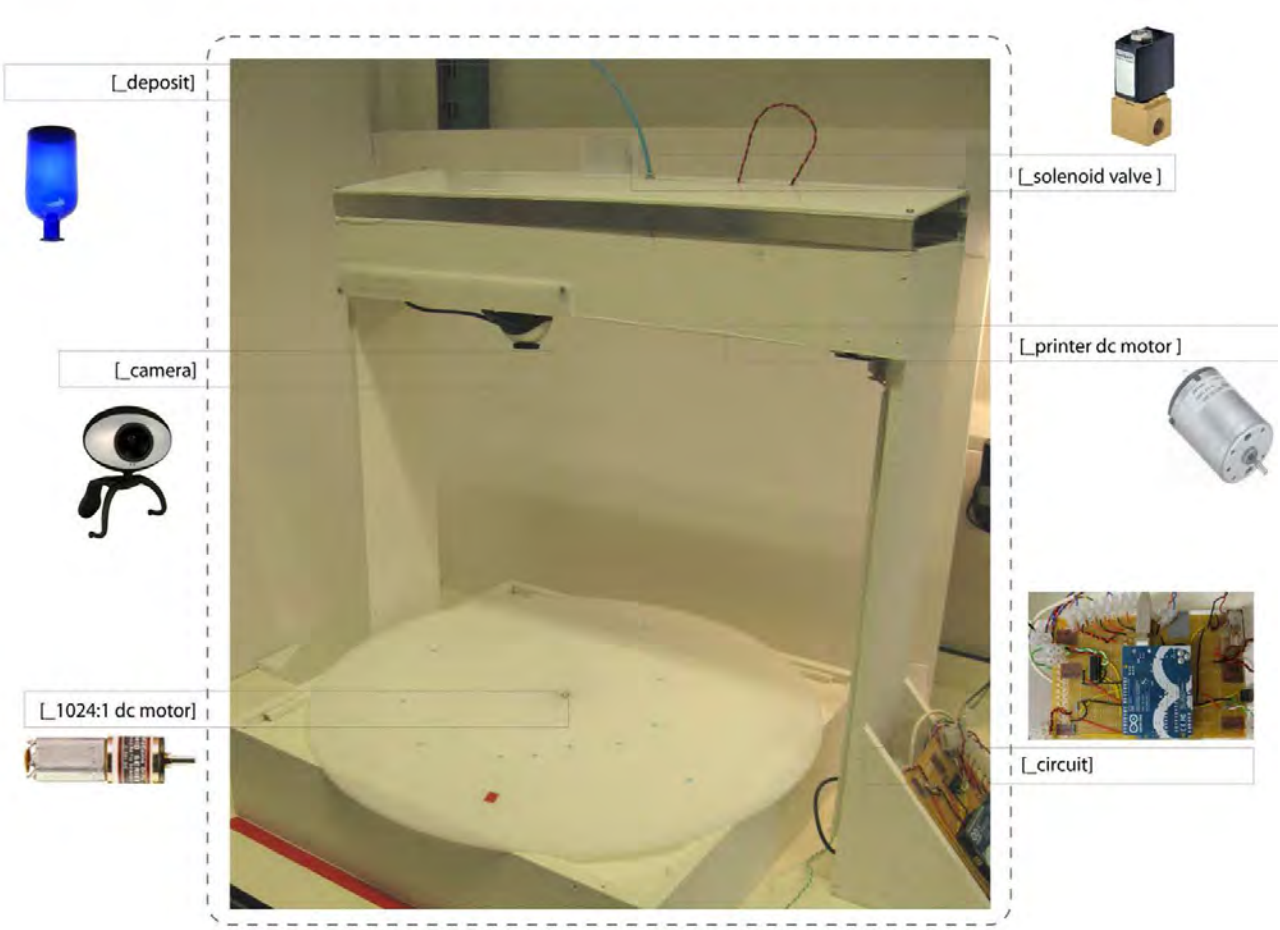
[_Aela] has a BRAIN, a BODY and an ENVIRONMENT.

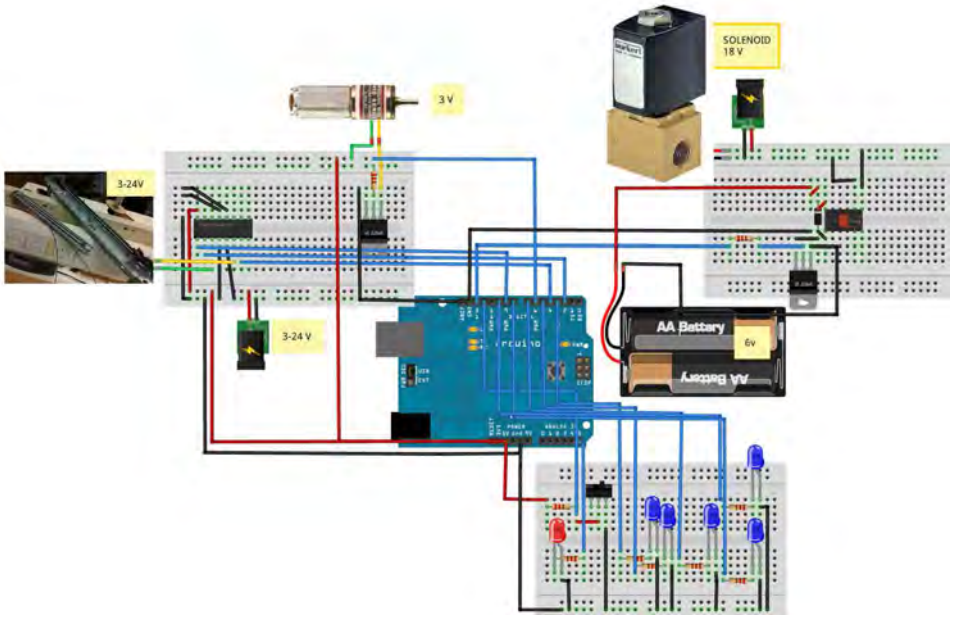


The environment is variable. The environment is changing without the machine knowing or being able to control it.

One of the main aims of the system will be to learn from the changing environment the parameters to make [_Aela] able to deal with it and get its aim.

The implementation of two behaviours was a definitive interesting step to be able to calibrate the errors of the script without the physical errors interfering.


[_deposit]
[_solenoid valve]
[_camera]
[_printer dc motor]
[_1024:1 dc motor]
[_circuit]

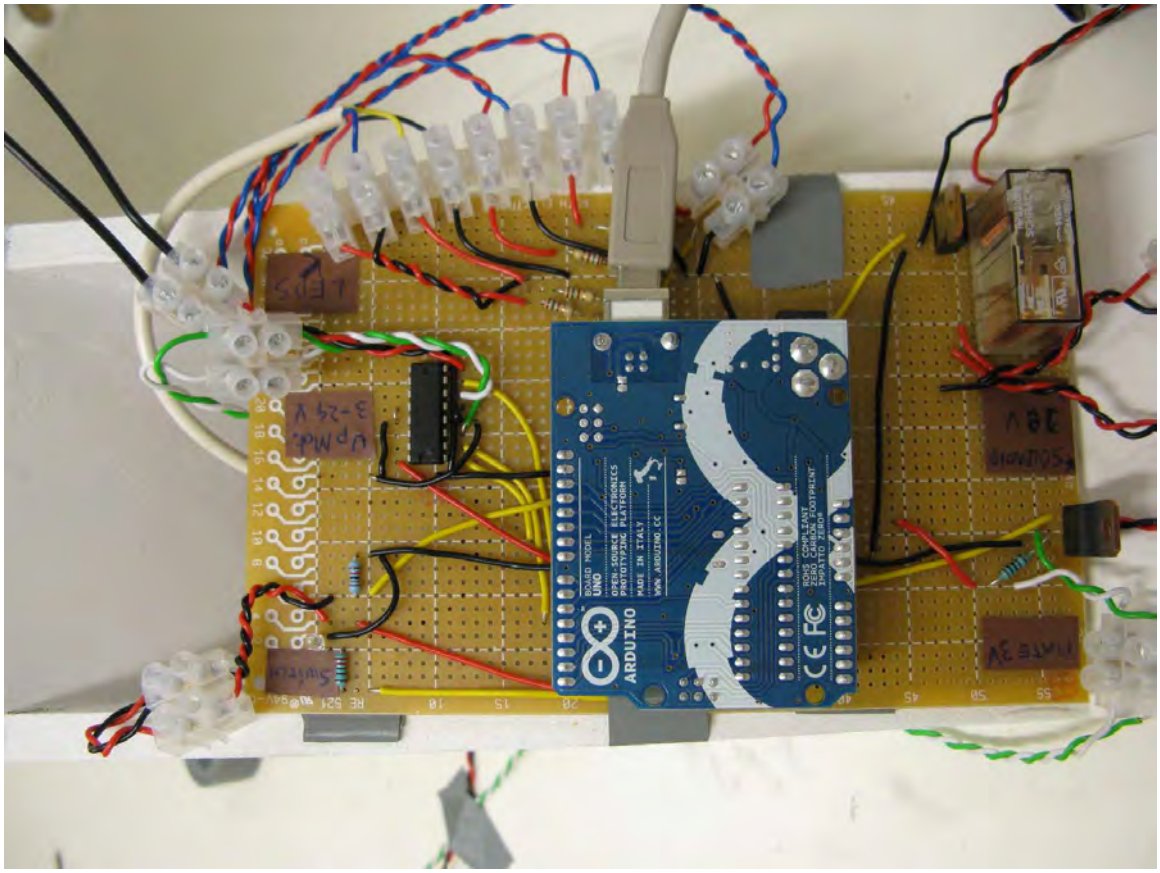

SOLENOID
18 V
3 V
3-24V
3-24 V
AA Battery
6v

Object Recognition vs. Dropping a line(s) on a rotating plate

[_Aela] learns from its own 2D product through a minimum number of trainings next to 15.

The system has learned how to recognise five different objects defined by their different 2D patterns and to draw a line within a variable environment which it is not able to control.

[__Aela]'s brain will be programmed in Java trough processing and Arduino microchip implemented.



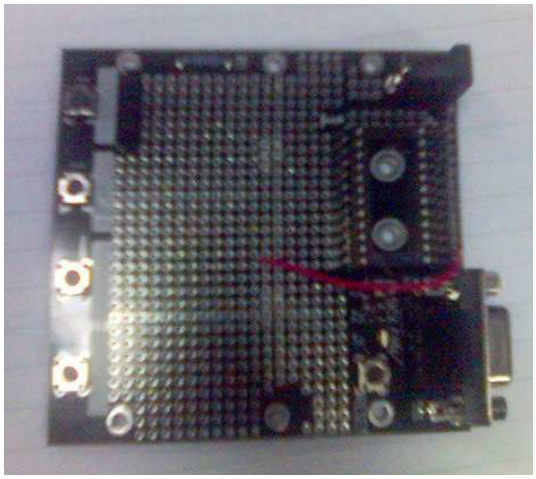
ARDUINO +PROCESSING IMPLEMENTATION

Arduino is a microprocessor for digital implementation of physical models designed for an easy collaboration with Processing-Java programming designs.

The board that will be used for the whole thesis is Arduino Uno, which has the characteristics shown below.

Microcontroller	ATmega368
Voltage	5V
Voltage input	7-12V
Voltage input limit	6-20V
Pins E/S digital	14
Pins analogical	6
pin	40 mA
pin 3.3V	50 mA
Flash memory	16 KB (ATmega168) OR 32 KB (ATmega328)
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)

Before choosing Arduino as the final chip used, several other were tried, as, for example the Basic Stamp 24, shown below. Nevertheless they were quite quickly discarded as its implementation was strongly more difficult and also the connexion between its software and Processing was very much complicated.

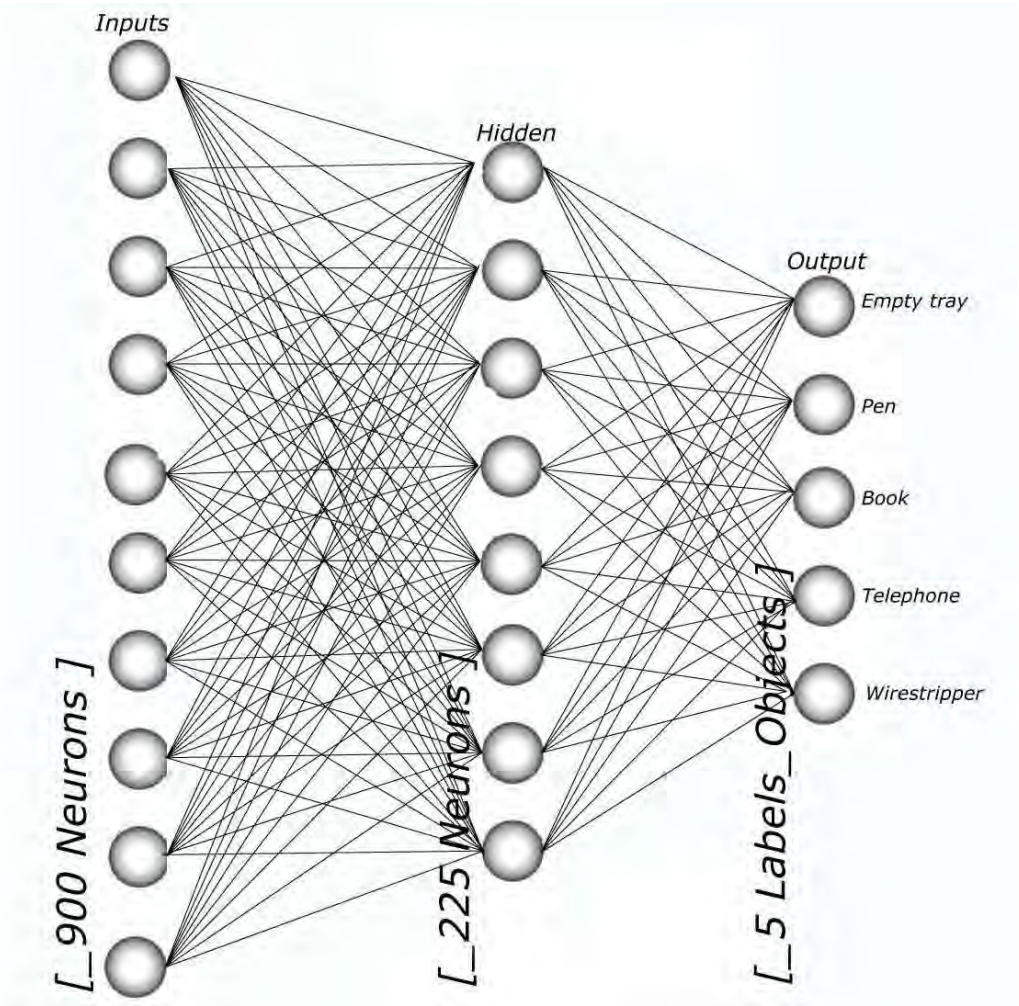


Behaviour 1 , [Object Recognition]

In the [behaviour 1], the machine recognises an environment in which patterns are changing through placing five different objects on the tray.

The output will be the correct answer to the question:

What object is the one on the tray?

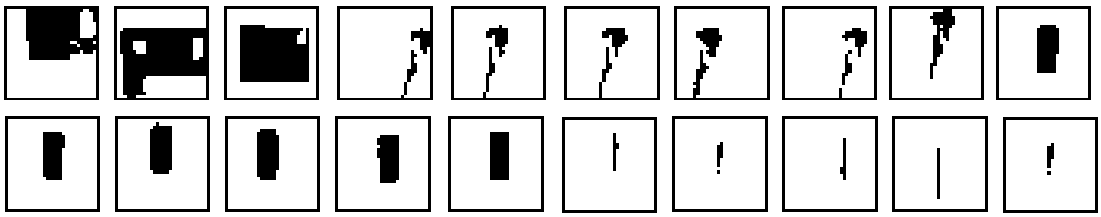


The system will be trying to distinguish between a pen, a hand, an empty tray , a book or a mobile phone. The training set will be the mappings as it is shown above of thousands of labelled pictures of the objects of 30x30 pixels definition.

So that, the input layer will be created will 900 neurons, one per each feature of a typical training set example.

[Training Set]

byte [] image= new byte[900] [Training Set]

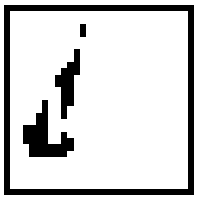


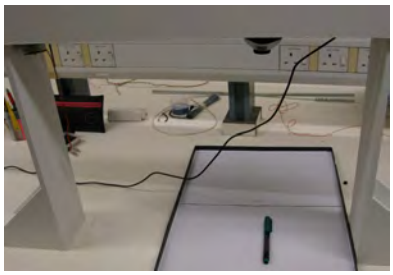
The testing set will be done, on the other hand, in real time. An object will be placed on the tray pictured and mapped by the camera, and finally classified with thr proper led indicator turning on.

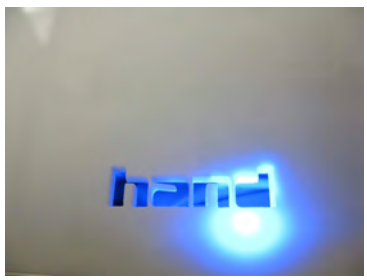
[Testing Set]

30 x 30 pix Camera pic

Map. byte [] image= new byte[900]







Behaviour 1 . Aela's brain, working code.

!. The ANN

```
int num_of_neurons=900;
int num_hidden_neurons=225;
int num_of_images=2000;
int num_of_outputs=5;
int imagesize=900;
int size_of_camera_pic=30;
int num_of_camera_pictures=1;
int label;
int train_pics=500;
import processing.video.*;
Capture cam;
import processing.serial.*;
Serial myPort;
Network neuralnet;
void setup()
{
size(30,30);
DATALOAD();
println("_DATA LOADED");
setupSigmoid();
neuralnet= newN etwork(num_of_neurons,num_hidden_neurons,num_of_outputs);
for(int i=0;i<train_pics;i++) {
int row= (int) floor(random(0,trainSet.length));
neuralnet.respond (trainSet[row].inputs);
neuralnet.train(trainSet[row].outputs);
}
println("_AELA FIRST TRAIN FINISHED");
String[] devices = Capture.list();
println(devices);
cam=new Capture(this,30,30,devices[2]);
String[]portList=Serial.list();
myPort=new Serial(this, portList[0], 9600);
}
void captureEvent(Capture cam) {
cam.read();
}
void draw()
{
image(cam,0,0);
//black and white
filter(THRESHOLD);
//keyPressed();
}
void mousePressed()
{
//set the testset float[].FROM THE CAMERA_____________________
if (mouseButton== LEFT) {
test_picture();
define_test_inputs(picture_arra);
println("_TEST INPUTS LOADED");
data_test();
}
}
void keyPressed() {
if(keyPressed) {
for(int j=0;j<5;j++) {
for(int i=0;i<train_pics;i++) {
```

```
int row= (int) floor(random(0,trainSet.length));
neuralnet.respond (trainSet[row].inputs);
neuralnet.train(trainSet[row].outputs);
}
}
}
println("_NEW 10 TRAININGs FINISHED");
  }
```

```
int imagesArrayLenght=(num_of_images*imagesize);
int []labels= new int[num_of_images];
byte[]images;
byte [] arrA;
java.io.File folder = new
java.io.File(dataPath("C:/users/usuario/Desktop/aela_//filter (returns true if
file's extension is .bmp)READ THE TRAIN IMAGES
java.io.FilenameFilter bmpFilter = new java.io.FilenameFilter() {
boolean accept(File dir, String name) {
return name.toLowerCase().endsWith("t.bmp");
}
};
class Data//_____________________________________________________________
{
float [] inputs;
float [] outputs;
int label_output;
Data()
{
inputs=new float [num_of_neurons];
outputs= new float[num_of_outputs];
}
//___load the images & create the inputs[] OF FLOATS___________________
void LoadImages(byte [] images, int imagesize) {
for(int i=0; i<num_of_neurons; i++) {
inputs[i]= int (images[i+imagesize]);//128.0-1.0;
}
}
//___load the labels & create the outputs [] OF FLOATS_________________
//there are 10 different labels but we scale the outputs to yes & no
void LoadLabels (int [] labels, int offset)
{
label_output= int(labels[offset]);
for (int i=0; i<5;i++) {
if(i==label_output) {
outputs[i]=1.0;
}
else {
outputs[i]=-1.0;
}
}
}
}
//end of the class_____________________________________________________
//FUNCTION TO LOAD THE DATA(we call it in the main tab)_________________
Data [] trainSet;
void DATALOAD() {
//create the images byte []_____________________________
PImage img;
size(size_of_camera_pic,size_of_camera_pic);
String[] filenames = folder.list(bmpFilter);
for (int i = 0; i < filenames.length; i++) {
println(filenames[i]);
}
```

```
for(int i=0; i<num_of_images;i++) {
img=loadImage(filenames[i]);
image(img, 0, 0);
img.loadPixels();
arrA = new byte[img.pixels.length];
images = new byte[int(img.pixels.length*num_of_images)];
for( int k = 0; k<img.pixels.length; ++k) {
arrA[k] = intToByteA(img.pixels[k]);
//BIG DATA ARRAY
images[(arrA.length*i)+k]=arrA[k];
}
}
//____loalda bels_____________________________________________________
String[] bits = loadStrings("labels.txt");
for (int i=0; i<10; i++) {
labels[i]=int(bits[i]);
}
println("_LABELS LOADED");
//set the trainSet float[]_____________________________________
trainSet=new Data [num_of_images];
int trainPos=0;
for(int i=0;i<num_of_images;i++) {
trainSet [trainPos]= new Data();
trainSet [trainPos].LoadImages(images, i*imagesize);
trainSet [trainPos].LoadLabels(labels, i);
trainPos++;
}
}
//END OF LOAD DATALOAD FUNCTION_____________________________________________
//to CREATE the ARRAY of IMAGES for TRAINNING SET_________
byte intToByteA(int x) {
return (byte) (x & 0xFF);
}
  //_____________________________________________________________


class Network
{
Neuron [] m_input_layer;
Neuron [] m_hidden_layer;
Neuron [] m_output_layer;
Network(int inputs, int hidden, int outputs)
{
m_input_layer = new Neuron [inputs];
m_hidden_layer = new Neuron [hidden];
m_output_layer = new Neuron [outputs];
// set up the network topology
for (int i = 0; i < m_input_layer.length; i++) {
m_input_layer[i] = new Neuron();
}
// route the input layer to the hidden layer
for (int j = 0; j < m_hidden_layer.length; j++) {
m_hidden_layer[j] = new Neuron(m_input_layer);
}
// route the hidden layer to the output layer
for (int k = 0; k < m_output_layer.length; k++) {
m_output_layer[k] = new Neuron(m_hidden_layer);
}
}
int respond(float [] inputs)// for TRAINNING
{
float [] responses = new float [m_output_layer.length];
for (int i = 0; i < m_input_layer.length; i++) {
m_input_layer[i].m_output = inputs[i];
}
for (int j = 0; j < m_hidden_layer.length; j++) {
```

```
m_hidden_layer[j].respond();
}
// and finally feed forward to the output layer
for (int k = 0; k < m_output_layer.length; k++) {
responses[k] = m_output_layer[k].respond();
}
// now check the best response:
int response = -1;
float best = max(responses);
for (int a = 0; a < responses.length; a++) {
if (responses[a] == best) {
response = a;
}
}
return response;
}
void train(float [] outputs)
{
// adjust the output layer
for (int k = 0; k < m_output_layer.length; k++) {
m_output_layer[k].finderror(outputs[k]);
m_output_layer[k].train();
}
// propagate back to the hidden layer
for (int j = 0; j < m_hidden_layer.length; j++) {
m_hidden_layer[j].train();
}
}
}
  //end of network class_____________________________________________


float LEARNING_RATE = 0.01;
class Neuron
{
Neuron [] m_inputs;
float [] m_weights;
float m_threshold;
float m_output;
float m_error;
// the input layer of neurons have no inputs:
Neuron()
{
m_threshold = 0.0;
m_error = 0.0;
// initial random output
m_output = lookupSigmoid(random(-5.0,5.0));
}
// all other layers (hidden and output) have
// neural inputs
Neuron(Neuron [] inputs)
{
m_inputs = new Neuron [inputs.length];
m_weights = new float [inputs.length];
for (int i = 0; i < inputs.length; i++) {
m_inputs[i] = inputs[i];
m_weights[i] = random(-1.0,1.0);
}
m_threshold = random(-1.0,1.0);
m_error = 0.0;
// initial random output
m_output = lookupSigmoid(random(-5.0,5.0));
}
// respond looks at the layer below, and prepares a response:
float respond()
{
```

```
float input = 0.0;
for (int i = 0; i < m_inputs.length; i++) {
input += m_inputs[i].m_output * m_weights[i];
}
m_output = lookupSigmoid(input + m_threshold);
// reset our error value ready for training
m_error = 0.0;
return m_output;
}
// find error is used on the output neurons
void finderror(float desired)
{
m_error = desired - m_output;
}
// train adjusts the weights by comparing actual output to correct outpu
void train()
{
float delta = (1.0 - m_output) * (1.0 + m_output) * m_error * LEARNING
for (int i = 0; i < m_inputs.length; i++) {
// tell the next layer down what it's doing wrong
m_inputs[i].m_error += m_weights[i] * m_error;
// correct our weights
m_weights[i] += m_inputs[i].m_output * delta;
}
}
  }
```

```
PImage cam_image_BW;
byte[]picture_arra;
void test_picture() {
println("_CAMERA READY");
frameRate(10);
saveFrame("#PIC.bmp");
cam_image_BW=loadImage("#PIC.bmp");
image(cam_image_BW, 0, 0);
cam_image_BW.loadPixels();
picture_arra = new byte[cam_image_BW.pixels.length];
for( int k = 0; k<cam_image_BW.pixels.length; ++k) {
picture_arra[k] = intToByteTESTpic(cam_image_BW.pixels[k]);
}
println("_OBJECT LOADED");
}
void define_test_inputs(byte [] picture_arra) {
float [] inputs;
inputs=new float [num_of_neurons];
for (int i = 0; i < num_of_neurons; i++) {
inputs[i] = int(picture_arra[i]);// / 128.0 - 1.0;
}
}
void data_test() {
Data [] testSet;
testSet=new Data [num_of_camera_pictures];
testSet [0]= new Data();
testSet [0].LoadImages(picture_arra, 0);
int response = -1;
response = neuralnet.respond(testSet[0].inputs);
//text(str(response),10,10);
println("THIS OBJECT IS LABEL "+response);
if (response==0) {
myPort.write('B');
println("THIS IS A BOOK");
}
if (response==1) {
myPort.write('H');
println("THIS IS A WIRESTRIPPER");
```

```
}
if (response==2) {
myPort.write('T');
println("THIS IS A TELEPHONE");
}
if (response==3) {
myPort.write('P');
println("THIS IS PEN");
}
if (response==4) {
myPort.write('V');
println("THE TRAY IS EMPTY");
}
}
byte intToByteTESTpic(int a) {
return (byte) (a & 0xFF);
 }
```

```
float [] g_sigmoid = new float [200];
void setupSigmoid(){
for (int i = 0; i < 200; i++) {
float x = (i / 20.0) - 5.0;
g_sigmoid[i] = 2.0 / (1.0 + exp(-2.0 * x)) - 1.0;
}
}
float lookupSigmoid(float x){
return g_sigmoid[constrain((int) floor((x + 5.0) * 20.0),0,199)];
 }
```

2. ARDUINO CODE

```
char val; // Data received from the serial port
void setup()
{
Serial.begin(9600); // Start serial communication at 9600 bps
pinMode(10, OUTPUT);
pinMode(8, OUTPUT);
pinMode(3, OUTPUT);
pinMode(5, OUTPUT);
pinMode(7, OUTPUT);
}
void loop()
{
if (Serial.available()) { // If data is available to read,
val = Serial.read(); // read it and store it in val
}
if (val == 'B') { // If BOOK was received
digitalWrite(10, HIGH); / / set the LED on
delay(3000); // wait for a second
digitalWrite(10, LOW); / / s et the LED off
}
if (val == 'H') {
digitalWrite(3, HIGH);
delay(3000);
digitalWrite(3, LOW);
}
if (val == 'T') {
digitalWrite(5, HIGH);
delay(3000);
```

```
digitalWrite(5, LOW);
}
if (val == 'P') {
digitalWrite(8, HIGH);
delay(3000);
digitalWrite(8, LOW);
}
if (val == 'V') {
digitalWrite(7, HIGH);
delay(3000);
digitalWrite(7, LOW);
}
  }
```

Behaviour 2 [Dropping line(s) on a Rotating Plate]

In the [behaviour 2] the machine will try to draw, through liquid material dropping, a line on the rotating disk, that the machine does not know that is rotating.

I was since the very beginning interested in a 3D building and mapping for the machine to deal with, nevertheless, the difficulties of 3D mapping for implementation in this kind of training set, made the ANN not achievable in terms of schedule and coding procedures.

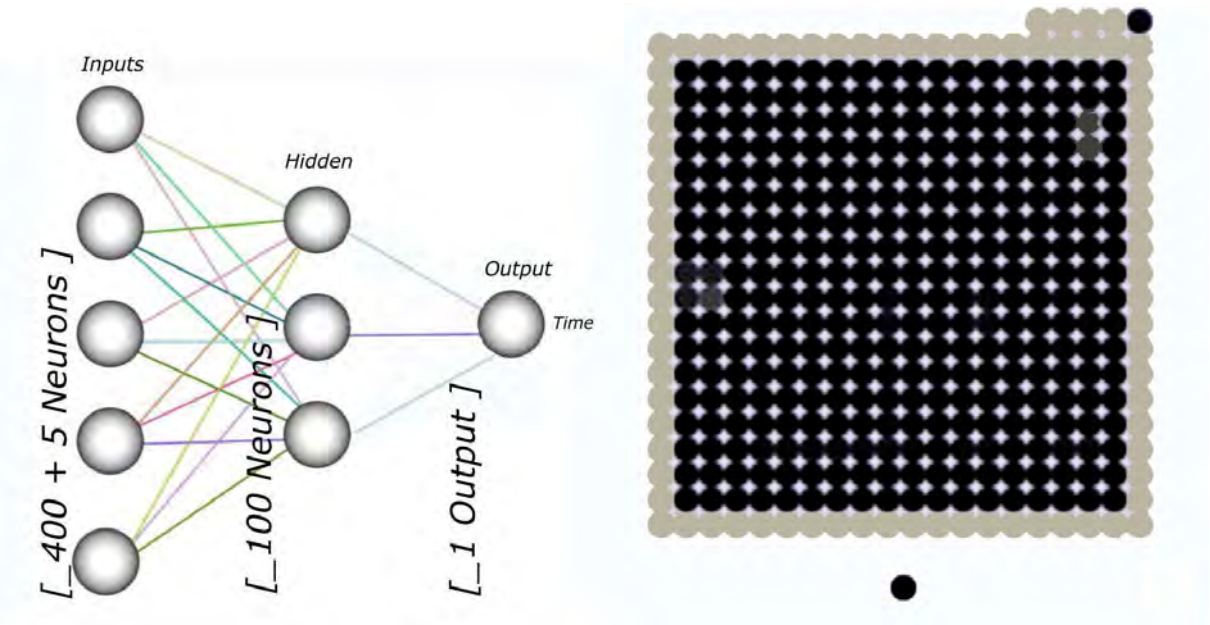
By using a dropping material [_Aela] will try to learn from the environment and work with its parameters. The machine will have to understand that the drop is moving somehow as, when checked with the camera, [__Aela] will find out that the drop is not at the same place it dropped it for an unknown reason.

Once this learning is done it allows the system to reach its goal:

to draw a line on a moving environment.

Output will be to learn the appropriate dropping time between 0-4 sec for drawing a straight line.

The use of liquid dropping will allow it to implement an infinite number of iterations. ANN output will be the proper timing for the machine to drop in order to reach the draw of a line.



The system will have 5 extra neurons for timing.

Process

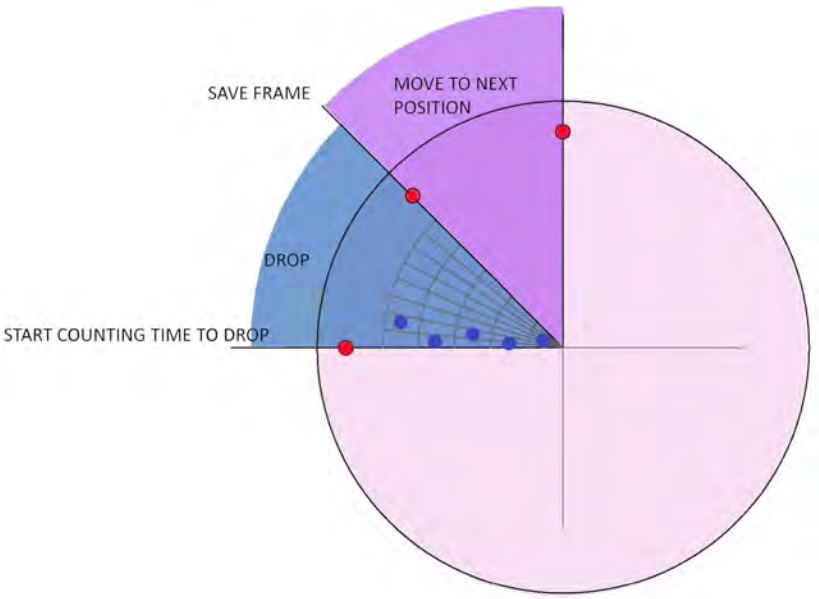
- The disc is rotating with a constant speed and four different dropping positions have been previously defined. (p1, p2, p3, p4)

- When the camera sees the red dot, [_Aela] waits a time within 0-4 seconds,(t0, t1, t2, t3, t4), and drops.

- Right after that, the camera captures the frame.

- While moving to the next position, [_Aela] processes the information and decides the right time for the next drop in order to draw a straight line.

- The process is repeated 4 times until a dotted line is drawn.

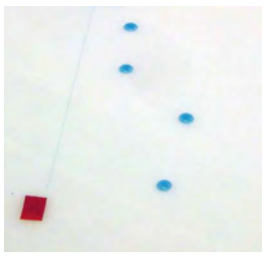


For this second behaviour the training set will be created with pictures of different configurations drawn on the rotating disc adequately labelled.

[Training Set] saved images printer_pos+time_drop.bmp

As in the behaviour number one the testing will be done in real time when the machine drops every drop through camera recognition.

[Testing Set] camera pic 20 x 20 pixels



Behaviour 2 . Aela's brain, working code.

1. The ANN script

```
import processing.video.*;
Capture cam;
int counter=0;
Network neuralnet;
void setup()
{
size (200,200);
loadData();
setupSigmoid();
neuralnet= new Network(405,100);
cam = new Capture(this, 20,20, 4);
// String[] devices = Capture.list();
// println(devices);
// cam = new Capture(this, 20, 20, devices[1]);
cam = new Capture(this, 20, 20);
}
void draw()
{
background(220,204,255);
noStroke();
smooth();
neuralnet.draw();
if(cam.available()) {
cam.read();
}
image(cam, 6,6);
// filter(THRESHOLD);
loadCamera();
}
void mousePressed()
{
if (mouseButton== LEFT) {
for(int i=0; i<10; i++) {
neuralnet.respond (training_set[i]);
neuralnet.train(training_set[i].output);
}
}
else if (mouseButton== RIGHT) {
testing_frame.imageLoad(camBytes);
neuralnet.respond (testing_frame);
saveFrame("####.jpg");
}
redraw();
  }
  }
  ----------------------------------------------------------------
```

```
byte[] camBytes;
class Camera
{
float [] inputs;
Camera()
{
inputs=new float [400];
}
void imageLoad(byte [] camBytes)
{
for(int i=0; i<400; i++)
{
inputs[i]= int (camBytes[i])/128.0-1.0;
}
}
}
Camera testing_frame;
void loadCamera()
{
cam.loadPixels();
camBytes= new byte[cam.pixels.length];
for( int i = 0; i<cam.pixels.length; i++) {
camBytes[i] = intToByteA(cam.pixels[i]);
}
testing_frame=new Camera();
testing_frame.imageLoad(camBytes);
  }
```

```
int imgNum=10;
byte[]images;
int []labels= new int[imgNum];
java.io.File folder = new
java.io.File(dataPath("C:/Users/Elena/Desktop/AELA_java.io.FilenameFilter
bmpFilter = new java.io.FilenameFilter() {
boolean accept(File dir, String name) {
return name.toLowerCase().endsWith(".bmp");
}
};
class Card
{
float [] inputs;
float output;
int label;
Card()
{
inputs=new float [400];
}
//.........................
void imageLoad(byte [] images, int offset)
{
for(int i=0; i<400; i++)
{
inputs[i]= int (images[i+offset])/128.0-1.0;
}
}
//.........................
void labelLoad (int [] labels, int offset)
{
label= labels[offset];
output=label/3; / / A N D divide with number eg.to make range cl
}
}
//.....................................end of Card class
Card [] training_set;
void loadData()
```

```
{
//load labels.....................
String[] bits = loadStrings("labels.txt");
for (int i=0; i<imgNum; i++) {
labels[i]=int(bits[i]);
}
//println(labels);
//load images.....................
PImage img;
String[] filenames = folder.list(bmpFilter);
for(int i=0; i<imgNum;i++) {
img=loadImage(filenames[i]);
image(img, 0, 0);
img.loadPixels();
byte [] arrA = new byte[img.pixels.length];
images = new byte[int(img.pixels.length*imgNum)];
for( int k = 0; k<img.pixels.length; ++k) {
arrA[k] = intToByteA(img.pixels[k]);
images[(arrA.length*i)+k]=arrA[k];
}
}
// println(images);
//.................................
training_set=new Card [imgNum];
int tr_pos=0;
for(int i=0;i<imgNum;i++)
{
training_set [tr_pos]= new Card();
training_set [tr_pos].imageLoad(images,i*400);
training_set [tr_pos].labelLoad(labels,i);
tr_pos++;
}
}
byte intToByteA(int x) {
return (byte) (x & 0xFF);
 }
```

```
class Network
{
Neuron [] m_input_layer;
Neuron [] m_hidden_layer;
Neuron m_output_layer;
Network (int inputs, int hidden)
{
m_input_layer=new Neuron [inputs];
m_hidden_layer=new Neuron [hidden];
for (int i=0; i<m_input_layer.length; i++) {
m_input_layer [i] =new Neuron();
}
for (int j=0; j<m_hidden_layer.length; j++) {
m_hidden_layer [j] =new Neuron(m_input_layer);
}
m_output_layer=new Neuron (m_hidden_layer);
}
void respond(Card card)
{
for (int i=0; i<m_input_layer.length; i++) {
if (i<400)
{
m_input_layer[i].m_output = card.inputs[i];
}
else if (i>=400)
{
```

```
if (i-400==counter) {
m_input_layer[i].m_output=1.0;
}
else {
m_input_layer[i].m_output=-1.0;
}
}
}
for (int j=0; j<m_hidden_layer.length; j++) {
m_hidden_layer[j].respond();
}
m_output_layer.respond();
}
void respond(Camera testing_frame)
{
for (int i=0; i<m_input_layer.length; i++) {
if (i<400)
{
m_input_layer[i].m_output = testing_frame.inputs[i];
}
else if (i>=400)
{
if (i-400==counter) {
m_input_layer[i].m_output=1.0;
}
else {
m_input_layer[i].m_output=-1.0;
}
}
}
for (int j=0; j<m_hidden_layer.length; j++) {
m_hidden_layer[j].respond();
}
m_output_layer.respond();
}
void draw ()
{
for (int i=0; i< m_input_layer.length; i++) {
pushMatrix();
translate((i%20) * width / 25.0 + width * 0.125, (i/20) * height /
m_input_layer[i].display();
popMatrix();
}
pushMatrix();
translate (width/2, height*0.075);
m_output_layer.display();
popMatrix();
}
void train (float output)
{
m_output_layer.setError(output);
m_output_layer.train();
for (int j=0; j< m_hidden_layer.length; j++) {
m_hidden_layer[j].train();
}
}
 }
```

```
float LEARNING_RATE=0.01;
class Neuron
{
Neuron [] m_inputs;
float [] m_weights;
```

```
float m_output;
float m_error;
Neuron () / / n e u r o n s f o r input layer: pixels+ positions
{
}
Neuron (Neuron [] inputs) / / neurons for hidden and output layers
{
m_inputs=new Neuron [inputs.length];
m_weights= new float [inputs.length];
for (int i=0; i<inputs.length; i++) {
m_inputs[i]=inputs[i];
m_weights[i]=random(-1.0,1.0);
}
}
void respond ()
{
float input=0.0;
for (int i=0; i<m_inputs.length; i++) {
input+= m_inputs[i].m_output * m_weights[i];
}
m_output=lookupSigmoid(input);
m_error=0.0;
}
void setError(float desired)
{
m_error= desired- m_output;
}
void train()
{
float delta= (1.0 - m_output) * (1.0 + m_output)* m_error* LEARNING_RA
for (int i=0; i< m_inputs.length; i++) {
m_inputs[i].m_error += m_weights[i]* m_error;
m_weights[i]+= m_inputs[i].m_output* delta;
}
}
void display()
{
fill(128*(1-m_output));
ellipse(0,0,8,8);
}
  }
```

2. ARDUINO CODE

```
int LED = 19; / / t h i s t r a n s lates to using analog pin 5 as a digital
#define SWITCH 12 / / p i n f o r switch
int switchVal=0;
int motor1 = 2; / / p i n f o r the motor going forward
int motor2 = 4; / / p i n f o r the motor going backward
int motorpmw = 9; / / + / - b a ttery power the motor is getting (speed)
int plateMotor=6; / / p l a t e motor
void setup() {
Serial.begin(9600);
// initialize the digital pins as an output
pinMode(SWITCH, INPUT);
pinMode(LED, OUTPUT);
pinMode(motor1, OUTPUT); / / p i n for the motor going in
pinMode(motor2, OUTPUT); / / p i n f or the motor going out
pinMode(motorpmw, OUTPUT); / / + /- battery power the motor is getting (sp
pinMode(plateMotor, OUTPUT); / / p in for plate motor
pinMode(13, OUTPUT); / / p i n f o r s olenoid valve
```

```
// digitalWrite(plateMotor, HIGH);// plate starts moving
digitalWrite(13, HIGH);
while(digitalRead(SWITCH)==LOW){ // to set up printer head. go out until
out();
}
in();
delay(100);
freeze();
Serial.print(88,BYTE); //sending 'X'
}
void loop() {
if (switchVal==HIGH){
Serial.print("swt");
in();
delay(100);
freeze();
}
serialStuff();
}
void serialStuff(){
if(Serial.available()>0) {
int val=Serial.read();
if(val=='Y')
{
drop();
}
if(val=='Z')
{
oneStep();
}
if(val=='W')
{
out();
}
//.....................extra
if(val=='R') // start plate rotation
{
digitalWrite(plateMotor, HIGH);
}
if(val=='S') // stop plate rotation
{
digitalWrite(plateMotor, LOW);
}
if(val==',') // move out and stop
{
out();
delay(200);
freeze();
}
if(val=='.') // move in and stop
{
in();
delay(200);
freeze();
}
//.................................
}
}
void oneStep(){
in();
delay(350);
freeze();
}
void drop(){
digitalWrite(13, LOW);
delay(70);
```

```
digitalWrite(13, HIGH);
}
void in(){
analogWrite(motorpmw, 255);
digitalWrite(motor1, HIGH);
digitalWrite(motor2, LOW);
}
void out(){
analogWrite(motorpmw, 255);
digitalWrite(motor1, LOW);
digitalWrite(motor2, HIGH);
}
void freeze(){
analogWrite(motorpmw, 0);
  }
```

5. Implementing Machine Learning @ EmDEplo

5.1 The stimulus-reaction model.

The first software model done wil be the stimulus-reaction model. After the analysis done of the Media-Tic building it was considered necessary to model the current proposal performance for being able to compare afterwards its behaviour with the one of the system implemented in EmDep Morphogenesis.

My main reason for my opposition to this kind of performance is based in the adaptability/defense contradiction.

As the bases of a well adapt system there are two concepts. The first one is understanding this adaptability as a reaction to changes in the environment. The second one is the timing.

Stimulus-reaction behaviour is just a defense. It is just an on/off button that turns on the shield against the environment and the weather.

I understand adaptability with a learning implemented in the timing of the response. As a learned behaviour that will improve with time and, that will make the building, to act before the situation is already affecting badly the organism (the building). So that, timing is essential.

A stimulus-reaction performance can maybe be implemented for a local behaviour of creating privacy or desired local shadow in the futures EmDeplo's brain. Nevertheless, the breeding of the building will be adapting to the environment, improving more and more as time passes by.

For simulating the stimulus reaction behaviour, (also the media-tic performance), the model will work with a set of random temperatures within a range that is consider to be possible in its scenario.

The sensors will define when the pillows of the façade will open, and the light meters, as implemented in the media-tic, will open or close the shaded layer for generating shadow inside the building.

In this case we will work with a summer scenario range of temperatures and with a façade of 100 ETFE pillows, which is approximately the size of the media-tic façade.

It will be simulated the behaviour of 100 Arduino chips, one per pillow and 100 light meters, as they are located in the real building.

The system will be composed by:

```
100 pillows
100 Arduinos
T° inputs = new float [100];
T° [i]= random(25, 35);

Sun_shade=new int [100];
Sun_shade [i]=random (-1, 2)
```

It will be designed a very simple visible layout of the performance so it can be appreciatad in an easier way without having to study deeply the code in each iteraction to evaluate the results.

The visual layout will emule the 100 ETFE pillows, grey when opened, white when closed, and the sun-shadind layer represented by a patterned grey set of circles that appears when open.

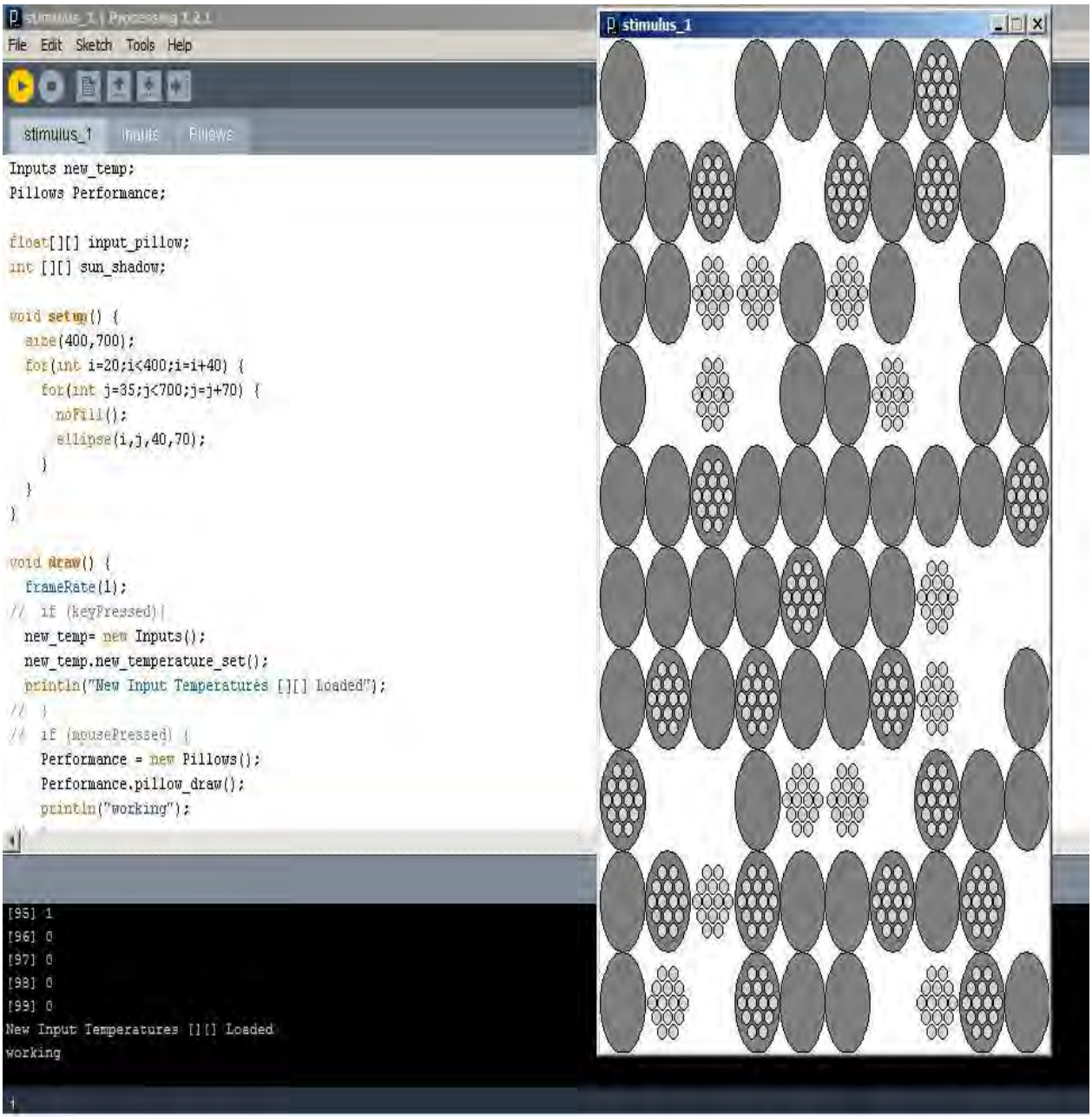


Figure 18

The sensors checking time will be supposed to work every 30 minutes to implement a similar timing as in the EmDep System. The two inputs arrays will simulate the readings of the sensors and light meters of the façade.

The main problem that is shown for this system configuration is the effectiveness of the temperature management.

There is no efficacy in no preventing the high increase of interior temperatures in advance. When the building acts, the temperatures inside are already quite high, eliminating the possibility of a better climate magnagement. Stimulus-reaction behviour has not shown a real environmental adaptability, not more than the one that can be achieved with a manually activated brise-soleil.

The Stimulus- Reaction model

```
Inputs new_temp;
Pillows Performance;
float[][] input_pillow;
int [][] sun_shadow;
void setup() {
      size(400,700);
      for(int i=20;i<400;i=i+40) {
            for(int j=35;j<700;j=j+70) {
                  noFill();
                  ellipse(i,j,40,70);
            }
      }
}
void draw() {
      frameRate(1);
// if (keyPressed){
      new_temp= new Inputs();
      new_temp.new_temperature_set();
      println("New Input Temperatures [][] Loaded");
// }
```

```
// if (mousePressed) {
      Performance = new Pillows();
      Performance.pillow_draw();
      println("working");
// }
}
```

```
int half_wide=20;
int x_coord;
int half_heigh=35;
int y_coord;

class Pillows {
      Pillows() {
      }
      void pillow_draw() {
            background(255);
            for(int i=0; i<10;i++) {
                  for(int j=0;j<10;j++) {
                        float T= input_pillow[i][j];
                        int sun_shade=sun_shadow[i][j];
                        if(T<19||T>25) {
                                    //open
                              x_coord=half_wide+(40*i);
                              y_coord=half_heigh+(70*j);
                              fill(126);
                              ellipse(x_coord,y_coord,40,70);
                        }
                        if(sun_shade>0) {
                              //open shade pattern
                              x_coord=half_wide+(40*i);
                              y_coord=half_heigh+(70*j);
                              fill(209);
                              ellipse(x_coord-5,y_coord,8,10);
                              ellipse(x_coord+5,y_coord,8,10);
```

```
                        ellipse(x_coord-14,y_coord,8,10);
                        ellipse(x_coord+14,y_coord,8,10);
                        ellipse(x_coord,y_coord+10,8,10);
                        ellipse(x_coord,y_coord-10,8,10);
                        ellipse(x_coord-10y_coord-10,8,10);
                        ellipse(x_coord+10,y_coord-10,8,10);
                    ellipse(x_coord-10,y_coord+10,8,10);
                    ellipse(x_coord+10,y_coord+10,8,10);
                    ellipse(x_coord-5,y_coord+20,8,10);
                    ellipse(x_coord+5,y_coord+20,8,10);
                    ellipse(x_coord-5,y_coord-20,8,10);
                    ellipse(x_coord+5,y_coord-20,8,10);
                }
            }
        }
    }
}
```

```
float [] Temperatures;
int [] random_sun_shade;

class Inputs{
      Inputs()   {
      }
      void new_temperature_set() {
            Temperatures= new float [100];
            random_sun_shade=new int [100];
            for(int i=0;i<100;i++) {
                  float Pillow_input_T=random(20,35);
                  Temperatures[i]=Pillow_input_T;
                  random_sun_shade[i]=int (random(-1,2));
            }
            println(Temperatures);
            println("New Set of Temperatures Created");
            println(random_sun_shade);
```

```
            input_pillow=new float [10][10];
            sun_shadow=new int [10][10];
            for(int i=0; i<10;i++) {
                  for(int j=0;j<10;j++) {
                  input_pillow[i][j]=Temperatures[i+(j*10)];
                  //Gives an input temperature
                  sun_shadow[i][j]=random_sun_shade[i+(j*10)];
                  }
            }
      }
}
```

5.2 Artificial Neural Network & Genetic Algorithm model.

When the study of the "intelligent" membrane appeared as a real option for efficient performance, the first question was obvious:

How should we make it intelligent?

The process is proposed in two steps:

1. To define the way of performing once the decision about what climate we are in is taken.
2. To be able to recognise after training and learning in which situation of T°, radiation and sun-shade we are in.
3. To define the dynamic process of the membrane once the training has been done and the decision about in which T°/sun-shade situation we are currently in.

Extra parameters to decide.

- System type of supervision. Autonomous or not?
- timing for checking real outside building situation.
- timing for checking inside situation.
- training process and dynamic learning timing.

Evolutionary computation for EmDeplo.

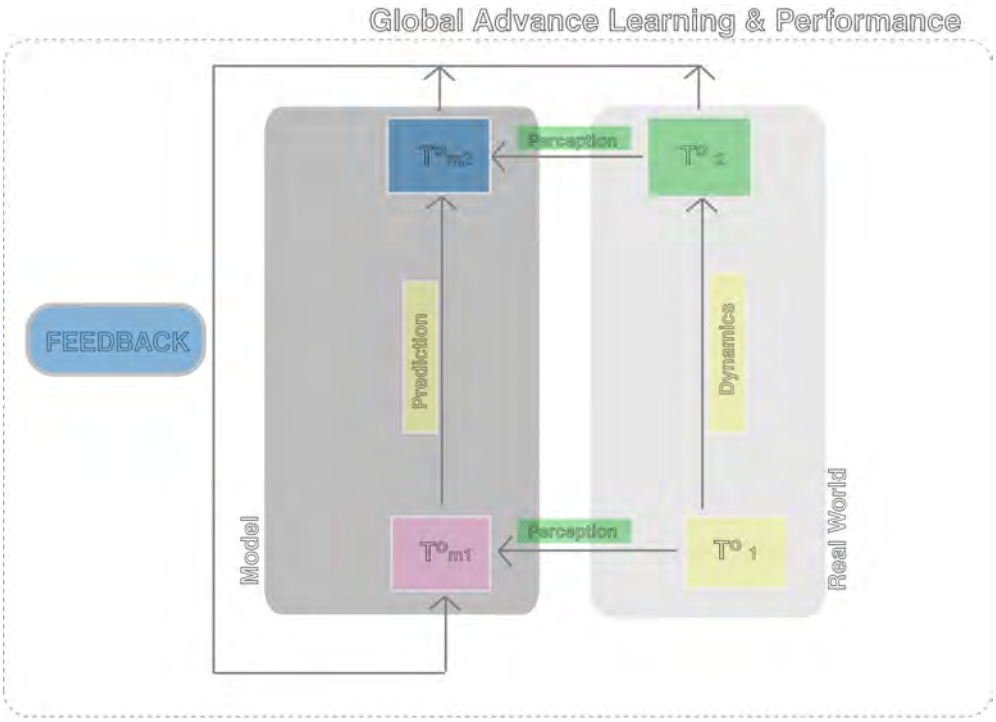
In computer science, evolutionary computation is a subfield of artificial intelligence that involves combinatorial optimization problems.

Solving systems based on principles of evolution and hereditary: such systems maintain a population of potential solutions, they have some selection process based on fitness individuals.

Evolutionary computation uses iterative progress, such as growth or development in a population. This population is then selected in a guided random search using parallel processing to achieve the desired end. Such processes are often inspired by biological mechanisms of evolution.

The process desired for the deployable system learning has to be a mixed one. It will need the power of neural processes for choosing and deciding situations, and the performance of a genetic algorithm for optimizing the pillows pattern and adaptability.

The combination of both sub-processes will generate a global behaviour, where, since the very first day after deployment, the system will perform properly.



The system will be composed by:

- An *ANN* for classifying and deciding the kind of situation we are in, that will learn through a series of labelled set of situations for training.

-A *Genetic Algorithm* that will optimize the performance of the whole set of pillows creating a pattern for adaptability and improvement. Phenotype, genotype, fitness and mutation will decide and teach *EmDeplo* how to act in each situation.

"Self-organization as the adaptive response of a system to changing external conditions."
(see # 145, Allen, 1997)

Scenarios approach. Labelling Situations.

One of the main aims of the implementation of Emergency Deployable in natural disasters will be its customization and adaptation to the different situations.

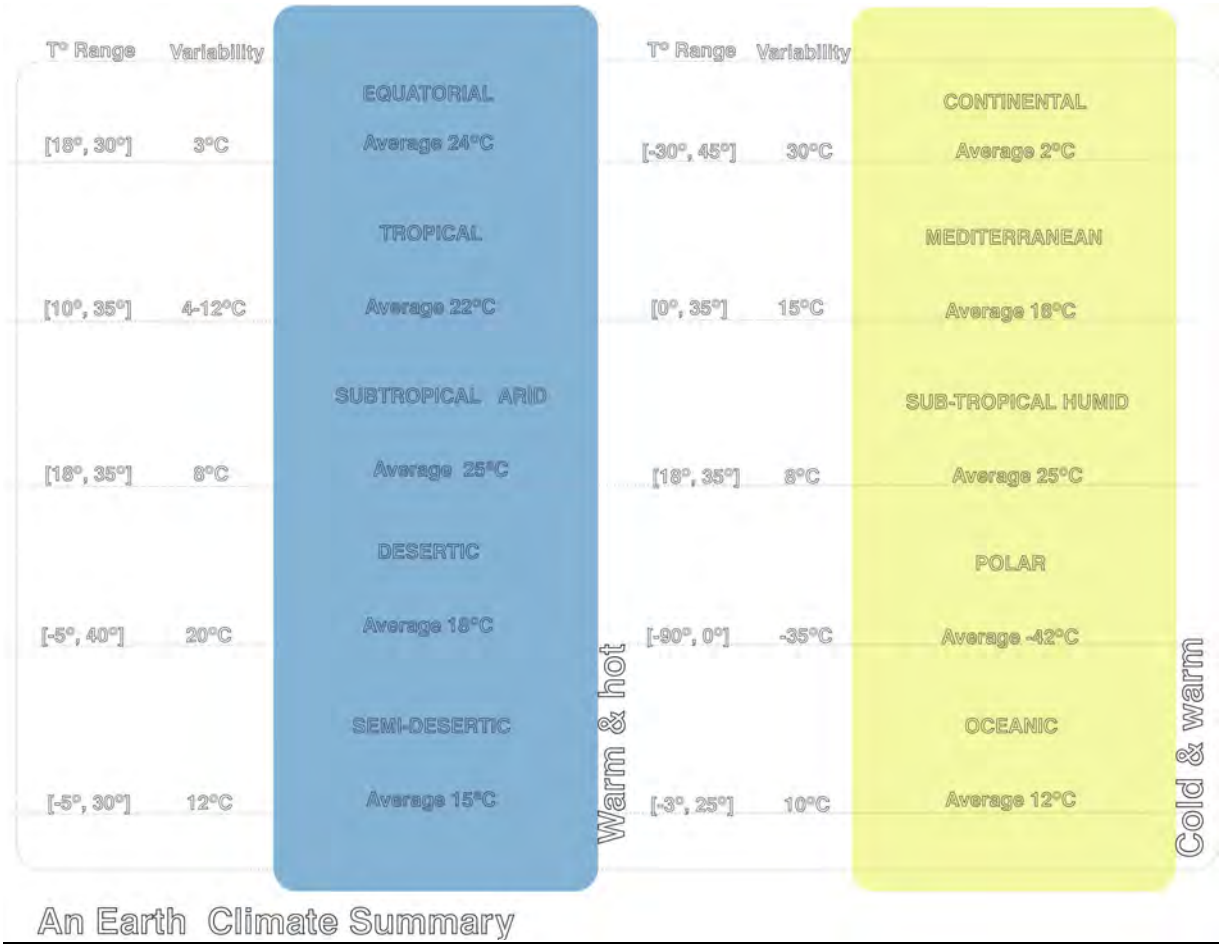
The customization to the different emergencies is based on different performances and designs.

The design customization will be done through the interface used for factory fabrication. The performance customization will be done at site though the internal environmental control.

The place where to draw the separation line between both was one of the first discussions in Emergency Deployable. As it was shown in the short brief chapter about what the system is, the configuration of the parts of the multilayer membrane is different according to four different scenarios. Extra foundation or elevation from the ground are just some of the elections to make with the interface at the factory when constructing the membrane.

The first idea of the learning behaviour was, to make the membrane to decide in which scenario it was, not speaking about kind of disaster but about kind of weather and environment.

So that, it was done an analysis of all the climate possible scenarios. This study was based on the two parameters that were able to be studied and evaluated for the membrane: temperature, humidity and sunlight.



Summarizing, the idea was to make, as a first step, to understand in which of the next scenarios the membrane has been deployed, being able to study the range of temperatures existent:

Scenario 1. Range T° [10 to 35°].
Scenario 2. Range T° [0 to 35°].
Scenario 3. Range T° [-5 to 40°].
Scenario 4. Range T° [-30 to 45°].
Scenario 5. Range T° [-90 to 0°].

On the other hand, the idea of making the system to understand its situation once it has been deployed had one main problem: the impossibility of properly working adapted while the period of learning by studying the temperatures was happening.

Being the system basically designed for emergency situations, the "out of order" period was not acceptable.

Also, being the membrane fabricated and partly customized at factory through the interface, it was considered really useful, just to implement one initial step there to define the climate zone of the emergency. That initial step will be done just adding, one question more to the process of the interface fabrication customization at the factory.

In that way when EmDeplo arrives to the site, the system has already implemented some initial training sets for that climate, and can begin to train and test, since the very first moment.

So, these five scenarios will be used as data input at the factory, implementing in the system and appropriate training set for the initial moments.

5.2.A EmDeplo's ANN.

A neural network is composed by groups of artificial neurones configured to perform some calculation. Neurons are activation units with some inputs and outputs.

ANN can learn association between patterns so , using them as a tool for making the membrane understand the situation trough different patterns of temperatures and data recorded by the sensors,seems quite appropriate.

For the realization of this thesis, it became necessary to reduce the complexity of the first software, considering that the first step of trainnings is already done.

We will work with a scenario 2, in most of the cases represented by a Mediterranean climate, being this also, the climate of Barcelona, where the media-Tic building criticized in previous chapters, is located. We will work with a general maximum-minimum range of temperatures from 0°C to 35°c.

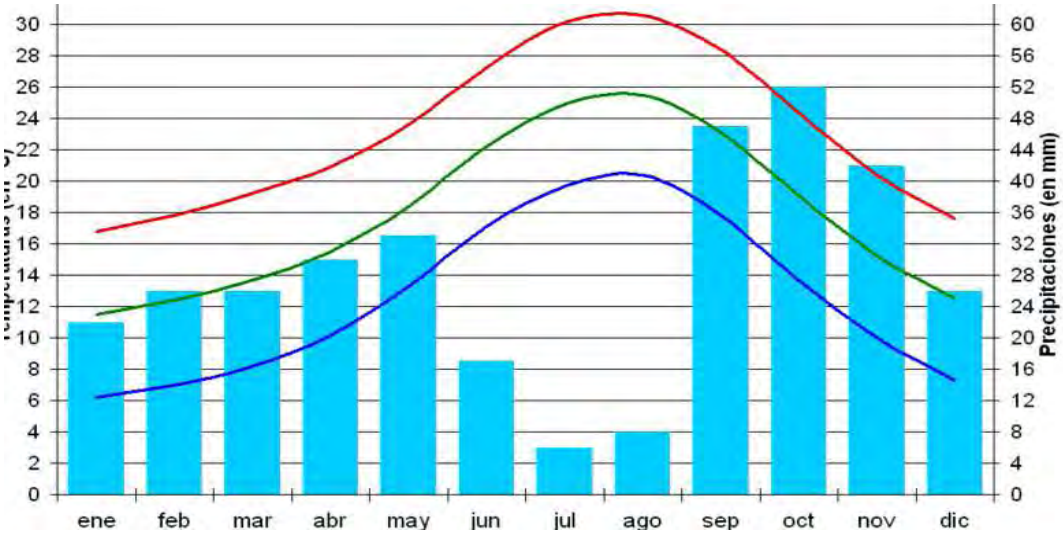
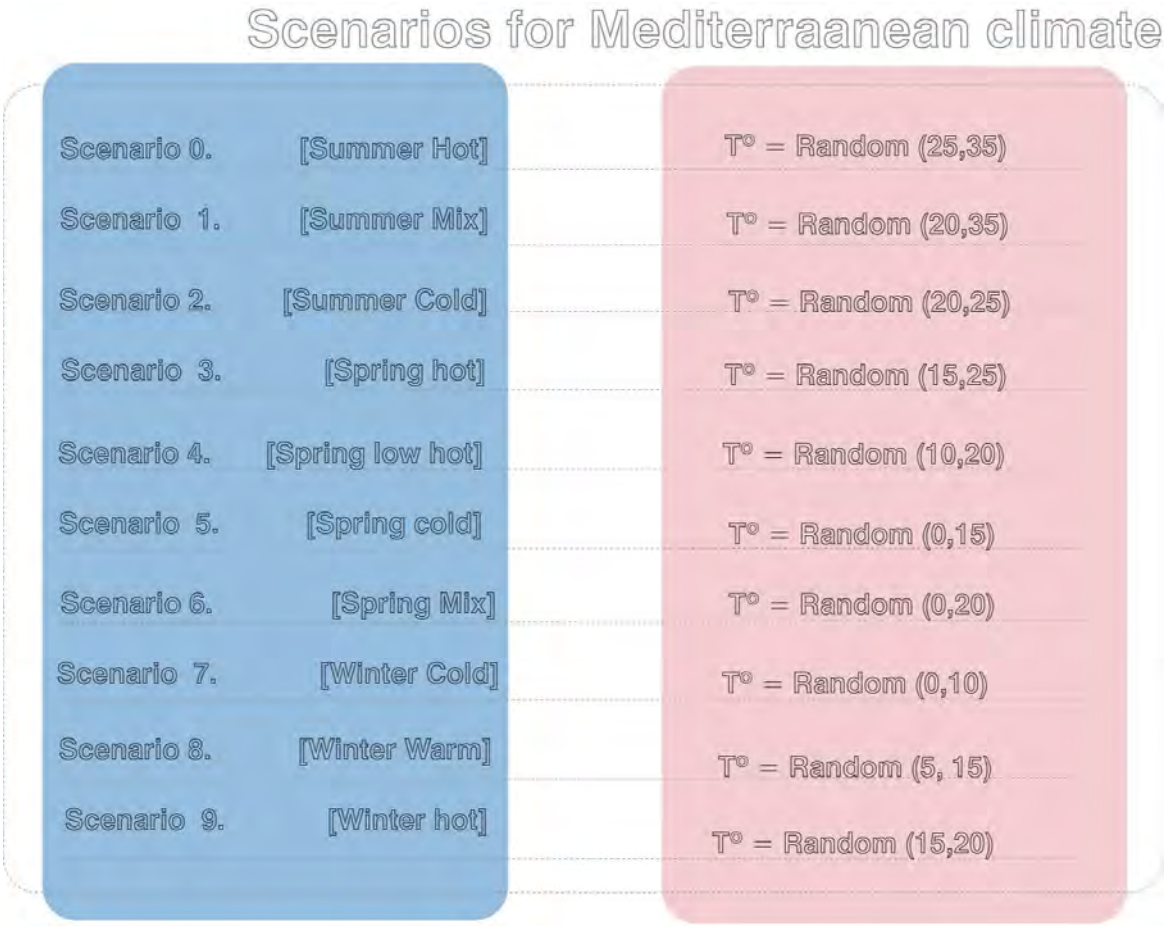


Figure 19. Mediterranean weather characteristics.

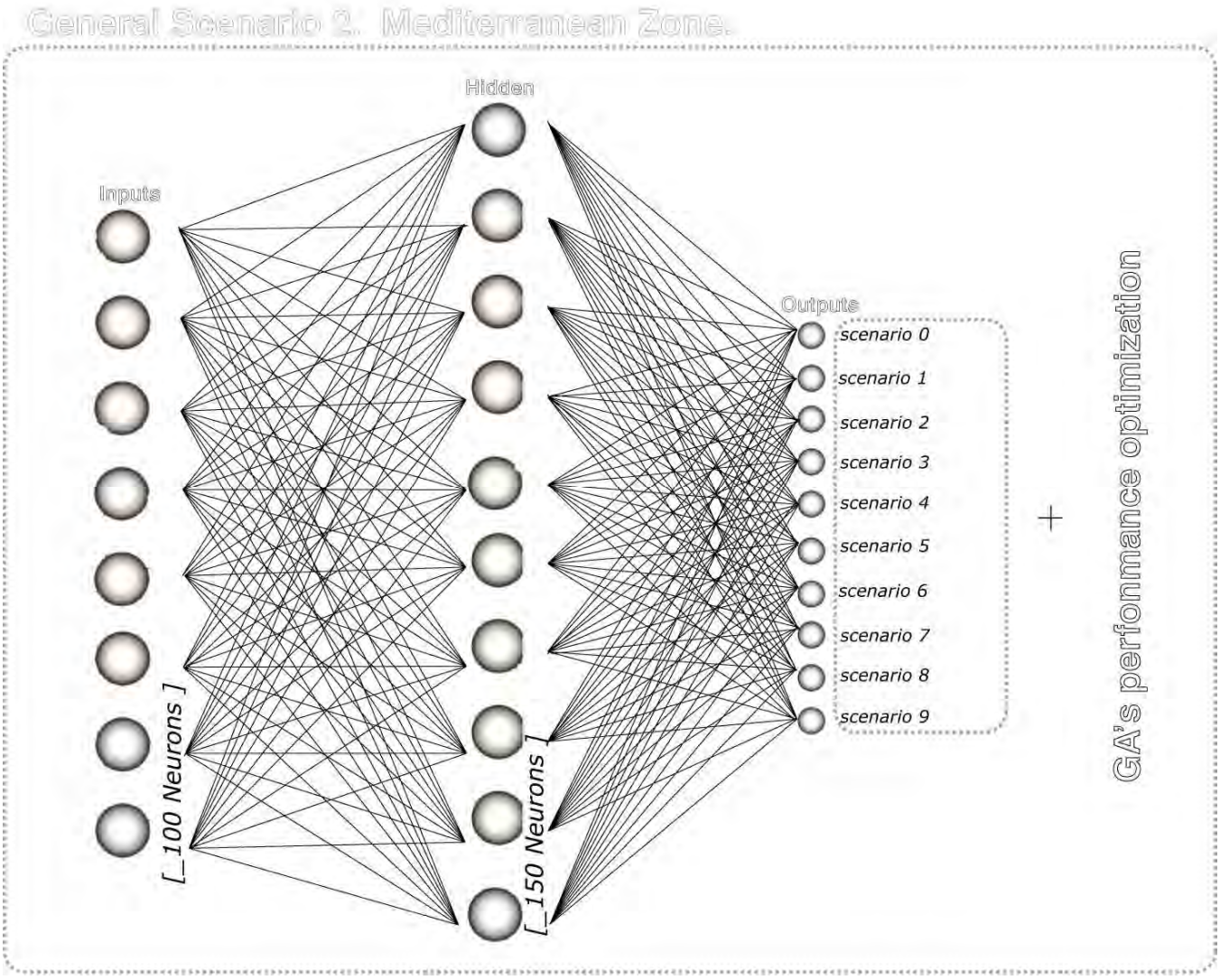
We will work with a sample of 100 pillows and developing a supervised learning process. Due to the impossibility of developing unsupervised learning for the scenario of this thesis and not having the appropriate methods for evaluating it, unsupervised learning, as it will be exposed in the conclusions, will be considered a possible step for further research, as well as the developmentof the system not taking into account this factory first step of software customization.

Remember that *Supervised learning* consist on a learning based on training data, classifying appropriate for classifying known patterns, while, unsupervised learning, consist basically, in finding structures in unlabeled data.

We will consider 10 different situations that we can be in, during the life of the façade. These situations will be the outputs to decide and choose by classification.



The proposed ANN will have 100 neurons in the Input layer, 150 neurons in the hidden layer, corresponding, each one of these 100 neurons to the behaviour and temperature of one pillow of the façade. The proposed outputs will be 10 different performances of the façade, optimized for the GAs previously run. So, depending on the output, one behaviour or the other will start.



Once the structure of the network was clear, It was decided that the testing time, checking the input temperatures pattern will be each 30 minutes. Period during which, the ANN will re-decide again in which scenario irt is in , and, will re-apply the GA. Then, in case some pattern of opening and close pillows will be more efficient than the current one, the façade will be readjusted.

Due to the low variability of temperatures along the night, the testing time during the night will be round 2 hours. In that way, the ANN arrives to the site trained, and it just will be testing for readjusting openings.

Sun shading through the intermediate layer of a patterned ETFE Membrane, can be decided to open and close through the decision taken for optimization of the façade taken by the GA, or, in a different way, as a stimulus-reaction response depending on the light meters readings.

Creating & labelling the training set for the model

The first part of the code implemented in the system was the one able to create and label the training sets for the ANN.

As it was mentioned before we will work with a façade of 100 pillows, so that, with 100 input neurons, having to create sets of temperatures labelled in the 10 different scenarios.

The training set will be a total of 10.000 arrays of 100 elements, each of which will be a possible temperature in the scenario the array is labelled as.

The complete training set will be an array stored as a *.dat file of [10000].

Training set = 10000 pseudo set

New pseudo set= new float [100]; for ex labelled as scenario 1.

Pseudo set[i]= possible temperature in the scenario 1.

Figure 20

Every time the network is run since the very beginning of the process, a new training set is created.

The network will train when started with reading initially a set of 500 training labelled arrays.

For extra training afterwards it will be implemented a function that, when pressing a key, starts an extra training period randomly through the training set, of the length desired.

```
  neuralnet= new Network(num_of_neurons,num_hidden_neurons,num_of_outputs);
  for(int i=0;i<train_pics;i++) {
    int row= (int) floor(random(0,trainSet.length));
    neuralnet.respond (trainSet[row].inputs);
    neuralnet.train(trainSet[row].outputs);
  }
  println("_FIRST TRAIN FINISHED");
}
```

```
_Trainning data T°[]s LOADED
_LABELS LOADED
_ALL DATA LOADED
_FIRST TRAIN FINISHED
_NEW 10 TRAININGs FINISHED
_NEW 10 TRAININGs FINISHED
_NEW 10 TRAININGs FINISHED
```

Figure 21

Defining the ANN

ANN will be implemented in the beginning separately from the training code and the GA and the performance façade guide code.

After defining how to load the data from the *.dat files previously created with the encoded byte info, it will be defined the inputs and outputs arrays plus the behaviour of one neuron.

The input temperatures will be scales for working within the network as floating points between -1 and 1.

In another class, it will be written the network behaviour, ordering the steps for the back propagation, checking always the previous layer errors.

On the other hand, it was considered the kind of sigmoid function to implement.

The simplest output, (either a neuron fires(1) or not(0)), implementing the *unit step function*, which is a function for which, a positive argument makes value 1, and a negative one, value 0, is considered too simple. It is very probable that having that kind of unsmooth type of results, it will be impossible for the ANN to learn appropriately.

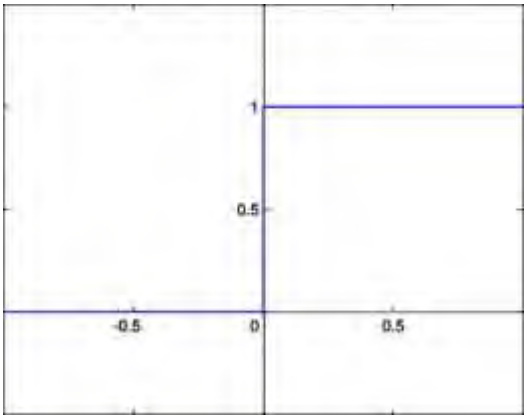


Figure 22

So, as it is necessary a differentiable function for implementing back propagation, the basis of the multilayer perceptron I am using, an hyperbolic tangent function is implemented, using its coding representation as Alasdair Turner in (see # 140, Turner, 2009).

In that way the activation function of the nodes will be,

$$\text{Output} = 2 / (1 + e^{-2*\text{inputs}}) - 1$$

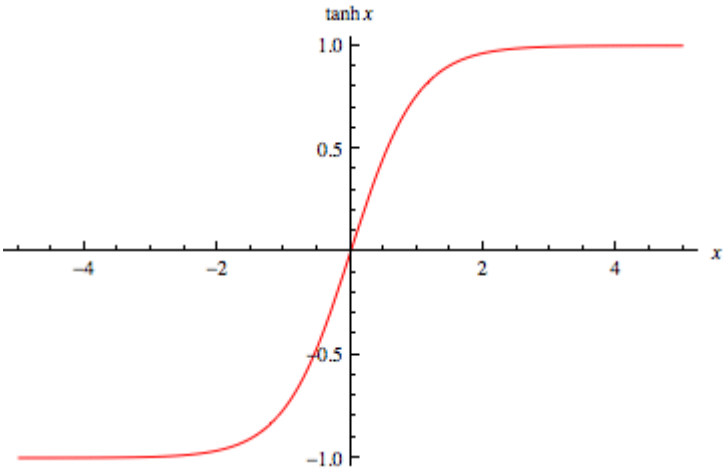


Figure 23

Testing the Learning

Once a minimum amount of training has been done, we can test the learning of our system. Nevertheless, testing the system with less that 100 trainings was demonstrated unsuccessful.

```
int label;
int train_pics=500;//Number of sets used in each trainning round

Network neuralnet;

void setup() {
  size(100,100);
  DATALOAD();
  println("_ALL DATA LOADED");
  setupSigmoid();
  neuralnet= new Network(num_of_neurons,num_hidden_neurons,num_of_outputs);
  for(int i=0;i<train_pics;i++) {
    int row= (int) floor(random(0,trainSet.length));
    neuralnet.respond (trainSet[row].inputs);
    neuralnet.train(trainSet[row].outputs);
  }
  println("_FIRST TRAIN FINISHED");
}
```

```
_LABELS LOADED
_ALL DATA LOADED
_FIRST TRAIN FINISHED
_NEW 10 TRAININGs FINISHED
_NEW 10 TRAININGs FINISHED
_NEW 10 TRAININGs FINISHED
_NEW 10 TRAININGs FINISHED
_TEST INPUTS LOADED
Test Set Number: 1244
Example of scenario number 6,
and
ANN thinks this is scenario  3

7
```

Figure 24

The ANN

```
int num_of_neurons=100;//Number of pillows of the façade
int num_hidden_neurons=225;
int num_train_sets=10000;//sets of labeled [] of Tº for trainning
int num_of_outputs=10;//Number of Scenarios for that climate
int facade_size=100;//Number of pillows of the façade
int label;
int train_pics=500;//Number of sets used in each trainning round
int num_train=100;
int set_sc_size=1000;
int number_scenarios=10;
int size_train_matrix=int(facade_size*set_sc_size);
int size_final_train_set=int(size_train_matrix*number_scenarios);
int label_size_array=10000;
Train new_train_set;
Labels new_label_set;
Network neuralnet;

void setup() {
new_train_set=new Train();
new_train_set.create_trainning_set();
println("Trainning Set file Saved");
new_label_set=new Labels();
new_label_set.create_labels();
println("Labels file Saved");
size(100,100);
DATALOAD();
println("_ALL DATA LOADED");
setupSigmoid();
neuralnet= new Network(num_of_neurons,num_hidden_neurons,num_of_outputs);
for(int i=0;i<train_pics;i++) {
int row= (int) floor(random(0,trainSet.length));
neuralnet.respond (trainSet[row].inputs);
neuralnet.train(trainSet[row].outputs);
}
println("_FIRST TRAIN FINISHED");
}

void draw()
{
//To start more trainning
if(keyPressed) {
for(int j=0;j<num_train;j++) {
for(int i=0;i<train_pics;i++) {
int row= (int) floor(random(0,trainSet.length));
neuralnet.respond (trainSet[row].inputs);
neuralnet.train(trainSet[row].outputs);
}
}
println("_NEW 10 TRAININGs FINISHED");
}
}
//this is for the test set_____________________________________________
void mousePressed()
```

```
{//set the testset float[]
if (mousePressed) {
println("_TEST INPUTS LOADED");
data_test();
}
}
```

```
int imagesArrayLenght=(num_train_sets*facade_size);
int []labels= new int[num_train_sets];
byte [] trainning_array;
byte []labels_byte;
float [] mediterranean_climate;//T° [] with values 0-45 degrees

class Data
{
float [] inputs;
float [] outputs;
float [] inputs_test;
int label_output;
Data()
{
inputs=new float [num_of_neurons];
outputs= new float[num_of_outputs];
}
//___load the images & create the inputs[] OF FLOATS_____________________
void LoadTemperatures(byte [] trainning_array, int facade_size) {
for(int i=0; i<num_of_neurons; i++) {
inputs[i]= int (trainning_array[i+facade_size])/23-1.0;//because range
}
}
//___load the labels & create the outputs [] OF FLOATS_____________________
//there are 10 different labels but we scale the outputs to yes & no
void LoadLabels (int [] labels, int offset)
{
label_output= int(labels[offset]);
for (int i=0; i<num_of_outputs;i++) {
if(i==label_output) {
outputs[i]=1.0;
}
else {
outputs[i]=-1.0;
}
}
}
//create the inputs for testing_____________________________________________
void define_test_inputs( byte [] trainning_array, int facade_size) {
inputs_test=new float [num_of_neurons];
for (int i = 0; i < num_of_neurons; i++) {
inputs_test[i] = int(trainning_array[i+facade_size])/ 23.0 - 1.0;
}
}
}
//end of the class_____________________________________________________________
//FUNCTION TO LOAD THE DATA(we call it in the main tab)_____________________
Data [] trainSet;
void DATALOAD() {
//____Load trainning []s_____________________________________
trainning_array =loadBytes("trainning_set.dat");
```

```
println("_Trainning data T°[]s LOADED");
//____Load labels( 10 scenarios)_____________________
labels_byte=loadBytes("labels.dat");
for(int i=0;i<labels_byte.length;i++) {
labels[i] = labels_byte [i];
}
println("_LABELS LOADED");
//set the trainSet float[]_____________________________
trainSet=new Data [int(num_train_sets/5*4)];
int trainPos=0;
int testPos = 0;
testSet=new Data [int(num_train_sets/5)];
for(int i=0;i<num_train_sets;i++) {
if((i % 5 != 0)) {
trainSet [trainPos]= new Data();
trainSet [trainPos].LoadTemperatures(trainning_array, i*facade_size);
trainSet [trainPos].LoadLabels(labels, i);
trainPos++;
}
else {
testSet[testPos] = new Data();
testSet[testPos].define_test_inputs(trainning_array, i * facade_size);
testSet[testPos].LoadLabels(labels, i);
testPos++;
}
}
}
//END OF LOAD DATALOAD FUNCTION_____________________________________


Trainning Set Creation_____________________________________________


//int size_train_matrix=int(facade_size*set_sc_size);
//int size_final_train_set=int(size_train_matrix*number_scenarios);
byte [] Final_trainning_array;
//this is the complete trainning set array
float [] set_of_temperatures;
float [] trainning_array_0;
float [] trainning_array_1;
float [] trainning_array_2;
float [] trainning_array_3;
float [] trainning_array_4;
float [] trainning_array_5;
float [] trainning_array_6;
float [] trainning_array_7;
float [] trainning_array_8;
float [] trainning_array_9;
byte[] b_0;
byte[] b_1;
byte[] b_2;
byte[] b_3;
byte[] b_4;
byte[] b_5;
byte[] b_6;
byte[] b_7;
byte[] b_8;
```

```
byte[] b_9;
byte[]big_array;
byte[]big_array_0;
byte[]big_array_1;
byte[]big_array_2;
byte[]big_array_3;
byte[]parcial_array_0;
byte[]parcial_array_1;
byte[]parcial_array_a;
class Train {
Train() {
}
void create_trainning_set() {
//"mix" situation of the façade result of a hot weather but with part of
//para clima mediterraneo, matriz general=[0,45]
//Scenario 0_____________________________________________________
//summer 1. [hot]
trainning_array_0 = new float[int(size_train_matrix)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_0[i]=random(25,35);
}
byte[] b_0=byte(trainning_array_0);
//Scenario 1_____________________________________________________
//summer 2. [mix]
trainning_array_1 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_1[i] =random(20,35);
}
byte[] b_1=byte(trainning_array_1);
//Scenario 2_____________________________________________________
//summer 3. [summer "cold"] = random(20,25)
trainning_array_2 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_2[i] =random(20,25);
}
byte[] b_2=byte(trainning_array_2);
//Scenario 3_____________________________________________________
//spring 1. [hot] =random (15, 25)
trainning_array_3 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_3[i] =random(15,25);
}
byte[] b_3=byte(trainning_array_3);
//Scenario 4_____________________________________________________
//spring 2. [not so hot]=random(10,20)
trainning_array_4 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_4[i] =random(10,20);
}
byte[] b_4=byte(trainning_array_4);
//Scenario 5_____________________________________________________
//spring 3. [cold]=random(0,15)
trainning_array_5 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_5[i] =random(0,15);
}
byte[] b_5=byte(trainning_array_5);
//Scenario 6_____________________________________________________
```

```
//spring 4. [mix]=random(0,15)+ random (10,20)
trainning_array_6 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_6[i] =random(0,20);
}
byte[] b_6=byte(trainning_array_6);
//Scenario 7_____________________________________________________
//winter 1. [cold]=random (0,10)
trainning_array_7 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_7[i] =random(0,10);
}
byte[] b_7=byte(trainning_array_7);
// //Scenario 8_____________________________________________________
// //winter 2. [warm]=random (5,15)
trainning_array_8 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_8[i] =random(5,15);
}
byte[] b_8=byte(trainning_array_8);
//Scenario 9_____________________________________________________
//winter 3. [hot]=random(10,20)
trainning_array_9 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_9[i] =random(15,20);
}
byte[] b_9=byte(trainning_array_9);
//Creation of the complete trainning set array___________________
big_array=concat(b_0, b_1);
big_array_0=concat(b_2,b_3);
big_array_1=concat(b_4, b_5);
big_array_2=concat(b_6, b_7);
big_array_3=concat(b_8,b_9);
parcial_array_0=concat(big_array,big_array_0);
parcial_array_1=concat(big_array_1,big_array_2);
parcial_array_a=concat(parcial_array_0,parcial_array_1);
Final_trainning_array=new byte [1000000];
//this is the complete trainning set array
for(int i=0;i<1000000;i++) {
if (i<800000) {
Final_trainning_array[i]= parcial_array_a[i];
}
if (i>=800000) {
Final_trainning_array[i]=big_array_3[i-800000];
}
}
//println(parcial_array_a.length);
println("Trainning Set created. []size= " + Final_trainning_array.length
saveBytes("trainning_set.dat", Final_trainning_array);
}
}
```

```
class Network
{
Neuron [] m_input_layer;
Neuron [] m_hidden_layer;
```

```
Neuron [] m_output_layer;
Network(int inputs, int hidden, int outputs)
{
m_input_layer = new Neuron [inputs];
m_hidden_layer = new Neuron [hidden];
m_output_layer = new Neuron [outputs];
// set up the network topology
for (int i = 0; i < m_input_layer.length; i++) {
m_input_layer[i] = new Neuron();
}
// route the input layer to the hidden layer
for (int j = 0; j < m_hidden_layer.length; j++) {
m_hidden_layer[j] = new Neuron(m_input_layer);
}
// route the hidden layer to the output layer
for (int k = 0; k < m_output_layer.length; k++) {
m_output_layer[k] = new Neuron(m_hidden_layer);
}
}
int respond(float [] inputs)// for TRAINNING
{
float [] responses = new float [m_output_layer.length];
for (int i = 0; i < m_input_layer.length; i++) {
m_input_layer[i].m_output = inputs[i];
}
for (int j = 0; j < m_hidden_layer.length; j++) {
m_hidden_layer[j].respond();
}
// and finally feed forward to the output layer
for (int k = 0; k < m_output_layer.length; k++) {
responses[k] = m_output_layer[k].respond();
}
// now check the best response:
int response = -1;
float best = max(responses);
for (int a = 0; a < responses.length; a++) {
if (responses[a] == best) {
response = a;
}
}
return response;
}
void train(float [] outputs)
{
// adjust the output layer
for (int k = 0; k < m_output_layer.length; k++) {
m_output_layer[k].finderror(outputs[k]);
m_output_layer[k].train();
}
// propagate back to the hidden layer
for (int j = 0; j < m_hidden_layer.length; j++) {
m_hidden_layer[j].train();
}
}
}
end of network class_____________________________________________________
```

```
float LEARNING_RATE = 0.01;

class Neuron
{
Neuron [] m_inputs;
float [] m_weights;
float m_threshold;
float m_output;
float m_error;
// the input layer of neurons have no inputs:
Neuron()
{
m_threshold = 0.0;
m_error = 0.0;
// initial random output
m_output = lookupSigmoid(random(-5.0,5.0));
}
// all other layers (hidden and output) have
// neural inputs
Neuron(Neuron [] inputs)
{
m_inputs = new Neuron [inputs.length];
m_weights = new float [inputs.length];
for (int i = 0; i < inputs.length; i++) {
m_inputs[i] = inputs[i];
m_weights[i] = random(-1.0,1.0);//here I enter the initial random weig
}
m_threshold = random(-1.0,1.0);
m_error = 0.0;
// initial random output
m_output = lookupSigmoid(random(-5.0,5.0));
}
// respond looks at the layer below, and prepares a response:
float respond()
{
float input = 0.0;
for (int i = 0; i < m_inputs.length; i++) {
input += m_inputs[i].m_output * m_weights[i];
}
m_output = lookupSigmoid(input + m_threshold);
// reset our error value ready for training
m_error = 0.0;
return m_output;
}
// find error is used on the output neurons
void finderror(float desired)
{
m_error = desired - m_output;
}
// train adjusts the weights by comparing actual output to correct output
void train()
{
float delta = (1.0 - m_output) * (1.0 + m_output) * m_error * LEARNING_R
for (int i = 0; i < m_inputs.length; i++) {
// tell the next layer down what it's doing wrong
m_inputs[i].m_error += m_weights[i] * m_error;
// correct our weights
m_weights[i] += m_inputs[i].m_output * delta;
```

```
}
}
}
```

```
float [] g_sigmoid = new float [200];

void setupSigmoid() {
for (int i = 0; i < 200; i++) {
float x = (i / 20.0) - 5.0;
g_sigmoid[i] = 2.0 / (1.0 + exp(-2.0 * x)) - 1.0;
}
}
float lookupSigmoid(float x) {
return g_sigmoid[constrain((int) floor((x + 5.0) * 20.0),0,199)];
}
```

```
class Labels {
byte[]labels_set;
Labels() {
}
void create_labels() {
labels_set=new byte[label_size_array];
for (int i=0;i<label_size_array; i++) {
if (i<1000) {
labels_set[i]=0;
}
if ((i>=1000) && (i<2000)) {
labels_set[i]=1;
}
if ((i>=2000)&&(i<3000)) {
labels_set[i]=2;
}
if ((i>=3000)&&(i<4000)) {
labels_set[i]=3;
}
if ((i>=4000)&&(i<5000)) {
labels_set[i]=4;
}
if ((i>=5000)&&(i<6000)) {
labels_set[i]=5;
}
if ((i>=6000)&&(i<7000)) {
labels_set[i]=6;
}
if ((i>=7000)&&(i<8000)) {
labels_set[i]=7;
}
if ((i>=8000)&&(i<9000)) {
labels_set[i]=8;
}
if ((i>=9000)&&(i<10000)) {
labels_set[i]=9;
```

```
}
}
println("Label created. []size= " + labels_set.length);
saveBytes("labels.dat", labels_set);
}
}
```

```
//For testing_____________________________________________
byte[]test_arra;
Data [] testSet;
float[] Temp_tested;
//// for (int i = 0; i < num_of_neurons; i++) {
//// float Tempe= (inputs[i]+1)*23;
//// //println(Tempe);
void data_test() {
int row = (int) floor(random(0,testSet.length));
println("Test Set Number: "+row);
//println("Above is the set of Tº tested");
int response=-1;
response = neuralnet.respond(testSet[row].inputs_test);
int actual=-1;
actual = testSet[row].label_output;
println("Example of scenario number "+actual+",");
println("and");
println("ANN thinks this is scenario "+response);
if (response==0) { //DO WHATEVER i WANT IT TO DO FOR ADJUSTING TEMPERATURE
}
if (response==1) {
}
if (response==2) {
}
if (response==3) {
}
if (response==4) {
}
if (response==5) {
}
if (response==6) {
}
if (response==7) {
}
if (response==8) {
}
if (response==9) {
}
}
//SCENARIOS FOR MEDITERRANEAN climate
//Scenario 0_____________________
//summer 1. [hot] random(25,35)
//Scenario 1_____________________
//summer 2. [mix]random(20,35);
//Scenario 2_____________________
//summer 3. [summer "cold"] = random(20,25)
//Scenario 3_____________________
//spring 1. [hot] =random (15, 25)
```

```
//Scenario 4_____________________
//spring 2. [not so hot]=random(10,20)
//Scenario 5_____________________
//spring 3. [cold]=random(0,15)
//Scenario 6_____________________________________
//spring 4. [mix]=random(0,15)+ random (10,20)
//Scenario 7_____________________________
//winter 1. [cold]=random (0,10)
//Scenario 8_____________________
//winter 2. [warm]=random (5,15)
//Scenario 9_____________________________
                        //winter 3. [hot]=random(15,20
```

5.2.B. The GA.

Searching or optimizing algorithms inspired by biological evolution are based on four basic ideas:

- Multiple searching points or solution candidates.
- Operations inspired by biological evolution, such crossover and mutation.
- Probabilistic search and probabilistic operations.
- Using little information of searching space.

For starting to configure the genetic algorithm structure that we are going to use, we need to decide the parameters to optimize. Once we have decided trough the ANN in which scenario we are in, we will be using:

Shading Parameterization

[sun, shade] = [1, 0]

These will be the parameters that will compose the local behaviour. Through a solar factor calculus, as it was done in the Media-tic, a local performance will be implemented for not allowing the solar factor be higher than FS 10. This kind of calculus, as mentioned Iaso's consultant is normally done by an external company due to its complexity. For our model then, it will be only used a binary system in which, if the FS is higher than 10, the intermediate pattern layer will open reducing 85% of ultraviolet light.

Temperature Range for Survival: Goal efficiency

[< T° desired, T° desired, < T° desired]= [22° (will be used for this example)]

Temperature Inputs of the current situation to optimise

Input_temperatures= new temperatures [100]

We will generate the chromosome of the façade we are working with, through an array that will be a sequence of opening and closing possibilities for the 100 pillows. In that way our genotype will be an array of 100 elements that indicates the initial position of the pillows we are starting with. It is also possible, to implement this initial gene array as a random array of positions of the pillows.

The positions considered will be closed, open or half-open.

Genes = new [100];

Genes [i]= [open/ close status]

Façade genotype = [c, o, c, h, c, c, h, o, c, c, o……..]

The fitness

In the beginning multi-objective optimization was not implemented.

The general idea was to optimize the genes [] of the façade for obtaining a desired temperature of 22º. Several simple fitness functions were implemented, based on the idea of obtaining an ideal temperature for each pillow of the façade.

```
fitness option 1_____________________
    float fitness = 0.0;
    fitness += sum;(addition of all pillows temperatures)
    fitness -=(sum/(ideal_temp*number_pillows));
   return fitness;

fitness option 2_____________________
    float fitness = 0.0;
   int average=sum/number_pillows;
   fitness+=100-abs(average-22);
   fitness-=100+abs(average-22);
    return fitness;

fitness option 3_____________________
  float fitness = 0.0;
  fitness += 1/1+abs(sum-(2200));
  fitness -=sum-(2200);
  return fitness;

fitness option 4_____________________
   float fitness = 0.0;
    int average=sum/number_pillows;
    fitness+=1/1+abs(average-22);
```

```
fitness-=abs(average-22);
return fitness;
```

The first results obtained with this kind of fitness functions were clearly unsatisfactory.

A very premature convergence and stagnation was shown with a fitness of 4401. Main reason of which was, probably, the existence of some individuals with a high predominance in fitness over the others.

Also, in this early fitness functions, it was not considered the extremely important relationship *length of opening period/temperature ascend-descend*.

The trials were done also with percentage of mutations between a 0.01-0.05%, basically:

Probability of mutation = 1/chromosome length

And with the initial parameters:

```
float mutation_rate=0.01;
int number_of_individuals=100;
int number_evolutions=100;
int ideal_temp=22; //desired temperature in each pillow
```

So that, it was started then a series of analysis of the different options to implement in the algorithm as the basis of possible improvements.

Selection Method & Phenotype Definition

Stagnation and premature convergence appeared in the very first trials. As the can be caused by a % of mutation too low or by a population size too small, several trials were done increasing both parameters but in the end, they resulted unsuccessful.

The fitness was stagnated in 4401. Maybe also a premature convergence existed.

So it was considered to revise the selection method implemented.

The first idea was to analyse the method implemented in (see # 140, Turner, 2009) as an interpretation for a *Rank Selection*:

```
Individual select() {
//we tried to obtain a high number as the []is ordered from lower fitness to big fitness
  int which = (int) floor( (100.0 - 1e-6) *(1.0 - (random(0,1))2 );
  return m_pop[which];
}
```

In this way, it is creating a much bigger possibility of obtaining a high number within the range [0,99], which will make the population to select the last members of the array which will be the ones with bigger fitness as we previously ordered it in that way.

So, the next step trying to improve stagnation, was to implement the thermal relationship between the degree of openness of the pillows and the temperature variability.

As we will be considering Area = 1,

$$\Delta \text{ Temperature} = (Q * \text{thickness})/\lambda$$

It will be implemented in the phenotype for each of the 100 pillows:

$$t\_f[i]\text{-temperatures}[i] = G * \text{genes}[i]/\lambda$$

Being genes[i], the thickness of that pillow, meaning:

1 if open (50 cm * 2 =1m)
0.5 m if half open
0.000125 if closed, (0 will be used in this case).

the façade genotype will be,

Genes = new [100];
Genes [i]= [open/ close status]
Façade genotype = [0, 1, 0.5, 0.5, 0, 0, 1, 1, 0……..]

T_f are the final temperatures and Temperatures, are the input temperatures at the beginning of the optimization performance.

0.017 will be the λ if open and 0.02 if closed.

G will be the heat flux, that is a variable that we do not know.

This genes array will be the data we will need to implement when the optimized the result is sent to the Arduino that will control the façade.

The genes array of the fittest individual will be sent to the chip as the orders for pillows.

It should be mentioned that the thermal calculus was simplified to a flat multiple surface calculus due to the high complexity of a curve non-circular multilayer calculus. In that way, for further and really strict results minimizing the temperatures error, the complete and exact curvature calculus should be done. this consideration is also taken into account as a possibility of further research.

The phenotype function will be changed to,

GA_6 | Genotype | Individual | Phenotype § | Population | input_temperatures

```
float Q;//Thermal flux
float c;
class Phenotype
{
  byte [] temperatures;    //T are the input temperatures
  float [] t_f;    //t_f are the output temperatures
  float [] temp;
  int sum;
  float [] a_gen;
  float a;
  Phenotype(Genotype g) {

    temperatures = new byte [number_pillows];
    temperatures= loadBytes("temperatures.dat");
    temp= new float [number_pillows];
    t_f= new float [number_pillows];
    a_gen= new float [number_pillows];

    int sum=0;
    int pillow_number=0;

    for (int i=0; i<number_pillows;i++) {
      temp[i]=float(temperatures[i]);//transforms byte into floats
      t_f[i]=temp[i]+((Q*g.m_genes[i])/0.017);
      sum++;
      pillow_number++;
      a_gen[i]=g.m_genes[i];
    }
  }
```

Figure 25

One of the most important concept in evolutionary computation is the relationship genotype-phenotype. Sometimes this relationship is not straight forward and it is not easy understand the environment to which the phenotype is related.

In EmDeplo, this environment in which our phenotype exists is the thermal relationship environment-material.

Once implemented the new relation genotype-phenotype, a new fitness function should be implemented to avoid premature convergence and stagnation. This function will be also related with the thermal behaviour.

```
float fitness_function() {

  float fitness = 0.0;

  //fitness function without Q_____________
  float all=0.0;
  for (int i=0; i<number_pillows;i++) {
    c=(a_gen[i])/0.017;
    t_f[i]=c-temperatures[i];
    all+=t_f[i];
  }
  println("Sum of temperatures = "+ all);
  fitness+=1/(1+sq(2200-all));
  fitness-=Q;

  if(all==2200) {
    println("solved");
  }
  //_______________________________________

  return fitness;
```

Figure 26

Stagnation disappears. Optimization begins. The system is working for obtaining a fitness of one that will mean that the whole temperature is 22ºC but also that all the different spaces in the building have 22ºC.

Of course this behaviour can be programmed for optimizing different temperatures in the different spaces according to the use of the room, or, also, according to the time of the day. For

the development on this thesis, it was simplified the goal to this unique temperature as the basis of the thesis was to demostrate the the optimization was possible. Possible configurations of different temperatures and spaced will be able to be programmed in the future.

Also, it can be considered that different solar factors can be needed depending on use and timing. Also it will be implemented a similar visualization of the façade as the one that was implemented for the stimulus-reaction model.

The maximum optimization is obtained at a 0,4 fitness with 1000 evolutions. Permanent performance will make the system to improve until arriving at a bigger fitness.

Considering that also having a small thermal flux is a condition, probably maximum fitness reachable will be 0,9. Nevertheless it is appreciated that this condition of the fitness is quite weak. It is not really provoking a real homogeneous desired temperature but an addition in which one pillow can be 41°C and other one 11°C as shown in the diagram.

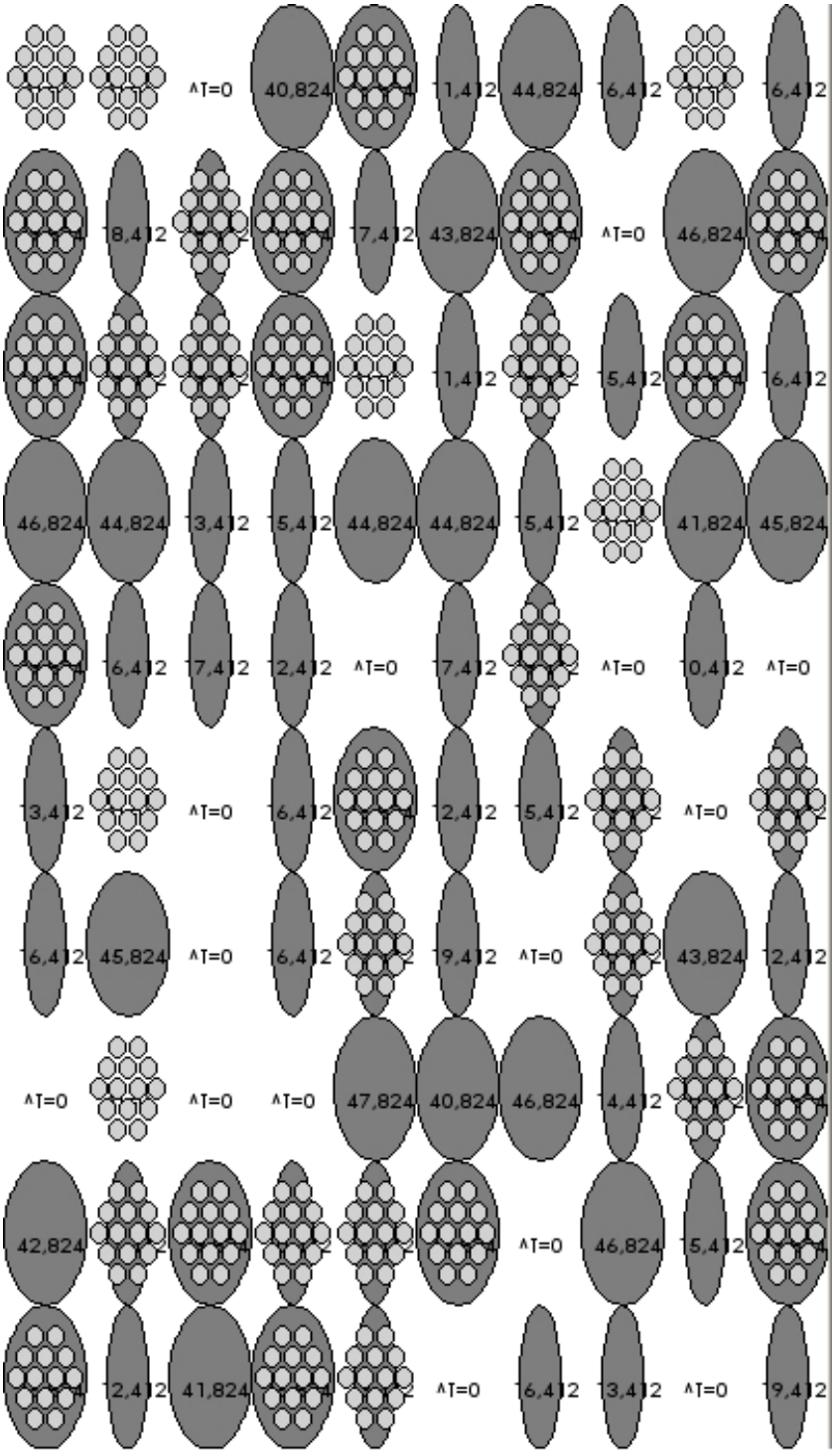


Figure 27

So that some changes on the fitness function were implemented. We will discard the sign of the ΔT and we will implement a new relationship that will help us to obtain a desired temperature for each pillow.

```
float fitness_function() {

  float fitness = 0.0;

  //fitness function without Q_____________________
  float all=0.0;
  float h=0.0;
  float a=0.0;
  for (int i=0; i<number_pillows;i++) {
    c=(a_gen[i])/0.017;
    t_f[i]=c-temperatures[i];
    h=abs(t_f[i]-22);
    println("h= "+h);
    all+=t_f[i];
    a+=h;
  }
  println("Sum of temperatures = "+ all);
  fitness+=1/(1+sq(2200-all));
  fitness-=a;
  fitness-=Q;
  if(a==0) {
    println("solved");
  }
  //_____________________________________________

  return fitness;
 }
}
```

Figure 28

With this improvements the optimization is improving significantly, reaching an homogeneous temperature on the façade near 18ºC. But it is still not achieving the final goal. Starting with a fitness smaller than -2000 evolution stagnation appears with a fitness of -727 around evolution number 1000.

The algorithm was left working for 24 hours to check the evolution convergence and the result of the experiment did not change. Maximum façade optimization for the situation was obtaining the performance shown below.

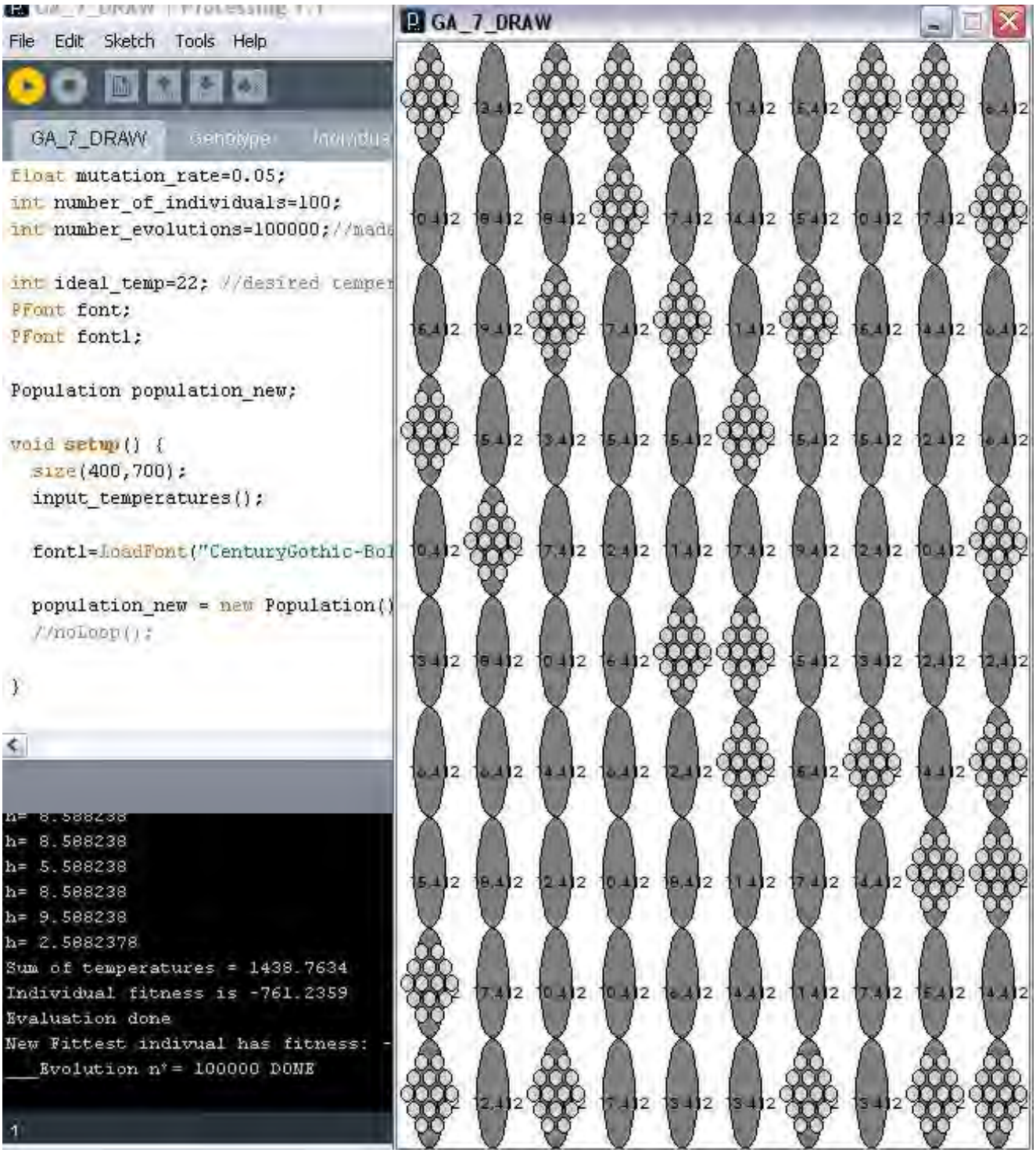

File Edit Sketch Tools Help
GA_7_DRAW
float mutation_rate=0.05;
int number_of_individuals=100;
int number_evolutions=100000;//made
int ideal_temp=22; //desired temper
PFont font;
PFont font1;
Population population_new;
void setup() {
size(400,700);
input_temperatures();
font1=loadFont("CenturyGothic-Bol
population_new = new Population()
//noLoop();
}
h= 8.588238
h= 5.588238
h= 8.588238
h= 9.588238
h= 2.5882378
Sum of temperatures = 1438.7634
Individual fitness is -761.2359
Evalustion done
New Fittest indivual has fitness: -
Evolution n°= 100000 DONE
GA_7_DRAW

Figure 29

Selection method.

Trying to improve the genetic algorithm performance it was carried out the study of the possibility of implementing different selection methods. The method implemented until this step of the thesis, is the Alasdair Tuner interpretation of the *Rank Selection*. Creating an array of fitness and sorting it from low to high fitness we are currently ranking the individuals' fitness from 0 to 99. Despite this method decreases the risk of premature convergence the maximum fitness obtained, -727 will try to be improve with different combination of selection methods and variations of the current fitness function.

Premature convergence can be caused by:

- A percentage of mutation too low.
- A population size too small.
- A selection pressure too low.

The Roulette Wheel Selection, based on probabilities,

$$P_i = f_i / \sum_{1:n} f_j$$

Will not be implemented due to the danger of premature convergence if it exists a clearly dominant individual.

The different possibilities of scaling methods were really good compared in the paper "*Comparison of fitness scaling functions in genetic algorithms with applications to optical processing*" of Farzad A. Sadjadi (see # 142, Sadjadi).

The author compares them on the basis:

1. Traditional linear scaling

$f'(x)=a*f(x) + b$

2. Rank scaling

$f_{ranked} = [(rank-1) * (desired\ selection\ pressure\ -1)] / population\ size\ -1$

3. Exponential scaling

$f_{exponential} = m^{(r-1)}$

Low will generate high selection pressure

4. Top scaling

f_{top} { S *population size for r ≤ number of individuals to scale
{ 0 for r< number of individuals to scale

Measuring Variance as what he considers a good measure for diversity, the author concludes, after a series of experiments, that Exponential Scaling will be the best option for cases in which the only interest is to find a solution as quickly as possible.

On the other hand, Sadjadi considers that, for a complicated fitness function with the real need of diversity to be kept through several evolutions, Top Scaling will be the more appropriate method.

Other author considered is Mr. S. N. Sivanandam. In his "Introduction to Genetic Algorithms" , he considers a different fitness scaling classification: (see # 146, Deepa, 2008)

- Linear Scaling;

  Considering a $f'_{average}$= to $f_{average}$ and, to avoid dominancy of super individuals,

  f'_{max} = nº copies of highly fit individuals * f' av

- σ Truncation;

  As a process to avoid negative scaled fitness results obtained because 1 or 2 very weak individuals.

- Power Law;

  As a previous step of Roulette Wheel Selection, considering,

  $f' = f^{k}$(raw fitness f)

As it has been already decided not to implement Roulette wheel Method it will not be considered power law and, having discarded traditional linear scaling, it will be in the first placed considered, definitely, Top Scaling.

Tournament Selection.

As a second option to implement we will consider Tournament selection. The competition within a subset of individuals for the best fitness appears to be too a good path to take.

This method allows us to control the selection pressure through the control of the subset size. With a minimum subset of two random individuals, a high selection pressure is considered above five members. In our case it will be implemented as shown:

```
void evolve() {//creates a new individual in the population locating

  //Torunament Selection Method_____________________________________
  int tournamentSize=5;

  Individual [] parents;
  parents= new Individual [2];
  // IChromosome[] parents = new IChromosome[2];
  Individual [] sub_set;
  sub_set = new Individual [tournamentSize];
  // pick a random place to start within the population
  int start=0;
  start = int(random(-1,(number_pillows-5)));
  println("start point for subset is ="+start);
  // get as many individuals as specified by subset tournament size
  for (int i = start; i < (start + tournamentSize); i++) {
    sub_set[i-start]= population_array[i];
  }
  println(sub_set);
  Arrays.sort(sub_set);
  //// return the best two
  parents[0] = (Individual) sub_set[tournamentSize-1];
  parents[1] = (Individual) sub_set[tournamentSize-2];
  println("Tournament Selection done");
  //_______________________________________________________________
  Individual a = parents[0];//Selection
  Individual b = parents[1];//Selection
  Individual x = breed(a,b);//Crosover & Mutation
```

The implementation of this selection method improves the algorithm performance, but, nevertheless, the experiment was done, as a first step,implementing the new method without varying the fitness function.

With the experiment finished it only proved to increment the fitness an 0,98 %.

On the other hand, when Top Scaling selection was implemented (see Appendix code), the fitness decreases a 1%, due to the variance decrease that it uses to produce.

Tournament selection will be the method used.

Next, they will be done some considerations about the improvement of the fitness function.

Multi-objective optimization. Pareto Optimal.

As the fitness function should improve after choosing the more effective selection method, a serial of extra experiments of its definition will be done.

The first step will be to implement Pareto frontier. Considering necessary the optimization of two values, Q, the thermal flux, and t_f, the final temperature, it will be implemented a multiobjective optimization.

Pareto frontier

Objective A: Thermal flux minimum
Objective B: t_f as close as possible to 22ºC

Relative Weights: Wa Wb

Fitness function:

f(x)= 1/ (1+ Wa*A + Wb*B)

Weights will be in relation to the importance of the objective, considering the obtainance of the desired temperature fundamental, and Q of relative importance as the flux will not condition extremely normal insulation situations.

Wa =0,2
Wb =0,8

The results obtained in this final experiment showed a maximum fitness of 0,016, optimizing the maximum insulation properties of the material and ETFE pillows. An homogeneous controlled decrease of temperatures has taken place in all the façade as desired in the inputs of the programme.

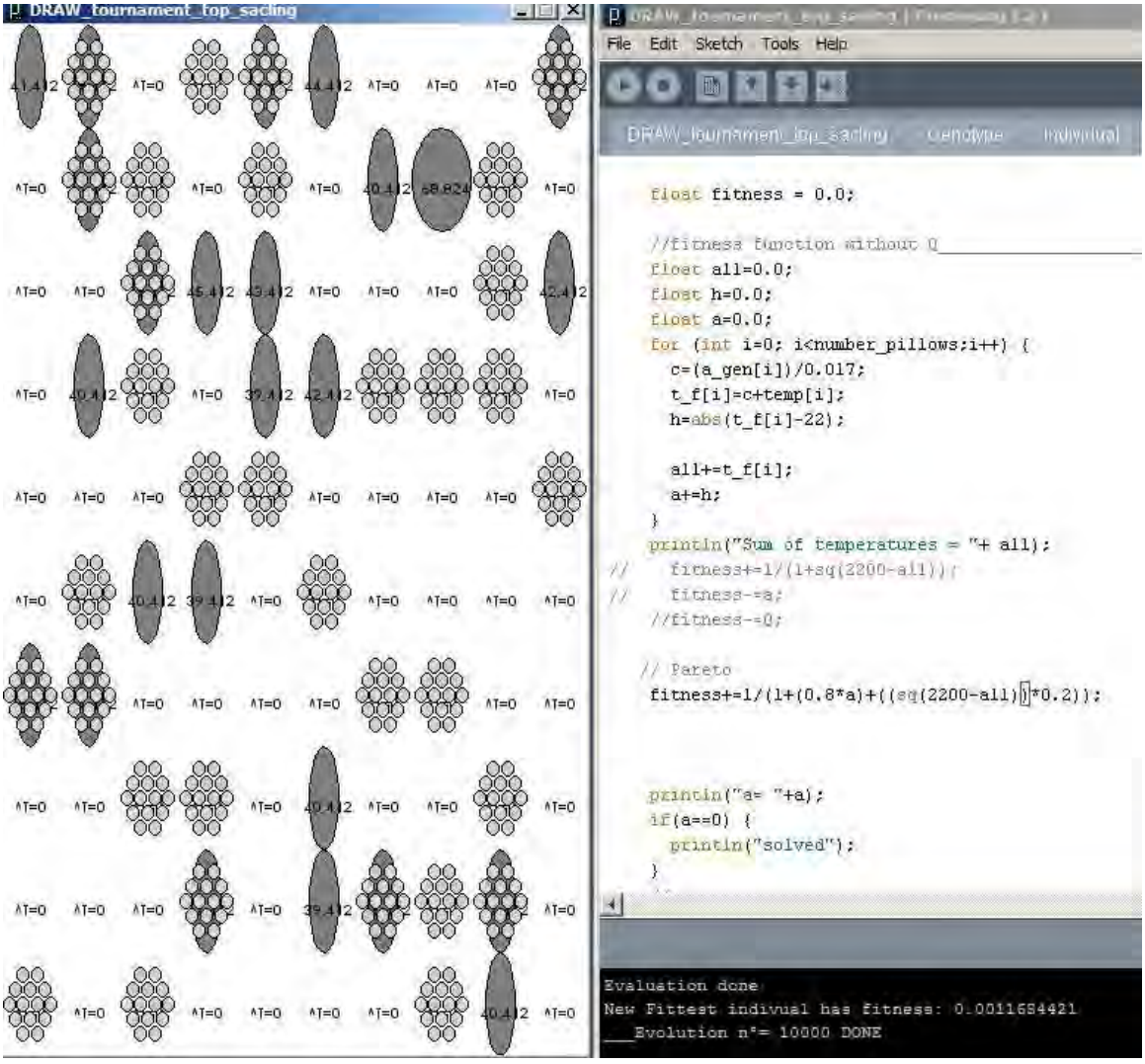


Figure 30

Due to the thermal properties realationship implemented,

t_f [i]= Q* genes[i]/0.017 + Temp [i]

and to the contrains on the degree of opening of the pillows, and that is, thickness of the pillow,

genes [i] = 0, genes[i] = 0.5 or genes [i] = 1

this is the maximum fitness that can be reached.

See the next examples with different input temperatures, and a t_f always 22ºC, that are really self-explanatory:

Example 1, Temp = 20 ºC

22-20= genes[i] *0.017 ⟶ so genes [i] = 0,034, degree of opening not allowed at the code.

Example 2, Temp = 15 ºC

22-15= genes[i] *0.017 ⟶ so genes [i] = 0,119, degree of opening not allowed at the code.

Example 3, Temp = 30 ºC

genes [i] = 0,034, degree of opening not allowed at the code.

These example demonstrates that with a constrained opening degree to three unique posistions, the fitness obtained by the code is the maximum achievable.

Nevertheless, If we consider a free opening degree of possibilities, from thickness 0 to thickness 1 meter of the pillowsection, so,

genes[i]=random (0,1)

all floats between 0 and 1 will be allowed as possible degrees of opening.

For demonstrating that idea, a different code with a non constrained genes array of opening was run.

After a certain time working, the code reached the maximum fitness, a fitness of 1, demonstrating that the possibility of optimizing insulation of the ETFE is real, was reached.

The GA

```
float mutation_rate=0.05;
intnumber_of_individuals=100;
intnumber_evolutions=1000000;//make it 0 to check the initial po
int ideal_temp=22; //desired temperature in each pillow
PFont font;
PFont font1;
Population population_new;
void setup() {
size(400,700);
input_temperatures();
font1=loadFont("CenturyGothic-Bold-10.vlw");
population_new = new Population();//imprime el mejor de la pobl
//noLoop();
}
void draw() {
int evolution_num=0;
if (mousePressed==true) {
for (int i=0; i<number_evolutions; i++) {
population_new.evolve();
evolution_num++;
println("___Evolution nº= "+evolution_num +" DONE");
}
}
}
```

```
float Q;//Thermal flux
float c;
float [] Performance;
int [] random_sun_shade;
float []new_temp;
float[][] input_pillow;
int [][] sun_shadow;
float [][] print_temp;
int half_wide=20;
int half_heigh=35;
int x_coord;
int y_coord;

class Phenotype
{
byte [] temperatures; / / T are the input temperatures
float [] t_f; / / t f are the output temperatures
float [] temp;
int sum;
float [] a_gen;
float a;
Phenotype(Genotype g) {
temperatures = new byte [number_pillows];
temperatures= loadBytes("temperatures.dat");
```

```
temp= new float [number_pillows];
t_f= new float [number_pillows];
a_gen= new float [number_pillows];
//int sum=0;
int pillow_number=0;
for (int i=0; i<number_pillows;i++) {
temp[i]=float(temperatures[i]);//transforms byte into float
t_f[i]=temp[i]+((Q*g.m_genes[i])/0.017);
//sum++;
pillow_number++;
a_gen[i]=g.m_genes[i];
}
}
void draw() {
background(255);
textFont(font1);
//Draw the façade and perform the local sun-shade behaviour__
Performance= new float [number_pillows];
random_sun_shade=new int [number_pillows];
new_temp=new float [number_pillows];
for(int i=0;i<number_pillows;i++) {
Performance[i]=a_gen[i];
new_temp[i]=t_f[i];
random_sun_shade[i]=int (random(-1,2));
}
input_pillow=new float [10][10];
sun_shadow=new int [10][10];
print_temp=new float [10][10];
for(int i=0; i<10;i++) {
for(int j=0;j<10;j++) {
input_pillow[i][j]=Performance[i+(j*10)];//Gives an input
sun_shadow[i][j]=random_sun_shade[i+(j*10)];
print_temp[i][j]=new_temp[i+(j*10)];
}
}
//draw the perfomance of pillows_____________________________
for(int i=0; i<10;i++) {
for(int j=0;j<10;j++) {
float P= input_pillow[i][j];
intsun_shade=sun_shadow[i][j];
if(P==1) {
//open
x_coord=half_wide+(40*i);
y_coord=half_heigh+(70*j);
fill(126);
ellipse(x_coord,y_coord,40,70);
fill(0);
text(print_temp[i][j],x_coord-15,y_coord+10);
}
if(P==0) {
//do not move
x_coord=half_wide+(40*i);
y_coord=half_heigh+(70*j);
```

```
fill(0);
text("^T=0",x_coord-10,y_coord+10);
}
if( P==0.5) {
//half open
x_coord=half_wide+(40*i);
y_coord=half_heigh+(70*j);
fill(126);
ellipse(x_coord,y_coord,20,70);
fill(0);
text(print_temp[i][j],x_coord-15,y_coord+10);
}
if(sun_shade>0) {
//open shade pattern
x_coord=half_wide+(40*i);
y_coord=half_heigh+(70*j);
fill(209);
ellipse(x_coord-5,y_coord,8,10);
ellipse(x_coord+5,y_coord,8,10);
ellipse(x_coord-14,y_coord,8,10);
ellipse(x_coord+14,y_coord,8,10);
ellipse(x_coord,y_coord+10,8,10);
ellipse(x_coord,y_coord-10,8,10);
ellipse(x_coord-10,y_coord-10,8,10);
ellipse(x_coord+10,y_coord-10,8,10);
ellipse(x_coord-10,y_coord+10,8,10);
ellipse(x_coord+10,y_coord+10,8,10);
ellipse(x_coord-5,y_coord+20,8,10);
ellipse(x_coord+5,y_coord+20,8,10);
ellipse(x_coord-5,y_coord-20,8,10);
ellipse(x_coord+5,y_coord-20,8,10);
}
}
}
}
float fitness_function() {
float fitness = 0.0;
//fitness function without Q_____________________________
float all=0.0;
float h=0.0;
float a=0.0;
for (int i=0; i<number_pillows;i++) {
c=(a_gen[i])/0.017;
t_f[i]=c+temp[i];
h=abs(t_f[i]-22);
all+=t_f[i];
a+=h;
}
println("Sum of temperatures = "+ all);
//fitness-=Q; It will be not finally considered due to its no
// Pareto
fitness+=1/(1+(0.8*a)+((sq(2200-all))*0.2));
//Variables weighted according to importance
```

```
println("a= "+a);
if(a==0) {
println("solved");
}
//_____________________________________________________
return fitness;
}
}
```

```
class Genotype {

float [] m_genes; / /my m_genes is [100] because of 100 pillow
float [] opening_possibilities;
Genotype() {
m_genes = new float [number_pillows];
opening_possibilities=new float[3];
opening_possibilities[0]=0.0;//cerrado el espesor es despreci
opening_possibilities[1]=0.5;//medio abierto 50 cm
opening_possibilities[2]=1.0;//abierto 50+50 cm , osea 1 metr
//esto se implementa como genes en el fenotipo
for (int i = 0; i < m_genes.length; i++) {
m_genes[i] = opening_possibilities[int(random(0,3))];//open
}
//println(m_genes);// prints the genes array of the 100 indiv
}
void mutate() {
for (int i = 0; i < m_genes.length; i++) {
if (random(0,1) < mutation_rate) {
m_genes[i] = opening_possibilities[int(random(0,3))];
}
}
}
}
Genotype crossover(Genotype a, Genotype b) {
Genotype c = new Genotype();
for (int i = 0; i < c.m_genes.length; i++) {
if (random(0,1) < 0.5) {
c.m_genes[i] = a.m_genes[i];
}
else {
c.m_genes[i] = b.m_genes[i];
}
}
return c;//new individual with half of the genes of each of the
}
```

```
class Individual implements Comparable {
```

```
Genotype m_genotype;
Phenotype m_phenotype;
float m_fitness;
Individual() {
m_genotype = new Genotype();
m_phenotype = new Phenotype(m_genotype);
m_fitness = 0.0;
}
void draw() {
background(255);
m_phenotype.draw();
}
float evaluate() {
m_fitness = m_phenotype.fitness_function();
println("Individual fitness is "+m_phenotype.fitness_functio
println("Evaluation done");
return m_fitness;
}
int compareTo(Object o) {
Individual other=(Individual)o;
if (m_fitness > other.m_fitness) {
return 1;
}
else if (m_fitness < other.m_fitness) {
return -1;
}
return 0;
}
}
Individual breed(Individual a, Individual b) {
Individual c = new Individual();
c.m_genotype = crossover(a.m_genotype,b.m_genotype);
c.m_genotype.mutate();
c.m_phenotype = newPhenotype(c.m_genotype);
return c;
}
```

```
class Population {

Individual [] population_array;
float[]fitness_of_popul_individuals;
Population() {
population_array = new Individual [number_of_individuals];
fitness_of_popul_individuals= new float[number_of_individual
for (int i = 0; i <population_array.length; i++) {
Individual ind = new Individual();
ind.evaluate();
population_array[i]=ind;
fitness_of_popul_individuals[i]= population_array[i].evalua
```

```
}
fitness_of_popul_individuals= sort(fitness_of_popul_individu
Arrays.sort(population_array);
population_array[99].draw();//Draws the fittest individual of
println("FIRST POPULATION CREATED");
println("___Population fitness Array reordered ");
println(fitness_of_popul_individuals);
println("Fittest indivual fitness: "+fitness_of_popul_indivi
}
void evolve() {//creates new individual in the population loca
for (int indi =0;indi<3;indi++) {
//Tournament Selection Method_____________________________
int tournamentSize=5;
Individual [] parents;
parents= new Individual [2];
Individual [] sub_set;
sub_set = new Individual [tournamentSize];
// pick a random place to start within the population
int start=0;
start = int(random(-1,(number_pillows-5)));
println("start point for subset is ="+start);
// get as many individuals as specified by subset tournamen
for (int i = start; i < (start + tournamentSize); i+
sub_set[i-start]= population_array[i];
}
println(sub_set);
Arrays.sort(sub_set);
//// return the best two
parents[0] = (Individual) sub_set[tournamentSize-1];
parents[1] = (Individual) sub_set[tournamentSize-2];
println("Tournament Selection done");
Individual a = parents[0];//Selection
Individual b = parents[1];//Selection
Individual x = breed(a,b);//Crosover & Mutation
println(" ");
println("___Population has evolved. New individual added");
fitness_of_popul_individuals[0]= x.evaluate();
fitness_of_popul_individuals= sort(fitness_of_popul_indivi
//println("New fitness [] =" +fitness_of_popul_individuals)
x.evaluate();
population_array[0]=x;
Arrays.sort(population_array);
//_____________________________________________________________
println("New Fittest indivual has : "+ population_array[99]
}
population_array[99].draw();//Draws the new fittest individua
}
}
```

```
//Input temperatures creation_____________________________________

int number_pillows=100;
float [] temperatures; / / T are the input temperatures, simulat
void input_temperatures() {
temperatures = new float [number_pillows];
for (int i=0; i<number_pillows;i++) {
temperatures[i]=random(10,20);//this is the scenario set
}
byte[] temp=byte(temperatures);
//println(temperatures);
saveBytes("temperatures.dat",temp);
println("input_temperatures saved");
}
```

```
// //Top scaling_____________________________________________________
// //assigns to 40% of the fittest individuals to the same sc
// //assigns the rest of the individuals to fitness value 0.
// float N=0.2;
// float scaled_fitness;
// scaled_fitness=number_pillows*N;
//
// for(int i=61;i<100;i++) {
// fitness_of_popul_individuals[i]=scaled_fitness;
// }
// for(int i=0;i<61;i++) {
// fitness_of_popul_individuals[i]=0;
// }
// Arrays.sort(population_array);
// Individual a = population_array[int(random(60,99))];//Selec
// Individual b = population_array[int(random(60,99))];//Selec
// Individual x = breed(a,b);//Crosover & Mutation
//
// println(" ");
// println("___Population has evolved. New individual added");
//
// fitness_of_popul_individuals[0]= x.evaluate();
// fitness_of_popul_individuals= sort(fitness_of_popul_individ
//
// println("New Fittest indivual has fitness: "+ fitness_of_pop
//
// population_array[0]=x;
// Arrays.sort(population_array);
// //_____________________________________________________________
```

Complete code implementation

Reached the final point of the GA experimentation the code of the full system developed in this thesis was joined together.

The Artificial Neural Network, designed for the Mediterranean Weather Scenario in this thesis, was completed with the General Algorithm code that will be chosen for optimizing the façade insulation performance in that scenario.

The ANN when completed with all weather scenarios cases, and after learning and deciding, will choose which GA to develop for the façade thermal performance.

In the case of this thesis, it was only developed the GA for the climate case is being considered but, in the complete software design, all the other GAs should be included in the software for being run depending on the ANN's decision about the particular scenario it consider the building is at that particular time.

EmDeplo's brain

```
//ANN Variables
int num_of_neurons=100;//Number of pillows of the façade
intnum_hidden_neurons=225;
int num_train_sets=10000;//sets of labeled [] of Tº for trainning
int num_of_outputs=10;//Number of Scenarios for that climate
int facade_size=100;//Number of pillows of the façade
int label;
int train_pics=500;//Number of sets used in each trainning round
int num_train=100;
int set_sc_size=1000;
int number_scenarios=10;
int size_train_matrix=int(facade_size*set_sc_size);
int size_final_train_set=int(size_train_matrix*number_scenarios)
intlabel_size_array=10000;
Train new_train_set;
Labels new_label_set;
Network neuralnet;
//GVAa riables_____________________________________________________
float mutation_rate=0.05;
intnumber_of_individuals=100;
int number_evolutions=100;//make it 0 to check the initial popula
int ideal_temp=22; //desired temperature in each pillow
PFont font;
PFont font1;
Population population_new;
void setup() {
```

```
new_train_set=new Train();
new_train_set.create_trainning_set();
println("Trainning Set file Saved");
new_label_set=new Labels();
new_label_set.create_labels();
println("Labels file Saved");
size(400,700);
DATALOAD();
println("_ALL DATA LOADED");
setupSigmoid();
neuralnet= newNetwork(num_of_neurons,num_hidden_neurons,num_of
for(int i=0;i<train_pics;i++) {
int row= (int) floor(random(0,trainSet.length));
neuralnet.respond (trainSet[row].inputs);
neuralnet.train(trainSet[row].outputs);
}
println("_FIRST TRAIN FINISHED");
}
void draw()
{
//To start more trainning
if(keyPressed) {
for(int j=0;j<num_train;j++) {
for(int i=0;i<train_pics;i++) {
int row= (int) floor(random(0,trainSet.length));
neuralnet.respond (trainSet[row].inputs);
neuralnet.train(trainSet[row].outputs);
}
}
println("_NEW 10 TRAININGs FINISHED");
}
}
//this is for the test set_____________________________________________
void mousePressed()
{
//set the testset float[]
if (mousePressed) {
println("_TEST INPUTS LOADED");
data_test();
}
}
//_____________________________________________________________________


intimagesArrayLenght=(num_train_sets*facade_size);
int []labels= new int[num_train_sets];
byte [] trainning_array;
byte []labels_byte;
```

```
float [] mediterranean_climate;//T° [] with values 0-45 degrees
float [] temper= new float [num_of_neurons];
//_____________________________________________________________
class Data
{
float [] inputs;
float [] outputs;
float [] inputs_test;
int label_output;
Data()
{
inputs=new float [num_of_neurons];
outputs= new float[num_of_outputs];
}
//___load the images & create the inputs[] OF FLOATS___________
void LoadTemperatures(byte [] trainning_array, int facade_size)
for(int i=0; i<num_of_neurons; i++) {
inputs[i]= int(trainning_array[i+facade_size])/23-1.0;//be
}
}
//___load the labels & create the outputs [] OF FLOATS_________
//there are 10 different labels but we scale the outputs to ye
void LoadLabels (int [] labels, int offset)
{
label_output= int(labels[offset]);
for (int i=0; i<num_of_outputs;i++) {
if(i==label_output) {
outputs[i]=1.0;
}
else {
outputs[i]=-1.0;
}
}
}
//create the inputs for testing_____________________________________
float [] define_test_inputs( byte [] trainning_array, int facad
inputs_test=new float [num_of_neurons];
for (int i = 0; i < num_of_neurons; i++) {
inputs_test[i] = int(trainning_array[i+facade_size])/ 23.0
float a =(inputs_test[i]+1)*23;
temper[i]=a;
}
return temper;
}
}
//end of the class_____________________________________________________
```

```
//FUNCTION TO LOAD THE DATA(we call it in the main tab)__________
Data [] trainSet;
void DATALOAD() {
//____Load trainning []s_____________________________________
trainning_array =loadBytes("trainning_set.dat");
println("_Trainning data T°[]s LOADED");
//____Load labels( 10 scenarios)_____________________
labels_byte=loadBytes("labels.dat");
for(int i=0;i<labels_byte.length;i++) {
labels[i] = labels_byte [i];
}
println("_LABELS LOADED");
//set the trainSet float[]_____________________________
trainSet=new Data [int(num_train_sets/5*4)];
int trainPos=0;
int testPos = 0;
testSet=new Data [int(num_train_sets/5)];
for(int i=0;i<num_train_sets;i++) {
if((i % 5 != 0)) {
trainSet [trainPos]= new Data();
trainSet [trainPos].LoadTemperatures(trainning_array, i*fac
trainSet [trainPos].LoadLabels(labels, i);
trainPos++;
}
else {
testSet[testPos] = new Data();
testSet[testPos].define_test_inputs(trainning_array, i * fa
testSet[testPos].LoadLabels(labels, i);
testPos++;
}
}
}
//END OF LOAD DATALOADFUNCTION_____________________________________
```

```
class Labels {
byte[]labels_set;
Labels() {
```

```
}
void create_labels() {
labels_set=new byte[label_size_array];
for (int i=0;i<label_size_array; i++) {
if (i<1000) {
labels_set[i]=0;
}
if ((i>=1000) && (i<2000)) {
labels_set[i]=1;
}
if ((i>=2000)&&(i<3000)) {
labels_set[i]=2;
}
if ((i>=3000)&&(i<4000)) {
labels_set[i]=3;
}
if ((i>=4000)&&(i<5000)) {
labels_set[i]=4;
}
if ((i>=5000)&&(i<6000)) {
labels_set[i]=5;
}
if ((i>=6000)&&(i<7000)) {
labels_set[i]=6;
}
if ((i>=7000)&&(i<8000)) {
labels_set[i]=7;
}
if ((i>=8000)&&(i<9000)) {
labels_set[i]=8;
}
if ((i>=9000)&&(i<10000)) {
labels_set[i]=9;
}
}
println("Label created. []size= " + labels_set.length);
saveBytes("labels.dat", labels_set);
}
                                                                    }


//int size_train_matrix=int(facade_size*set_sc_size);
//ints ize_final_train_set=int(size_train_matrix*number_scenarios
byte [] Final_trainning_array;
//this is the complete trainning set array
float [] set_of_temperatures;
float [] trainning_array_0;
```

```
float [] trainning_array_1;
float [] trainning_array_2;
float [] trainning_array_3;
float [] trainning_array_4;
float [] trainning_array_5;
float [] trainning_array_6;
float [] trainning_array_7;
float [] trainning_array_8;
float [] trainning_array_9;
byte[] b_0;
byte[] b_1;
byte[] b_2;
byte[] b_3;
byte[] b_4;
byte[] b_5;
byte[] b_6;
byte[] b_7;
byte[] b_8;
byte[] b_9;
byte[]big_array;
byte[]big_array_0;
byte[]big_array_1;
byte[]big_array_2;
byte[]big_array_3;
byte[]parcial_array_0;
byte[]parcial_array_1;
byte[]parcial_array_a;
class Train {
Train() {
}
void create_trainning_set() {
//"mix" situation of the façade result of a hot weather but w
//para clima mediterraneo, matriz general=[0,45]
//Scenario0 ____________________________________________________
//summer 1. [hot]
trainning_array_0 = new float[int(size_train_matrix)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_0[i]=random(25,35);
}
byte[] b_0=byte(trainning_array_0);
//Scenario1 ____________________________________________________
//summer 2. [mix]
trainning_array_1 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_1[i] =random(20,35);
}
byte[] b_1=byte(trainning_array_1);
//Scenario2 ____________________________________________________
```

```
//summer 3. [summer "cold"] = random(20,25)
trainning_array_2 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_2[i] =random(20,25);
}
byte[] b_2=byte(trainning_array_2);
//Scenario3 ____________________________________________
//spring 1. [hot] =random (15, 25)
trainning_array_3 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_3[i] =random(15,25);
}
byte[] b_3=byte(trainning_array_3);
//Scenari4o_ ___________________________________________
//spring 2. [not so hot]=random(10,20)
trainning_array_4 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_4[i] =random(10,20);
}
byte[] b_4=byte(trainning_array_4);
//Scenari5o_ ___________________________________________
//spring 3. [cold]=random(0,15)
trainning_array_5 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_5[i] =random(0,15);
}
byte[] b_5=byte(trainning_array_5);
//Scenari6o_ ___________________________________________
//spring 4. [mix]=random(0,15)+ random (10,20)
trainning_array_6 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_6[i] =random(0,20);
}
byte[] b_6=byte(trainning_array_6);
//Scenari7o_ ___________________________________________
//winter 1. [cold]=random (0,10)
trainning_array_7 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_7[i] =random(0,10);
}
byte[] b_7=byte(trainning_array_7);
// //Scenario 8____________________________________________
// //winter 2. [warm]=random (5,15)
trainning_array_8 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_8[i] =random(5,15);
}
byte[] b_8=byte(trainning_array_8);
```

```
//Scenario9 ____________________________________________________
//winter 3. [hot]=random(10,20)
trainning_array_9 = new float[int(facade_size*set_sc_size)];
for (int i=0; i<int(facade_size*set_sc_size); i++) {
trainning_array_9[i] =random(15,20);
}
byte[] b_9=byte(trainning_array_9);
//Creation of the complete trainning set array_____________
big_array=concat(b_0, b_1);
big_array_0=concat(b_2,b_3);
big_array_1=concat(b_4, b_5);
big_array_2=concat(b_6, b_7);
big_array_3=concat(b_8,b_9);
parcial_array_0=concat(big_array,big_array_0);
parcial_array_1=concat(big_array_1,big_array_2);
parcial_array_a=concat(parcial_array_0,parcial_array_1);
Final_trainning_array=new byte [1000000];
//this is the complete trainning set array
for(int i=0;i<1000000;i++) {
if (i<800000) {
Final_trainning_array[i]= parcial_array_a[i];
}
if (i>=800000) {
Final_trainning_array[i]=big_array_3[i-800000];
}
}
//println(parcial_array_a.length);
println("Trainning Set created. []size= " + Final_trainning_a
saveBytes("trainning_set.dat", Final_trainning_array);
}
}
```

```
float LEARNING_RATE = 0.01;
class Neuron
{
Neuron [] m_inputs;
float [] m_weights;
float m_threshold;
float m_output;
float m_error;
// the input layer of neurons have no inputs:
Neuron()
{
m_threshold = 0.0;
m_error = 0.0;
// initial random output
```

```
m_output = lookupSigmoid(random(-5.0,5.0));
}
// all other layers (hidden and output) have
// neural inputs
Neuron(Neuron [] inputs)
{
m_inputs = new Neuron [inputs.length];
m_weights = new float [inputs.length];
for (int i = 0; i < inputs.length; i++) {
m_inputs[i] = inputs[i];
m_weights[i] = random(-1.0,1.0);//here I enter the initial
}
m_threshold = random(-1.0,1.0);
m_error = 0.0;
// initial random output
m_output = lookupSigmoid(random(-5.0,5.0));
}
// respond looks at the layer below, and prepares a response:
float respond()
{
float input = 0.0;
for (int i = 0; i < m_inputs.length; i++) {
input += m_inputs[i].m_output * m_weights[i];
}
m_output = lookupSigmoid(input + m_threshold);
// reset our error value ready for training
m_error = 0.0;
return m_output;
}
// find error is used on the output neurons
void finderror(float desired)
{
m_error = desired - m_output;
}
// train adjusts the weights by comparing actual output to corr
void train()
{
float delta = (1.0 - m_output) * (1.0 + m_output) * m_error *
for (int i = 0; i < m_inputs.length; i++) {
// tell the next layer down what it's doing wrong
m_inputs[i].m_error += m_weights[i] * m_error;
// correct our weights
m_weights[i] += m_inputs[i].m_output * delta;
}
}
}
```

```
class Network
{
Neuron [] m_input_layer;
Neuron [] m_hidden_layer;
Neuron [] m_output_layer;
Network(int inputs, int hidden, int outputs)
{
m_input_layer = new Neuron [inputs];
m_hidden_layer = new Neuron [hidden];
m_output_layer = new Neuron [outputs];
// set up the network topology
for (int i = 0; i < m_input_layer.length; i++) {
m_input_layer[i] = new Neuron();
}
// route the input layer to the hidden layer
for (int j = 0; j < m_hidden_layer.length; j++) {
m_hidden_layer[j] = new Neuron(m_input_layer);
}
// route the hidden layer to the output layer
for (int k = 0; k < m_output_layer.length; k++) {
m_output_layer[k] = newNeuron(m_hidden_layer);
}
}
int respond(float [] inputs)// for TRAINNING
{
float [] responses = new float [m_output_layer.length];
for (int i = 0; i < m_input_layer.length; i++) {
m_input_layer[i].m_output = inputs[i];
}
for (int j = 0; j < m_hidden_layer.length; j++) {
m_hidden_layer[j].respond();
}
// and finally feed forward to the output layer
for (int k = 0; k < m_output_layer.length; k++) {
responses[k] = m_output_layer[k].respond();
}
// now check the best response:
int response = -1;
float best = max(responses);
for (int a = 0; a < responses.length; a++) {
if (responses[a] == best) {
response = a;
}
}
```

```
return response;
}
void train(float [] outputs)
{
// adjust the output layer
for (int k = 0; k < m_output_layer.length; k++) {
m_output_layer[k].finderror(outputs[k]);
m_output_layer[k].train();
}
// propagate back to the hidden layer
for (int j = 0; j < m_hidden_layer.length; j++) {
m_hidden_layer[j].train();
}
}
}
//end of network class_____________________________________________
```

```
float [] g_sigmoid = new float [200];
void setupSigmoid() {
for (int i = 0; i < 200; i++) {
float x = (i / 20.0) - 5.0;
g_sigmoid[i] = 2.0 / (1.0 + exp(-2.0 * x)) - 1.0;
}
}
float lookupSigmoid(float x) {
return g_sigmoid[constrain((int) floor((x + 5.0) * 20.0),0,199)
}
```

```
byte[]test_arra;
Data [] testSet;
int number_pillows=100;
float [] temperatures; / / T are the input temperatures, simulat
void data_test() {
int row = (int) floor(random(0,testSet.length));
println("Test Set Number: "+row);
//println("Above is the set of T° tested");
int response=-1;
response = neuralnet.respond(testSet[row].inputs_test);
```

```
int actual=-1;
actual = testSet[row].label_output;
println("Example of scenario number "+actual+",");
println("and");
println("ANN thinks this is scenario "+response);
//DO WHATEVER i WANT IT TO DO FOR ADJUSTING TEMPERATURES
//In further research here more specific ga's should be impleme
GA();
GA_DRAW();
}


void input_temperatures() {
temperatures = new float [number_pillows];
for (int i=0; i<number_pillows;i++) {
temperatures[i]=temper[i];//this is the scenario set
}
byte[] temp=byte(temperatures);
//println(temperatures);
saveBytes("temperatures.dat",temp);
println("input_temperatures saved");
}




input_temperatures();
font1=loadFont("CenturyGothic-Bold-10.vlw");
population_new = new Population();
}
void GA_DRAW() {
int evolution_num=0;
if (mousePressed==true) {
for (int i=0; i<number_evolutions; i++) {
population_new.evolve();
evolution_num++;
println("___Evolution n°= "+evolution_num +" DONE");
}
}
                                                                    }



class Genotype {
float [] m_genes; / /my m_genes is [100] because of 100 pillow
float [] opening_possibilities;
Genotype() {
m_genes = new float [number_pillows];
```

```
opening_possibilities=new float[3];
opening_possibilities[0]=0.0;//cerrado el espesor es despreci
opening_possibilities[1]=0.5;//medio abierto 50 cm
opening_possibilities[2]=1.0;//abierto 50+50 cm , osea 1 metr
//esto se implementa como genes en el fenotipo
for (int i = 0; i < m_genes.length; i++) {
m_genes[i] = opening_possibilities[int(random(0,3))];//open
}
//println(m_genes);// prints the genes array of the 100 indiv
}
void mutate() {
for (int i = 0; i < m_genes.length; i++) {
if (random(0,1) < mutation_rate) {
m_genes[i] = opening_possibilities[int(random(0,3))];
}
}
}
}
Genotype crossover(Genotype a, Genotype b) {
Genotype c = new Genotype();
for (int i = 0; i < c.m_genes.length; i++) {
if (random(0,1) < 0.5) {
c.m_genes[i] = a.m_genes[i];
}
else {
c.m_genes[i] = b.m_genes[i];
}
}
return c;//new individual with half of the genes of each of the
}
float Q;//Thermal flux
float c;
float [] Performance;
int [] random_sun_shade;
float []new_temp;
float[][] input_pillow;
int [][] sun_shadow;
float [][] print_temp;
int half_wide=20;
int half_heigh=35;
int x_coord;
int y_coord;


class Phenotype
{
byte [] temperatures; / / T are the input temperatures
float [] t_f; / / t f are the output temperatures
```

```
float [] temp;
int sum;
float [] a_gen;
float a;
Phenotype(Genotype g) {
temperatures = new byte [number_pillows];
temperatures= loadBytes("temperatures.dat");
temp= new float [number_pillows];
t_f= new float [number_pillows];
a_gen= new float [number_pillows];
//int sum=0;
int pillow_number=0;
for (int i=0; i<number_pillows;i++) {
temp[i]=float(temperatures[i]);//transforms byte into float
t_f[i]=temp[i]+((Q*g.m_genes[i])/0.017);
//sum++;
pillow_number++;
a_gen[i]=g.m_genes[i];
}
}
void draw() {
background(255);
textFont(font1);
//Draw the façade and perform the local sun-shade behaviour__
Performance= new float [number_pillows];
random_sun_shade=new int [number_pillows];
new_temp=new float [number_pillows];
for(int i=0;i<number_pillows;i++) {
Performance[i]=a_gen[i];
new_temp[i]=t_f[i];
random_sun_shade[i]=int (random(-1,2));
}
input_pillow=new float [10][10];
sun_shadow=new int [10][10];
print_temp=new float [10][10];
for(int i=0; i<10;i++) {
for(int j=0;j<10;j++) {
input_pillow[i][j]=Performance[i+(j*10)];//Gives an input
sun_shadow[i][j]=random_sun_shade[i+(j*10)];
print_temp[i][j]=new_temp[i+(j*10)];
}
}
//draw the perfomance of pillows_____________________________
for(int i=0; i<10;i++) {
for(int j=0;j<10;j++) {
float P= input_pillow[i][j];
intsun_shade=sun_shadow[i][j];
if(P==1) {
```

```
//open
x_coord=half_wide+(40*i);
y_coord=half_heigh+(70*j);
fill(126);
ellipse(x_coord,y_coord,40,70);
fill(0);
text(print_temp[i][j],x_coord-15,y_coord+10);
}
if(P==0) {
//do not move
x_coord=half_wide+(40*i);
y_coord=half_heigh+(70*j);
fill(0);
text("^T=0",x_coord-10,y_coord+10);
}
if( P==0.5) {
//half open
x_coord=half_wide+(40*i);
y_coord=half_heigh+(70*j);
fill(126);
ellipse(x_coord,y_coord,20,70);
fill(0);
text(print_temp[i][j],x_coord-15,y_coord+10);
}
if(sun_shade>0) {
//open shade pattern
x_coord=half_wide+(40*i);
y_coord=half_heigh+(70*j);
fill(209);
ellipse(x_coord-5,y_coord,8,10);
ellipse(x_coord+5,y_coord,8,10);
ellipse(x_coord-14,y_coord,8,10);
ellipse(x_coord+14,y_coord,8,10);
ellipse(x_coord,y_coord+10,8,10);
ellipse(x_coord,y_coord-10,8,10);
ellipse(x_coord-10,y_coord-10,8,10);
ellipse(x_coord+10,y_coord-10,8,10);
ellipse(x_coord-10,y_coord+10,8,10);
ellipse(x_coord+10,y_coord+10,8,10);
ellipse(x_coord-5,y_coord+20,8,10);
ellipse(x_coord+5,y_coord+20,8,10);
ellipse(x_coord-5,y_coord-20,8,10);
ellipse(x_coord+5,y_coord-20,8,10);
}
}
}
}
float fitness_function() {
```

```
float fitness = 0.0;
//fitness function without Q_____________________________
float all=0.0;
float h=0.0;
float a=0.0;
for (int i=0; i<number_pillows;i++) {
c=(a_gen[i])/0.017;
t_f[i]=c+temp[i];
h=abs(t_f[i]-22);
all+=t_f[i];
a+=h;
}
println("Sum of temperatures = "+ all);
//fitness-=Q; It will be not finally considered due to its no
// Pareto
fitness+=1/(1+(0.8*a)+((sq(2200-all))*0.2));
//Variables weighted according to importance
println("a= "+a);
if(a==0) {
println("solved");
}
//_____________________________________________________
return fitness;
}
```

```
class Individual implements Comparable {
Genotype m_genotype;
Phenotype m_phenotype;
float m_fitness;
Individual() {
m_genotype = new Genotype();
m_phenotype = new Phenotype(m_genotype);
m_fitness = 0.0;
}
void draw() {
background(255);
m_phenotype.draw();
}
float evaluate() {
m_fitness = m_phenotype.fitness_function();
println("Individual fitness is "+m_phenotype.fitness_functio
println("Evaluation done");
return m_fitness;
}
int compareTo(Object o) {
Individual other=(Individual)o;
```

```
if (m_fitness > other.m_fitness) {
return 1;
}
else if (m_fitness < other.m_fitness) {
return -1;
}
return 0;
}
}
Individual breed(Individual a, Individual b) {
Individual c = new Individual();
c.m_genotype = crossover(a.m_genotype,b.m_genotype);
c.m_genotype.mutate();
c.m_phenotype = newPhenotype(c.m_genotype);
return c;
}
```

```
class Population {
Individual [] population_array;
float[]fitness_of_popul_individuals;
Population() {
population_array = new Individual [number_of_individuals];
fitness_of_popul_individuals= new float[number_of_individual
for (int i = 0; i <population_array.length; i++) {
Individual ind = new Individual();
ind.evaluate();
population_array[i]=ind;
fitness_of_popul_individuals[i]= population_array[i].evalua
}
fitness_of_popul_individuals= sort(fitness_of_popul_individu
Arrays.sort(population_array);
population_array[99].draw();//Draws the fittest individual of
println("FIRST POPULATION CREATED");
println("___Population fitness Array reordered ");
println(fitness_of_popul_individuals);
println("Fittest indivual fitness: "+fitness_of_popul_indivi
}
void evolve() {//creates new individual in the population loca
for (int indi =0;indi<3;indi++) {
//Tournament Selection Method_____________________________
int tournamentSize=5;
Individual [] parents;
```

```
parents= new Individual [2];
Individual [] sub_set;
sub_set = new Individual [tournamentSize];
// pick a random place to start within the population
int start=0;
start = int(random(-1,(number_pillows-5)));
println("start point for subset is ="+start);
// get as many individuals as specified by subset tournamen
for (int i = start; i < (start + tournamentSize); i+
sub_set[i-start]= population_array[i];
}
println(sub_set);
Arrays.sort(sub_set);
//// return the best two
parents[0] = (Individual) sub_set[tournamentSize-1];
parents[1] = (Individual) sub_set[tournamentSize-2];
println("Tournament Selection done");
Individual a = parents[0];//Selection
Individual b = parents[1];//Selection
Individual x = breed(a,b);//Crosover & Mutation
println(" ");
println("___Population has evolved. New individual added");
fitness_of_popul_individuals[0]= x.evaluate();
fitness_of_popul_individuals= sort(fitness_of_popul_indivi
//println("New fitness [] =" +fitness_of_popul_individuals)
x.evaluate();
population_array[0]=x;
Arrays.sort(population_array);
//_____________________________________________________
println("New Fittest indivual has : "+ population_array[99]
}
population_array[99].draw();//Draws the new fittest individua
}
}
```

6 . Conclusions & further questions

Conclusions

This thesis demonstrates the advantages of having a global + local behaviour versus a stimulus-reaction performance on the EmDeplo ETFE façade.

The results obtained show that a perfect optimization through a specific GA for each scenario is maximizing the insulation performance of the pillows of the system.

After implementing the thermal relationship of the material in the façade performance, a complete process through the ANN and the GA, is carried out for each scenario possible and, particularly focused on the one chosen by the ANN, obtaining, a much more efficient thermal performance.

The different façade patterns created by the learning of the system through the different scenarios and experience , plus the local behaviour for shading, is more effective nevertheless when a non-constrained opening of the pillows is allowed.

On the contrary to the media-Tic building, the EmDeplo system is working with the environment, and not defending against it, showing a really good capacity to adapt, and, because of that, a really effective relationship with the environment, generating, a high decrease on the cost of climate systems for interior spaces.

EmDeplo's brain has been designed as a complex system, non-divisible, and non-reducible in which, the only way to know its real behaviour, is to run the system as a whole, being the behaviour of it, the sum of the behaviour of its parts studied separately.

Further Research proposed

A further research should be proposed at this point based on completing the whole software for all the weather scenarios possible, introducing all the GAs.

Also for a complete system it must be considered the suppression or not of the first step of the software customization in factory. In that way a whole scenario customization should be able to be done on arrival to site.

Another point that must be taken into account for further research must be a strict thermal calculus by a proper environmental software. This calculus should be carried out for a specialized engineering office not assuming flat surfaces, as in this thesis for simplification, but taking into account the different curvature degrees of the pillows in each phase of the opening-closing procedures.

For example, if considering a really thin material, dr, and a radius r, supposing a cylindrical curvature of the pillows,

$\delta Q = - \lambda * S * dt/dx = - \lambda\ 2\pi r^* L * dt/dr$ (λ=material conductivity)

$S = 2\pi r. L \qquad dx = dr$

$\varphi\ dr/r = - \lambda\ 2\pi L\ dt$

$$\varphi \int_{r_1}^{r_2} dr/r = - \lambda\ 2\pi L \int_{t_1}^{t_2} d\tau \qquad \text{so} \qquad \varphi\ \ln r_{2/}\ r_1 = - \lambda\ 2\pi L\ (t_2 - t_1)$$

The thermal resistance considered should be something similar to,

$\rho = 1 / 2\pi L \lambda * \ln r_{2/}\ r_1$

It will need also to be necessarily done a simulation with several number of rooms/spaces with different desired temperatures and timing.

This bigger research, combined with, rigorous physical Arduino controlled model, will be the last step for completing definitely the argument this thesis is trying to demonstrate.

Reinforced Learning

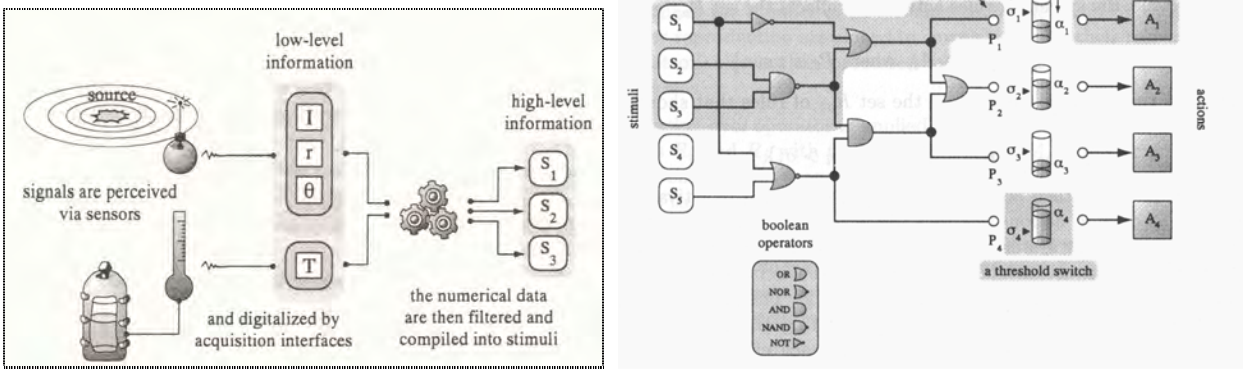
Reinforced learning is a machine learning method , as mentioned in previous chapters, that uses agents, autonomous, for adaptive learning and perception.

Considering the performance of the façade or the membrane as the performance of a set of autonomous agents improving its behaviour and adaptability in reaching their goal by practise and experience, it is a really good basis for a more developed system for EmDeplo's brain.

Options like interpreting the possible behaviour of the system with, for example, a Q-Learning algorithm, look really a good approach for future developments.

this agents, each of which, can be considered to be a pillow of the façade, can shape their behaviour according to the environmental context through the learning including a reward function.

The proposed method will work in a diagram like the one proposed by Calderoni and Mancenac at the 12th Artificial Intelligent Conference in their paper "*MUTANT:A genetic learning system*".



(see # 160, *Advanced topics in artificial intelligence*, 1999) p.205

The paper presented and demonstrated, in 1999, the efficacy of the application of Q-Learning to a similar situation that the one this thesis presented. The agent simulation with and encoded GA as the basis of agents behavioural rules are proposed as the initial idea for further development of the intelligent system of EmDeplo.

As a Reinforce Learning Agents, the ETFE pillows, can explore the environment through the sensors and the light meters, find where and what the rewards are, and find the best policy to move to them. At the initial points of the learning, the agents always will return to the initial fixed values decided by the designer. In that way, the environment is unknown at the beginning of the learning, which is exactly the kind of situation EmDeplo Membrane will be in on site's arrival.

Ana Cocho Bermejo
Spring 2012

7. Bibliography & figures

Bibliography

1, Douglis, E. (2009). *Autogenic structures*. New York: Taylor & Francis Group.

2, García Merayo, F. (2009). Los primeros tiempos del sofware. Parte i. *ACTA, Vol.52*.

3, Kwinter, S. (2008). *Far from equilibrium. Essays on technology and design culture*. Barcelona/New York: Actar.

4, Lorenz, E. n. (1993). *The essence of chaos.* Seattle: University of Washinto press.

5, Manovich, L. (2001). *The language of new media*. Cambridge, Massachusetts: MIT Press.

6, MOMA. (2008). *Design and the elastic mind*. New York: The Museum of Modern Art.

7, Silver, M., Balmori, D., The Yale simposium. (2003). *Mapping in the age of the digital media*. London: Wiley Academy.

8, Various. (2006). Techniques and technologies in morphogenetic design. *Architectural Design, 76*.

9, Tierney, T. (2007). *Abstract space.Beneath the media surface*. New York: Taylor & Francis Group.

10, Dictionary., O. E. (2010). Algorithm. First published in a supplement to the oed in 1972., *Oxford English Dictionary. online version November 2010. http://www.oed.com:80/Entry/4959, accessed 19 January 2011.* (Second edition, 1989; online version November 2010. ed.): Oxford University Press.

11, Knuth, D. (1962). *The art of computer programming*.

13, Webdevelopersnotes, (2010). Programming languajes on the internet. from www.webdevelopernotes.com

14, Dictionary, O. E. (2010). Computer. An entry for this word was first included in new english dictionary, 1891., *Oxford English Dictionary. Third edition, August 2010; online version November 2010. http://www.oed.com:80/Entry 37975; accessed 19 January 2011.* : Oxford University Press.

18, Waliczky, T., (1989). The manifesto of computer art. from *http: www.waliczky.com accessed 19 January 2011.*

19, Ball, P. (2001). *The self made tapestry: Pattern formation in nature*: Oxford University Press.

31, Petit, J.-P. (1980). *Le geometricon. Les aventures d'anselme lanturlu.*: Belin.

32, Issa, R. (2010). Essential mathematics for computational design.

38, Eliasson, O. (2007). Models are real. *Models, 306090 books, 11,* 4.

39, Hernandez, V. A. (2009). Hasta donde las matematicas nos lleven. *ACTA, 51*.

40, García Merayo, F. (2009). Los primeros tiempos del software. Parte ii. *ACTA,53*.

41, Dempsey, A. (2010). Aa / fab designing fabrication. *Issues 10. News from the Architectural Association., 10*.

42, Spyropoulos, T. S., Stephen. (2010). Memory cloud. *AA Files, 59*.

43, Furján, H. (2007). Cities of complexity. *Models, 306090 books, 11*.

44, Despommeier, D. E., Eric. (2007). Proof of concept. *Models, 306090 books,11*.

45, Sabin, J. (2007). Body blanket. *Models, 306090 books, 11*.

46, Benjamin, J. (2007). Analog dreams. *Models, 306090 books*.

47, Various. (2007). Models. *Models, 306090 books, 11*.

49, Yerkes, C. Y. (2007). The castle. *Models, 306090 books, 11*.

50, Russo, R. (2001). The sensation of deep & cryptic. *Models, 306090 books, 11*.

51, Cross-catalityc architectures: In conversation. (2007). *Models, 306090 books,11*.

54, Alexander, C. (1965). A city is not a tree.

55, Malé-Alemany, M. (2007). Parametric constructions: An exploration on virtual standarization. *Models, 306090 books, 11*.

57, Ito, T. I., Arata; Sasaki, Matsuro; Steele, Brett. (2007). *Morphogenesis of flux structure*. London: AA Publications.

58, L-Legendre, G. S., John; Wise, Crise. (2006). *Mathematical form. Jonhn pickering and the architecture of the inversion principle.*: AA Publications.

59, Kolarevic, B. (2008). *Manufacturing material effects: Rethinking design and making in architecture.*: Routledge.

60, Meredith, M. A. A. d. A. L. S., Mutsuro. (2007). *From control to design: Parametric/algoritmic architecture.*: Meredith, Michael; Aranda-lasch, Mutsuro Sasaki.

61, Moussave, F. (2010). *The function of form*: Actar 6 Harvard University School of Design.

62, Various. (2004). Emergence: Morphogenetic design strategies. *Architectural Design, 74*.

63, Various. (2008). Versatility and vicissitude: Performance in morpho ecological design *Architectural Design*.

64, Kolarevic, B. (2003). *Architecture in the digital age: Design and manufacturing*. New York: Spoon press.

65, Terzidis, K. (2006). *Algoritmic architecture*. Oxford: Elsevier Ltd.

66, Drexler, E. (1986). *Engines of creation: The coming era of nanotechnology*. New York: Anchor Books.

69, Koolhaas, R. B., Stefano; Kwinter, Sanford; Tazi, Nadia. (2001). *Mutations*: Actar.

70, Steele, B. (2005). *Corporate fields. New office enviroments by the aa drl.* London: AA Publications.

71, AA. (2006). *Projects review 2005/06*. London: AA Publications.

72, Jameson, F. (1991). *Postmodernism, or, the cultural logic of late capitalism*.

73, Feenberg, A. (Spring 2000). Do we need a critical theory of technology? Reply to tyler veak. *Science, Technology, & Human Values, 25*(nº 2), 238-242.

74, Dreyfus, H. D., Stuarte. (1990). Making a mind versus modelling the brain: Artificial intelligence back at a branch-point. In M. A. Boden (Ed.), *The philosophy of artificial intelligence*: Oxford University Press.

75, Weeler, M. (1990). From robots to rotko:The bringing forth of worlds. In M. A. Boden (Ed.), *The philosophy of artificial life* (pp. 209-236): Oxford University Press.

76, Terzidis, K. (2009). *Algorithms for visual design; using the processing language*. Indianapolis: Wiley Publishing Inc.

77, Ortega, L. (2009). *La digitalización toma el mando*: Gustavo Gili.

78, Jacobs, J. (1961). *The death and life of great american cities*: Penguin.

79, Lefebvre, H. (1984). *The production of space*. Malden: Blackwell.

80, Descartes, R. (1968). *Discourse on the method and meditations*. Suffolk: Penguin classics.

81, Brighton, H. S., Howard. (2003). *Introducing artificial intelligence*: Icon Books.

82, Greenberg, I. (2007). *Processing.Creative coding and computational art.*: Friendsoft.

83, Shiffman, D. (2008). *Learning processing. A beginner's guide to programming images, animation and interaction*: Morgan Kaufmann.

84, Simon, H. A. (1996). *The sciences of the artificial* (3rd ed.). Cambridge,Massachusetts: The MIT Press.

85, Cogswell, D. G., Paul. (2006). *Chomsky para principiantes*. New York: ERA Naciente ARL.

86, Boden, M. A. (1990). *The creative mind: Myths and mechanism* (2nd ed.). London.

87, Jaffe, M. J. (1973). Thigmomorphogenesis: The response of plant growth and development to mechanical stimulation. *Planta, Vol 114, No* 2, 143-157.

88, Dictionary, O. E. (2010). Script. Earlier version first published in new english dictionary, 1911. Current definition addition 1993-7, *Oxford English Dictionry* (Second edition, 1989; online version November 2010. <*http: www.oed.com:80/Entry/173567*>; accessed 19 January 2011. ed.): Oxford University Press.

89, Haque, U. (2007). The architectural relevance of gordon pask. *Architectural Design, 77,* 54-61.

90, García, M. (2007). Otherwise engaged new projects in interactive design. *Architectural Design, 77,* 44-53.

91, Thomsen, M. R. (2008). Robotic membranes. Exploring a textile architecture of behaviour. *Architectural Design, 78,* 98-103.

92, Sheil, B. (2008). Protoarchitecture between the analogue and the digital. *Architectural Design, 78,* 7-11.

93, Various. (2010). The new structuralism: Design, engineering and architectural technologies. *Architectural Design, 80*.

94, Various. (1997). *Intelligent hybrid systems. Fuzzy logic, neural networks and genetic algorithms.* Massachussets: Kluwer Academic Publishers.
95, Michalrwicz, z. (1999). *Genetic algorithms + data structures =evolutionary programs*. New York: Springer.
96, Najarian, K. (2001). On learning of sigmoid neural networks. *Complexity*, 39-45.
97, Eveling, S. (2010 (original 1926)). *Space as membrane*. London: AA Publications.
98, Barger, R. N. (2008). *Computer ethics*: Cambridge University Press.
99, Epstein, R. G. (1997). *The case of the killer robot*. new York: John Wiley & Sons Inc.
101, Hopkins, P. D. (2008). A moral vision for transhumanism. *Journal of Evolution & Technology, 19*(1), 3-7.
102, Dyson, G. (2006). Barricelli's universe. *Make, 8*, 190-192.
103, Various. (2010). *Fabricating architecture. Selected readings in digital design and manufacturing.* New York: Princeton Architectural Press.
104, Picon, A. (2010). *Digital culture in architecture. An introduction for design professions.* Basel: Birkhäuser.
105, Picon, A. (2011). *Architecture and subjectivity: Ornament*
106, Negroponte, N. (1975). *Soft architecture machines*. Massachussets: MIT Press.
107, Alexander, C. (167). *Notes on the synthesis of form*. Cambridge, Massachussets: Harvard University Press.
108, Varela;, H. R. M. F. J. (1980). *Autopoiesis and cognition. The realization of the living*. london: D. Reide Publishing Company.
109, Fogel, D. B. (2006). *Evolutionary computation. Toward a new philosophy of machine intelligence*. New Jersey: Wiley-Interscience.
110, Chomsky, N. (1968). *Language and mind* (Third ed.). Cambridge, UK: Cambridge University Press.
111, Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge,USA: MIT Press.
112, Collins, J. M., Bill. (2000). *Introducing derrida*. Cambridge: Icon Books.
113, Wilson, S. (2010). *Art + science. How scientific research and technogical innovation are becoming key to 21st century aesthetics*. London: Thames & Hudson.
114, Deasy, D. (2001). The transhuman age. *Ethical Technology*,
115, Barbour, I. (1990). *Ethics in an age of technology* (Vol. 2). San Francisco: Harper San Francisco.
116, Bainbridge, W. S. (2008). Cognitive expansion technologies. *Journal of Evolution & Technology, 19*(1).
117, Bostrom, N. (2005). A history of transhumanist thought. *Journal of Evolution & Technology*.
118, Hauskeller, M. (2010). Nietzche, the overhuman and the posthuman: A reply

to stefan sorgner. *Journal of Evolution & Technology, 21*(1).
119, Hibbard, B. (2010). Nietzche's overhuman is an ideal whereas posthumans will be real. *Journal of Evolution & Technology, 21*(1).
120, Hickman, J. (2009). Mapping a small moral universe. *Journal of Evolution & Technology, 20*(2).
122, Jordan, G. E. (2008). The invention man: A response to c.S. Lewis's the abolition of man. *Journal of Evolution & Technology, 19*(1).
123, Kenyon, S. H. (2008). Would you still love me if i was a robot? *Journal of Evolution & Technology, 19*(1).
124, McInstosch, D. (2010). The transhuman security dilema. *Journal of Evolution & Technology, 21*(2).
125, More, M. (2010). The overhuman in the transhuman. *Journal of Evolution & Technology, 21*(1).
126, Sorgner, S. L. (2010). Beyond humanism: Reflection on trans- and posthumanism. *Journal of Evolution & Technology, 21*(2).
127, Stambler, I. (2010). Life extension - a conservative enterprise? Some fin-de-siècle and early 20th century precursors of transhumanism. *Journal of Evolution & Technology, 21*(1).
128, Verdoux, P. (2009). Transhumanism, progress and the future. *Journal of Evolution & Technology, 20*(2).
129, Armen, M. A. S. L. A. C. (2006). An approach to computing ethics. *Machine ethics, July/August*,
132, Wheeler, M. From robots to rothko: The bringing forth of the worlds. In M. Boden (Ed.), *The philosophy of artificial life* Oxford: Oxford University Press.
133, Zoghbi, C. B. D. K. E. C. S. (2008). Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *Springerlink*,
134, Umpleby, S. A. (2001). What comes after second order cybernetics. *Journal of Evolution & Technology*.
135, Joslyn, F. H. C. (2001). Cybernetics and second-order cybernetics. In R. A. Meyers (Ed.), *Encyclopedia of physical science & technlogy* (3rd ed.). New York: Academic Press.
136, Penny, S. (2009). *Art & artificial life- a primer*. Paper presented at the Art & Artificial live.
137, Sharkey, N. S. A. (2007). Artificial intelligence and natural magic. *Science & Business*,
139, Eveling, S. (1926). *Space as a membrane*.Unpublished manuscript.
140, Turner, A., (2009). Ann in open processing. Retrieved 20 July 2011
141, Baresel, A., Sthamer, H., & Schmidt, M. (2002). *Fitness function design to improve evolutionary structural testing*. Paper presented at the Proceedings of the Genetic and Evolutionary Computation Conference.

142, Sadjadi, F. A. *Comparison of fitness scaling functions in genetic algorithms with applications to optical processing*.
143, Jani, A. K. D. N. N. (2010). A novel genetic algorithm approach for network design with robust fitness function. *International Journal of Computer Theory and Engineering, 2*(3),
144, Kitchin, M. D. R. (2001). *Mapping cyberspace*. London: Routledge.
145, Allen, P. M. (1997). *Cities and regions as self-organizing systems*. The Netherlands: Gordon and Breach Science Publishers.
146, Deepa, S. N. S. S. N. (2008). 3. 16 fitness scaling *Introduction to genetic algorithms* (pp. 70-77). Berlin: Springer
147, Mastorakis, N. E. (2005). *On the solution of ill-conditioned systems of linear and non-linear equations via genetic algorithms (gas) and nelder-mead simplex search*. Paper presented at the 6th WSEAS Int. Conf. on EVOLUTIONARY COMPUTING.
148, Schumacher, P. (2008). *Parametricism as style - parametricist manifesto*. Paper presented at the Dark Side Club1 , 11th Architecture Biennale.
149, Senagala, M. Post-spatial architectures: The emergence of time-like parametric worlds www.mahesh.org, *accessed 19 January 2011.*
150, Kronenburg, R. (1995). *Houses in motion. The genesis, history and development of portable buildings*. London: Academy Editions.
151, Engeli, M. (2001). *Bit and spaces. Architecture and computing for physical, virtual, hybrid dreams*: Birkhäuser.
152, Berger, H. (1996). *Light structures, structures of light*. Berlin: Birkháuser.
153, Joo, J. K., Kalaya. (2006). *Soft skin skyscrapper: Digital design and fabrication for tall building façade*. Paper presented at the Digital mock-up workshop.
154, LeCuyer, A. (2008). *Etfe. Technology and design*. Berlin: Birkhäuser.
155, Sharp, D. *Twentieth century architecture: A visual history*.
156, Boles, D. D. (1987). Modernism in the city– paris *Progressive Architecture 68*.
157, Vonier, T. (1995). Critique: Arab world institute. *Progressive Architecture, 76*, 62.
158, Casamonti, M. (2009). *Jean nouvel* (1st English ed. ed.).
159, Sabouni, I. (1991). *Lavapolis. Mars (modular assembly reusable structure*
160, *Advanced topics in artificial intelligence*. (1999).
161, Gail Peter Borden , M. M. (2011). *Matter: Material processes in architectural production* Routledge.

Figures References

Figure 1, from www.ruiz-geli.com.

Figure 2, from www.ruiz-geli.com.

Figure 3, from www.ruiz-geli.com.

Figure 4 , *(see # 97, Eveling, 2010 (original 1926))*.

Figure 5, from www.ruiz-geli.com.

Figure 6, from www.ruiz-geli.com.

Figure 7, from www.ruiz-geli.com.

Figure 8, from www.ruiz-geli.com.

Figure 9, from: http://organic.yukonfood.com/bubblehouse.htm

Figure 10, table from: http://organic.yukonfood.com/bubblehouse.htm.

Figure 11, pictures from: *http://solaroof.org/wiki*

Figure 12A, 12B www.greatbuildings.com Consulted 22 Feb 2012.

Figure 12C http://www.flickriver.com/photos/11413503@N03/sets/72157608880093896/ Consulted, 22 Feb 2012.

Figure 13, from www.and-architecture.org

Figure 14, from www.and-architecture.org

Figure 15, from www.and-architecture.org

Figure A,B,C, from Sean Hanna's, computational Analysis Module, at the MRes Adaptive Architecture & Computation. Lecture 9.

Figure M, from Alasdair Turner, computational Analysis Module, at the MRes Adaptive Architecture & Computation. Lecture 4.

Figure 16, from ,http://science.howstuffworks.com/environmental/conservation/conservationists/eden3.htm.

Figure 17, ETFE pillows from Iaso.

Figure 18, from Appendix code.

Figure 19, from http://elmundogcrideunido.wikispaces.com/VALENCIA

Figure 20, from Appendix code.

Figure 21, from Appendix code.

Figure 22, from http://ansonabey.hubpages.com/hub/Artificial-Neural-Network.

Figure 23, from http://itee.uq.edu.au/~cogs2010/cmc/chapters/BackProp/index2.html.

Figure 24, from Appendix code.

Figure 25, from Appendix code.

Figure 26, from Appendix code.

Figure 27, from Appendix code.

Figure 28, from Appendix code.

Figure 29, from Appendix code.

Figure 30, from Appendix code.