



UNIVERSIDAD DE MURCIA
DEPARTAMENTO DE INFORMÁTICA Y
SISTEMAS

Framework Basado en el Estándar de
Calidad del Software ISO/IEC
25000:2005 (SQuaRE) para la
Evaluación de la Calidad de las
Ontologías

D^a Astrid Duque Ramos
2016



FACULTAD DE INFORMÁTICA

Framework BASADO EN EL ESTÁNDAR DE
CALIDAD DEL SOFTWARE ISO/IEC
25000:2005 (*SQUARE*) PARA LA
EVALUACIÓN DE LA CALIDAD DE LAS
ONTOLOGÍAS

Tesis doctoral presentada en la Facultad de Informática de la Universidad de Murcia

por:

Astrid Duque Ramos

Dentro del Programa de Doctorado:

Matemática e Informática Aplicadas en Ciencias e Ingeniería

Dirigida por:

Dr. Jesualdo Tomás Fernández Breis



FACULTAD DE INFORMÁTICA

Framework BASADO EN EL ESTÁNDAR DE
CALIDAD DEL SOFTWARE ISO/IEC
25000:2005 (*SQUARE*) PARA LA
EVALUACIÓN DE LA CALIDAD DE LAS
ONTOLOGÍAS

Tesis doctoral presentada en la Facultad de Informática de la Universidad de Murcia
por:

Astrid Duque Ramos

Dentro del Programa de Doctorado:

Matemática e Informática Aplicadas en Ciencias e Ingeniería

Dirigida por:

Dr. Jesualdo Tomás Fernández Breis

El doctorando

El director

Murcia, 2016

A mis dos familias.

Mis padres: Gilberto su energía, Flor Ángela su perseverancia.

Mis hermanos: Yasmin, mi mejor ejemplo;

Bayron, su nobleza y Victor, su alegría.

Mi esposo Alejandro, mi gran amor.

Mi hijo David, el ángel que cayó del cielo.

El Divino Niño, que no me abandona

A todos ellos, los amo.

Agradecimientos

En primera instancia quiero dar infinitas gracias a mi tutor Jesualdo Tomás Fernández Breis, por la confianza depositada en mí, por darme la oportunidad de hacer el doctorado bajo su tutoría, mil gracias por su paciencia y por llevarme paso a paso por el sendero del conocimiento, porque gracias a él he podido avanzar a nivel investigativo, a él mi agradecimiento y admiración. A Pablo Artal por permitirme trabajar en su grupo de investigación mientras hacía mi doctorado, por las ayudas con el manejo del tiempo y por la financiación, mi respeto y aprecio. A Miguela Iniesta por brindarme su conocimiento y por su paciencia frente a mis dudas. A Stefan Schulz por su coautoría en varios artículos y sus sugerencias sobre el *framework*. A Ambrosio Toval que me recibió en su grupo y me ayudó en las primeras etapas del doctorado, por su colaboración le estoy muy agradecida.

A mis compañeros y amigos del Laboratorio, mis “*crazy ontologists*” Miguel, Luly Manu, Mari Carmen, Jojo, Juani, con los que tantas horas compartí mientras hacíamos nuestros doctorados; entre artículos, programación, preparación de congresos y escrituras de tesis, siempre hubo un momento para actualizarme en temas de historia de España, canciones, explicarme expresiones, etc. los cafés, las charlas, las cenas, por su amistad y por ser parte de mi “familia de España”. También a la sección internacional, con los que compartí los últimos meses de mi estancia en España: Pilar, Mario y Omar. A Manu especialmente, por su ayuda en estos últimos meses con los artículos y la entrega de la tesis. A los compañeros del laboratorio de Ingeniería del Software con los que compartí mis primeros años del doctorado: Fran, Fernando, Joaquín, Javi y Oscar. También a Joaquín Nicolás por su colaboración.

A mis amigos del LOUM, Juan Ma, Pedro, Eloy, Josua, Antonio, Esther, Juan, Luis, Silvestre, Guillermo, Manu, Martin, Chris, Alexandros, Dimitris, Haris, por todos los años compartidos, las conversaciones en la hora del café, los juegos de padel, las fiestas, los bailes, la Arrixaca, por enseñarme a hablar “español de España”, muchos momentos y muchas alegrías. A los amigos de la “comida donde los toros”: Silves, Esther, Juan, Estela, Dieguito y Maria, por las comidas compartidas y su amistad. En la Arrixaca al Doctor José María Marín a Sandra, Inés, Encarna, Lucía y Eloy por todos los buenos momentos mientras ajustábamos y bloqueábamos lentes intraoculares.

Al Club *Patu* o *¿Musul?*: *Chispa*, *Codi*, *Patu*, *Petite*, *Pegui*, *Patatoullie*, *Pulgarcito*, *Sapito*, *Vechi*, *Vice-Vice*, *Walle*, *Wauzi* y *Wiki* porque entre venezolanos, españoles, mexicanos, alemanes y colombianos, recreamos un pedacito de nuestras tierras en España, por todos los eventos y por el apoyo en los momentos difíciles, porque me ayudaban a distraerme cuando lo necesitaba por ser otra parte de “mi familia en España” y continuar siéndolo en la distancia. A mis músicos *Chispa*, *Codi*, *Wauzi* y coreógrafos *Walle* y *Patu*, por todas las horas invertidas para preparar los bailes.

A mis princesas Clem, Ire, Marina, Laura y Maria Luisa, y a Pily por compartir la danza del vientre y las tardes de ensayos, las cenas y el té y los churros. A mis amigos Mike y Karen por brindarnos su casa y ayudarnos con el inglés. A mi *teacher* Jason Blair por sus clases de inglés, los almuerzos compartidos y por sus enseñanzas en valores como la del “Flying Spaghetti Monster”. A mi amigo Terry por las conversaciones y las correcciones del inglés.

A Liz Maryori Builes por todas las horas invertidas en las correcciones de esta tesis, por su asesoría y paciencia. A Lili y Ana por su palabras de aliento por su amistad a pesar de la distancia. A Patricia, por ocuparse de David para que yo pudiera dedicar tiempo a la escritura de la tesis.

A mis hermanos por ser mi apoyo en todo momento, por estar siempre conmigo a través de la distancia, a mi hermana por ser mi amiga y

estar a mi lado cuando más la he necesitado, a ellos tres, mi cariño incondicional. A mis padres por todas sus enseñanzas y su empeño para que nosotros sus hijos pudiéramos lograr nuestras metas, a ellos infinitas gracias. A mi esposo Alejandro Mira, por llevarme a España y ayudarme siempre, por su amor y dedicación, por su comprensión, por ser mi hombre ideal. A nuestro hijo David, por sus sonrisas y por venir al mundo para acompañarme durante este último año del doctorado. Al Divino Niño, por estar siempre ahí.

Finalmente, quiero dar las gracias a todas aquellas personas que de una u otra forma han contribuido con la realización de mi doctorado.

Astrid Duque Ramos, 2016

Índice general

Índice de figuras	XI
Índice de tablas	XIII
1. Introducción	1
2. Estado del Arte	7
2.1. Definición de Ontologías	7
2.2. Elementos de las Ontologías	8
2.3. Representación de Ontologías	9
2.4. Razonamiento e inferencia de Ontologías	11
2.5. Calidad de Ontologías	13
2.6. Evaluación y métricas de Ontologías	14
2.6.1. Clasificación	14
2.6.2. Corrección	17
2.6.3. Calidad	18
2.7. Estándar de evaluación de calidad del software	19
2.8. Métricas de calidad del software	22
2.9. Clasificación de los métodos existentes de evaluación de ontologías, por subcaracterísticas de calidad	24
3. Objetivos y metodología	27
3.1. Motivación	27
3.1.1. Hipótesis	28
3.1.2. Formulación de preguntas de investigación	28

ÍNDICE GENERAL

3.2. Objetivo	28
3.2.1. Objetivos específicos	29
3.3. Metodología	29
3.3.1. Análisis del estado del arte	29
3.3.2. Definición del <i>framework</i>	30
3.3.3. Aplicaciones del <i>framework</i>	31
3.3.4. Evaluación y validación del <i>framework</i>	31
4. Framework OQuaRE (<i>Ontology Quality Requirements and Evaluation</i>)	33
4.1. <i>Framework</i> OQuaRE	33
4.2. Modelo de Calidad	34
4.3. Métricas de Calidad	40
4.3.1. Métricas básicas	40
4.3.1.1. Métricas no jerárquicas	41
4.3.1.2. Métricas jerárquicas (<i>asserted vs inferred</i>)	41
4.3.1.3. Métricas de rutas	41
4.3.2. Métricas derivadas	41
4.3.2.1. Métricas adaptadas desde la Programación Orientada a Objetos (POO)	42
4.3.2.2. Métricas desarrolladas por la comunidad de ontologías	42
4.4. Asociación de métricas de calidad al modelo de calidad	45
4.4.1. Asociación de métricas y subcaracterísticas	45
4.5. Ejemplo de OQuaRE	50
5. Aplicación del <i>framework</i> OQuaRE	53
5.1. Ontologías de tipos de células	53
5.2. Ontologías de unidades de medida	56
5.3. Ontologías desarrolladas por medio de una guía de buenas prácticas	59
5.4. Versionado de Ontologías	60
5.5. Ontologías financieras	63

6. Validación del <i>framework</i> OQuaRE	67
6.1. Comparación con evaluaciones humanas	68
6.2. Comparación con resultados de otras técnicas	70
6.2.1. Validación con <i>Gold Standard</i>	70
6.2.2. Validación con <i>Competency Questions</i>	71
6.3. Evaluación externa del modelo de calidad	72
7. Publicaciones que componen esta tesis	75
7.1. <i>OQuaRE: A SQuaRE based Approach for Evaluating the Quality of Ontologies</i>	75
7.2. <i>Evaluation of the OQuaRE framework for ontology quality</i>	95
7.3. <i>Evaluating the Good Ontology Design Guideline (GoodOD) with the Ontology Quality Requirements and Evaluation Method and Metrics (OQuaRE)</i>	106
8. Conclusiones y trabajo futuro	123
8.1. Contribución científica	123
8.2. Limitaciones identificadas	124
8.3. Trabajos futuros	125
8.4. Conclusiones	126
8.5. Contribuciones	127
8.5.1. Publicaciones	127
8.5.2. Conferencias Internacionales y <i>Workshops</i>	127
8.5.3. Premios	128
Bibliografía	129

Índice de figuras

2.1. Relación entre RDFS y los perfiles de OWL	11
2.2. Arquitectura de un sistema de representación de conocimiento (<i>KR</i>) basado en lógica descriptiva (<i>DL</i>)	12
2.3. Divisiones del Estándar de evaluación de la calidad del software <i>SQuaRE</i>	20
2.4. Modelo del estándar de evaluación de la calidad del software <i>SQuaRE</i>	21
4.1. Modelo de Calidad de OQuaRE	35
4.2. Ejemplo de OQuaRE	49
5.1. Evaluación manual de las ontologías de tipos de células	55
5.2. Evaluación automática de las ontologías de tipos de células	55
5.3. Evaluación manual de las ontologías de unidades de medida	57
5.4. Evaluación automática de las ontologías de unidades de medida	58
5.5. Diferencias significativas entre las ontologías de estudiantes entre- nados, no entrenados y el <i>Gold Standard</i>	60
5.6. Pasos para evaluar la calidad de las versiones de una ontología con base en el <i>framework</i> OQuaRE.	61
5.7. Representación de métricas que han cambiado su valor en la escala.	62
6.1. Comparación de la evaluación manual y automática para ontologías de tipos de células	69
6.2. Comparación de la evaluación manual y automática para ontologías de unidades de medida	69
6.3. Clasificación de subcaracterísticas en un clúster	73

Índice de tablas

2.1. Autores por Característica	24
4.1. Asociación de subcaracterísticas y métricas de calidad(parte 1) . .	46
4.2. Asociación de subcaracterísticas y métricas de calidad (parte 2) .	47
4.3. Aporte (+ ó -) de las métricas a la subcaracterística mantenibilidad	48
4.4. Escala 1 a 5 de las métricas de calidad	48
4.5. Métricas de EDAM escaladas	51
4.6. Cálculo de subcaracterísticas de EDAM	51

The beginning is the most important part of the work.

Plato

CAPÍTULO

1

Introducción

La web semántica *“es una extensión de la web que proporciona un lenguaje para expresar los datos y reglas de razonamiento acerca de los datos, y que permite a las reglas de cualquier sistema de representación de conocimiento ser exportadas a la Web, permitiendo a las personas y a los ordenadores trabajar en cooperación”* [Berners-Lee et al., 2001].

La web semántica necesita contar con información estructurada que incluya reglas de inferencia que puedan ser evaluadas por máquinas de procesamiento de forma automática. Las ontologías son estructuras conceptuales que definen formalmente las relaciones entre términos. Según Gruber, *“una ontología define (o especifica) los conceptos, relaciones, y otras distinciones que son relevantes para modelar un dominio”* [Gruber, 1993], es por ello que *“son una tecnología clave para la Web Semántica”*[Fensel and Musen, 2001].

El uso de las ontologías ha aumentado significativamente en los últimos años, debido a que *“proporcionan un conjunto de términos estructurados jerárquicamente para describir un dominio que puede ser usado para una base de conocimiento”* [Patil et al., 1997]. Dentro de los dominios en los que se han aplicado con éxito las

1. INTRODUCCIÓN

ontologías podemos mencionar *el comercio electrónico* [Hepp, 2008], *la biomedicina* [Ashburner et al., 2000, Beisswanger et al., 2008], *el elearning* [Sánchez-Vera et al., 2012] o *los datos geoespaciales* [Tian and Huang, 2012]. Existen también varios repositorios de ontologías como *Bioportal*¹, que cuenta con más de cuatrocientas cincuenta ontologías biomédicas y vocabularios controlados, o como el repositorio *TONES*² que también contiene más de cuatrocientas, entre otros.

Debido a esta creciente utilización de ontologías, los constructores de ontologías y desarrolladores de aplicaciones se enfrentan a múltiples opciones a la hora de seleccionar la ontología adecuada para un propósito particular o para su reutilización. Es por ello que identificar las características que son relevantes para el uso previsto y los requerimientos que debe cumplir la ontología se ha tornado difícil, haciéndose necesario contar con mecanismos de asistencia en la toma de decisiones informadas. Sabou dice: “*la selección de ontologías es el proceso que permite identificar una ontología o módulos de ontología que cumplen ciertos criterios. El proceso de chequear si una ontología cumple ciertos criterios es una tarea de evaluación de una ontología*” [Sabou et al., 2006].

La evaluación de ontologías, según Brank, [Brank et al., 2005] tiene como objetivos principales:

1. *Ayudar a los usuarios a seleccionar, entre un conjunto de ontologías, aquella que mejor se adapte a sus necesidades.*
2. *Ayudar a los desarrolladores a evaluar sus ontologías a partir de un proceso de desarrollo, con el fin de generar confianza para la construcción y reutilización de ontologías.*

Con estos dos objetivos se han desarrollado metodologías de evaluación de ontologías desde diferentes enfoques, algunas de ellas basadas en la comparación con una ontología de referencia predefinida (*Gold Standard*) [David and Euzenat, 2008, Dellschaft and Staab, 2006, Maedche and Staab, 2002], otras basadas

¹<http://bioportal.bioontology.org/>

²<http://owl.cs.manchester.ac.uk/repository/>

en el uso de la ontología en una aplicación determinada y la evaluación de los resultados [Tello and Gómez-Pérez, 2004, Alani et al., 2006, Tartir et al., 2005], otras se basan en la comparación con una fuente de datos del dominio que cubre la ontología ([Corcho et al., 2004, Guarino and Welty, 2004, Sleeman and Reul, 2006], [Vrandečić, 2010]) y, finalmente, la evaluación realizada por expertos del dominio que verifican que la ontología cumpla con requisitos, criterios y estándares previamente establecidos ([Gangemi et al., 2006, Rogers, 2006, Stvilia, 2007]). Sin embargo, cada método tiene un proceso propio de evaluación y no existe un consenso sobre un modelo estándar para evaluar la calidad de las ontologías. Según Kalfoglou *et al.*, “*El desarrollo y la evaluación de ontologías requieren la asistencia y aplicación de metodologías y estándares*” [Kalfoglou and Schorlemmer, 2003].

Gracias a su naturaleza, las ontologías se pueden considerar como artefactos software, haciendo posible aplicar métodos de Ingeniería del Software a su desarrollo y evaluación. Además, en términos de madurez de disciplinas, la Ingeniería del Software se encuentra más avanzada y desarrollada que la Ingeniería de Ontológica. Es por ello que se han aplicado técnicas de Ingeniería del Software a la Ingeniería del Conocimiento; entre ellas se pueden destacar *TOVE* [Gruninger and Fox, 1994], el enfoque *Enterprise Model* [Uschold et al., 1996] o el uso de métodos de Ingeniería del Software para mejorar la Ingeniería Ontológica como *Methontology* [López et al., 1999]). Recientemente se han propuesto enfoques basados en buenas prácticas, como los Patrones de diseño de Ontologías¹, principios como los definidos por el *OBO Foundry*² o directrices como *GoodOD* [Boeker et al., 2012]. Adicionalmente, se han usado métodos de Ingeniería de Software para la mejora de las metodologías de Ingeniería Ontológica o enfoques de Ingeniería Inversa adaptada a Reingeniería Ontológica ([Chikofsky and Cross II, 1990, Gómez-Pérez and Rojas-Amaya, 1999]),

Según Kalfoglou, “*la evaluación debe ser hecha por profesionales y adherida a estándares y prácticas aprobadas por organizaciones reconocidas*” [Kalfoglou and Schorlemmer, 2003]. Sin embargo, en la literatura no se han encontrado intentos sistemáticos para evaluar la calidad de las ontologías. En el ámbito de la Ingeniería

¹<http://ontologydesignpatterns.org>

²<http://www.obofoundry.org/crit.shtml>

1. INTRODUCCIÓN

del Software los estándares proponen medir la calidad del software a través de su diseño (calidad de diseño), y por la correspondencia entre lo implementado y lo diseñado (calidad de conformidad).

Aplicar un estándar de calidad es recomendable ya que proporciona una especificación y un modelo de evaluación exhaustivo para la calidad de un producto software; supone un apoyo a la comprensión de las necesidades del usuario, dotando a evaluadores y desarrolladores de un lenguaje común para especificar requisitos de calidad; y proporciona una manera objetiva de evaluar la calidad del producto software, que no depende de opiniones de terceros. Como consecuencia de lo anterior, al evaluar dos veces el mismo producto software, sin haber introducido cambios, se deben obtener los mismos resultados de calidad, por tanto, la evaluación de la calidad es reproducible.

Estas características son deseables para la evaluación de calidad de ontologías, ya que van en línea con los criterios que debe cumplir una estrategia de evaluación de ontologías que, según Kalfoglou [Kalfoglou and Schorlemmer, 2003], son:

1. *Abierta al usuario con referencias a los estándares que adhiere a sus certificados.*
2. *Fácil de cambiar y adaptar a casos de uso diferentes.*
3. *Que especifica el dominio y refleja opiniones de varios stakeholders, no sólo de los ingenieros de ontologías.*

En este trabajo se combina la Ingeniería Ontológica con la Ingeniería del Software a través de la adaptación de un estándar de evaluación de la calidad del software y de las métricas de software a las ontologías con el fin de buscar una estandarización, por lo que se propone el *framework* OQuaRE para la evaluación de ontologías, basado en el estándar de evaluación de calidad del software *Standard Quality Requirements and Evaluation (SQuaRE)*. *SQuaRE* es un estándar de evaluación de requisitos de calidad de productos software, que sustituye a la norma de las series ISO / IEC 9126 serie y la serie 14598.

El método se ha aplicado para la evaluación de ontologías de diferentes dominios, se ha validado en diferentes escenarios y se ha evaluado con un grupo de expertos. Esta tesis doctoral se compone de tres publicaciones que se presentan en el capítulo 7. Aquí se incluye una breve explicación de cada una de ellas para mostrar la unidad científica de la tesis.

- En el artículo *OQuaRE: a SQuaRE-based approach for evaluating the quality of ontologies* se crea el método de evaluación de ontologías OQuaRE. En este artículo se estudia el estándar de calidad del software *ISO 25000:2005* o *SQuaRE*, y a partir de él se define el método de evaluación de calidad de ontologías OQuaRE (*Ontology Quality Requirements and Evaluation*). Inicialmente se definieron el modelo y las métricas de calidad en términos ontológicos; se incluyeron las características y subcaracterísticas que no pertenecían al modelo estándar de calidad de software SQuaRE, pero que son requeridas para la evaluación de ontologías; se identificaron métricas de calidad del software aplicables a las ontologías, algunas de ellas tomadas de autores anteriores y otras tomadas de las métricas de software y adaptadas a las ontologías; finalmente, se creó una escala con el fin de cuantificar las subcaracterísticas, características y métricas. A continuación, se implementó una herramienta JAVA para aplicar el *framework* OQuaRE y se evaluaron 13 ontologías de dos dominios diferentes. Los resultados mostraron la factibilidad de la adaptación de un método de calidad de software a ontologías. Además, el *framework* OQuaRE permitió identificar las fortalezas y debilidades de las ontologías en términos de subcaracterísticas, aportando información detallada sobre la calidad de las ontologías.
- En el artículo *Evaluation of the OQuaRE framework for ontology quality* se presenta la evaluación del *framework* OQuaRE, realizado por un grupo de expertos en construcción de ontologías biomédicas. En este artículo se realizó una evaluación de la utilidad y relevancia del modelo y las métricas de calidad de OQuaRE. Se siguió el método *Goal Question Metric (GQM)*, un método de evaluación basado en objetivos. Con base en éste, se diseñó el experimento, se generaron los artefactos y se recolectaron los datos para la evaluación de OQuaRE. Se realizó el análisis estadístico que incluyó las

1. INTRODUCCIÓN

pruebas *ANOVA* de un factor. Los resultados obtenidos permitieron identificar la relevancia y utilidad de las combinaciones métrica-subcaracterística. Se identificaron además los aspectos positivos y negativos del método, y se recogieron sugerencias de los expertos, que luego se tuvieron en cuenta para hacer una redefinición de las métricas y la escala de valores.

- En el artículo *Evaluating the Good Ontology Design guideline (GoodOD) with the Ontology Quality Requirements and Evaluation method and metrics (OQuaRE)* se aplica el *framework* OQuaRE para evaluar un grupo de ontologías desarrolladas por un grupo de estudiantes, antes y después de ser entrenados con la guía para desarrollo de buenas ontologías *Good Ontology Design Guideline (GoodOD)*, la cual ha sido diseñada por un grupo de expertos en construcción de ontologías. Inicialmente, se diseñó el experimento y se calcularon las métricas de calidad de OQuaRE, se realizó el estudio comparativo de los estudiantes entrenados y no entrenados, por medio de pruebas estadísticas como *ANOVA* bifactorial, análisis de componentes principales y análisis de clúster. Luego se procedió a comparar los resultados con los obtenidos en dos evaluaciones previas, *Competency Questions* y *Gold Standard*, realizadas por los desarrolladores de la guía. Los resultados mostraron la utilidad de OQuaRE para la evaluación de ontologías desarrolladas siguiendo buenas prácticas. Además, se mostró que OQuaRE aporta más información sobre la calidad de las ontologías evaluadas que otros métodos de evaluación y, además, permite comparar ontologías.

Las tres publicaciones representan la unidad científica de la tesis, la cual se centra en la definición, aplicación y evaluación del *framework* OQuaRE (*Ontology Quality Requirements and Evaluation*), un *framework* de evaluación de la Calidad de Ontologías, basado en el estándar de Calidad del Software, *ISO/IEC 25000:2005* o *SQuaRE*.

*Life is long enough to do
everything.*

Argemira Agudelo

CAPÍTULO

2

Estado del Arte

2.1 Definición de Ontologías

El término “Ontología” proviene de la filosofía, significa el “estudio de lo que es, lo que existe” [Hofweber, 2009]. En la filosofía, las ontologías han sido definidas por diferentes autores y desde diferentes perspectivas. Desde la perspectiva de Inteligencia Artificial (IA), las definiciones de ontología se asemejan a la interpretación del filósofo Quine [Fernández-Breis, 2003], quien dijo que: *“todo lo que puede ser cuantificado existe”* [Quine, 1961]. La primera definición de ontología en Inteligencia Artificial apareció en [Neches et al., 1991]: *“una ontología define los términos básicos y relaciones que conforman el vocabulario de un área específica, así como las reglas para combinar dichos términos y las relaciones para definir extensiones de vocabularios”*. Sin embargo, una de las definiciones más extendidas es la dada por Tom Gruber *“una ontología es una especificación de una conceptualización”* [Gruber, 1993]. Borst mejora esta definición así: *“una ontología es una especificación formal de una conceptualización compartida”* [Borst, 1997]. Posteriormente, las definiciones de Gruber y Borst fueron explicadas por Studer de la siguiente forma: *“conceptualización se refiere a un modelo abstracto de algún fenómeno en el mundo a través de la identificación de los conceptos relevantes de dicho fenómeno. Explícita significa que el tipo de conceptos y restricciones usados*

2. ESTADO DEL ARTE

se definen explícitamente. Formal representa el hecho de que la ontología debería ser entendible por las máquinas. Compartida refleja la noción de que una ontología captura conocimiento consensuado, esto es, que no es de un individuo, sino que es aceptado por un grupo” [Studer et al., 1998].

2.2 Elementos de las Ontologías

Según Gruber, existen cuatro tipos de elementos en las ontologías: relaciones, funciones, axiomas e instancias [Gruber, 1993]. Fernández-Breis describe estos elementos de la siguiente forma [Fernández-Breis, 2003]:

- *Las clases en la ontología se suelen organizar en taxonomías. Algunas veces, la noción de ontología se diluye en el sentido que las taxonomías se consideran ontologías completas [Studer et al.; 98]. Se suele usar tanto el término clases como conceptos. Un concepto puede ser algo sobre lo que se dice algo y, por lo tanto, también podría ser la descripción de una tarea, función, acción, estrategia, proceso de razonamiento, etc.*
- *Las relaciones representan un tipo de interacción entre los conceptos del dominio. Se definen formalmente como cualquier subconjunto de un producto de n conjuntos, esto es: $R: C1 \times C2 \times \dots \times Cn$. Como ejemplos de relaciones binarias incluimos: subclase de y conectado a.*
- *Los axiomas son expresiones que son siempre ciertas. Pueden ser incluidas en una ontología con muchos propósitos, tales como definir el significado de los componentes ontológicos, definir restricciones complejas sobre los valores de los atributos, argumentos de relaciones, etc. verificando la corrección de la información especificada en la ontología o deduciendo nueva información. Tales ontologías son llamadas ontologías pesadas, en contraste con las ontologías ligeras que no incluyen axiomas.*
- *Las instancias se usan para representar elementos específicos.*

2.3 Representación de Ontologías

Existen diferentes lenguajes de representación de ontologías, los más utilizados son el *Resource Description Framework Schema (RDFS)* y el *Web Ontology Language (OWL)*. RDFS es una extensión de *RDF, Resource Description Framework (RDF)*, el cual define un modelo estándar para el intercambio de recursos en Internet, los recursos se identifican por medio del identificador *Uniform Resource Identifiers (URIs)* y se describen en términos de sentencias simples en forma de tripletas, las cuales se componen de un sujeto, un predicado y un objeto. El sujeto es una *URI* que identifica a un recurso, el objeto puede ser un valor literal (cadena de caracteres, número o fecha), o la *URI* de otro recurso que está relacionado con el sujeto. El predicado es una *URI* que indica el tipo de relación que existe entre el sujeto y el objeto [Legaz-García, 2015]. RDFS permite expresar ontologías simples utilizando la sintaxis RDF, define los elementos *rdfs:Class*, *rdfs:Resource* y *rdf:Property* para definir clases (conceptos), recursos y propiedades, respectivamente. El contenido expresado en RDFS está organizado por relaciones de subclases y subpropiedades, restricciones de dominio y rango e instancias de clases [Antoniou and Harmelen, 2003].

El lenguaje *Web Ontology Language (OWL)* [W3C, 2004b], diseñado por el *W3C Web Ontology Working Group* “*puede ser usado para representar explícitamente el significado de términos en vocabularios y las relaciones entre esos términos*”¹. OWL extiende las capacidades de RDF y RDF Schema. OWL declara clases y propiedades y las organiza en jerarquías de subclases y subpropiedades respectivamente. Las clases pueden ser especificadas como combinaciones lógicas (intersecciones, uniones o complementos) de otras clases o como enumeraciones de objetos específicos. OWL proporciona dominio y rango para las propiedades, las propiedades pueden ser transitiva, simétrica, funcional o la inversa de otra propiedad. OWL puede expresar la pertenencia de objetos (individuos) a clases y aquellos valores de propiedades que son de individuos específicos. OWL proporciona restricciones sobre el comportamiento de las propiedades en una cierta clase [Horrocks et al., 2003].

¹<http://www.w3.org/2007/09/OWL-Overview-es.html>

2. ESTADO DEL ARTE

“El poder de la expresividad de un lenguaje como OWL está determinado por los constructores de clases (y propiedades) soportados, y por las clases de axiomas que pueden ocurrir en la ontología” [Horrocks et al., 2003].

OWL define tres sublenguajes con diferentes niveles de expresividad [W3C, 2004b]:

- OWL Lite: soporta sólo un subconjunto de constructores de OWL, una jerarquía de clasificación con restricciones simples. Es el lenguaje menos expresivo.
- OWL Full: proporciona la máxima expresividad y la libertad sintáctica de RDF sin garantías computacionales. Permite mezclar libremente OWL y RDF. Por lo tanto, cualquier documento RDF es un documento válido en OWL Full.
- OWL DL: proporciona la mayor expresividad garantizando que todas las conclusiones son computables y finalizarán en un tiempo finito. Incluye todos los constructores del lenguaje OWL, pero solo pueden ser usados bajo ciertas restricciones. El nombre de OWL DL proviene de su correspondencia con la lógica descriptiva $\{\mathcal{SHOIN}(\mathcal{D})\}$ [Baader and Nutt, 2003], un campo de la investigación que ha estudiado las lógicas que constituyen la base formal de OWL. La lógica descriptiva es una familia de formalismos de representación del conocimiento basado en clases.

OWL 2 es fruto de la evolución de OWL, incrementando su nivel de expresividad de $\{\mathcal{SHOIN}\}$ a $\{\mathcal{SHOIR}\}$ [Grau et al., 2008]. OWL 2 introduce tres perfiles adicionales a OWL DL con distinto nivel de expresividad [W3C, 2009]:

- OWL 2 EL: es adecuado para aplicaciones que requieren ontologías muy grandes y donde se puede sacrificar poder de expresividad para obtener mejor rendimiento. Habilita un tiempo polinomial para las tareas de razonamiento.
- OWL 2 QL: destinado a aplicaciones donde ontologías relativamente ligeras se usan para organizar un gran número de individuos y donde es útil acceder a los datos directamente a través de consultas relacionales (SQL). Permite responder a consultas de un modo formal y completo en un tiempo computacional razonable.

- **OWL 2 RL**: destinado tanto a aplicaciones OWL 2 que sacrifican la total expresividad del lenguaje para mejorar la eficiencia, como aplicaciones RDFS que necesitan mayor expresividad proveniente de OWL 2. Permite la implementación de algoritmos con tiempo de razonamiento polinomial usando tecnologías de bases de datos basadas en reglas que operan directamente en tripletas RDF.

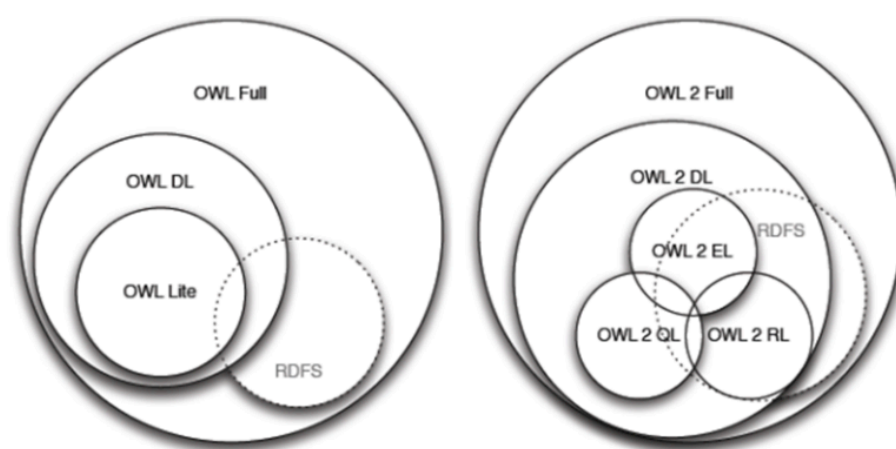


Figura 2.1: Relación entre RDFS y los perfiles de OWL

2.4 Razonamiento e inferencia de Ontologías

La inferencia en la Web Semántica es el procedimiento para generar nuevo conocimiento a partir del existente y de conjuntos de reglas existentes [W3C, 2004a]. “El razonamiento permite inferir conocimiento representado implícitamente, del conocimiento que está explícitamente contenido en la base del conocimiento” [Baader and Nutt, 2003]. Las reglas de inferencia son usadas por los razonadores como método para inferir dicho conocimiento.

Un sistema de Representación de conocimiento (*KR*) basado en *DL* proporciona facilidades para configurar la base de conocimiento, razonar acerca de su contenido y manipularlo. Una base de conocimiento (*KB*) está compuesta de TBox, que introduce la terminología (el vocabulario de un dominio de aplicación, como clases, propiedades y restricciones) y el ABox que contiene las declaraciones acerca de los

2. ESTADO DEL ARTE

individuos en términos de su vocabulario. [Baader and Nutt, 2003]. En la figura 2.2 se muestra la arquitectura de un sistema de representación de conocimiento (*KR*) basado en lógica descriptiva (*DL*).

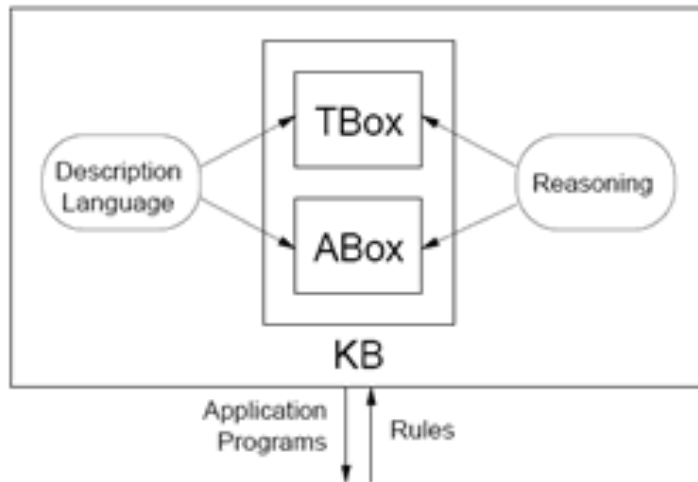


Figura 2.2: Arquitectura de un sistema de representación de conocimiento (*KR*) basado en lógica descriptiva (*DL*)

Cualquier sistema *KR* basado en lógica descriptiva es capaz de proporcionar un conjunto específico de razonamientos como mínimo [Baader and Nutt, 2003]. OWL DL es una variante sintáctica de la lógica descriptiva $\{\mathcal{SHOIN}(\mathcal{D})\}$ [Horrocks et al., 2007] por lo que un razonador OWL debe proporcionar al menos el siguiente conjunto de servicios de inferencia de lógica descriptiva:

- Chequeo de consistencia: comprueba si una ontología contiene alguna definición contradictoria.
- Satisfacibilidad de clase: comprueba si es posible que una clase tenga instancias. Si una clase no puede tener instancias, definir instancias para esa clase causa que la ontología sea inconsistente.
- Clasificación: computa las relaciones de subclase entre todas las clases para crear la jerarquía de clases completa.

- Realización: encuentra la clase más específica a la que un individuo pertenece. Sólo se puede realizar después de la clasificación, ya que el tipo de un individuo se define respecto a la jerarquía de clases.

“Existen varios axiomas en OWL-DL que son relevantes para el razonamiento. El axioma `subClassOf` se conoce como la condición necesaria. Una condición necesaria es una condición que debe cumplirse para que un individuo pertenezca a una clase, pero no es suficiente por sí sola. El axioma `equivalentTo` se conoce como condición necesaria y suficiente. Una condición necesaria y suficiente es una condición que, si se cumple, basta para garantizar que un individuo pertenece a una clase. Otros axiomas de interés son aquellos que expresan identidad en las entidades en OWL, afirmando que una entidad dada es diferente a otras. Los axiomas `sameAs` y `differentFrom` son usados en individuos, mientras que los axiomas `disjointFrom` y `equivalentTo` se utilizan en clases” [Legaz-García, 2015].

Algunas de las herramientas de razonamiento disponibles para OWL son: Pellet [Sirin et al., 2007], FaCT++ [Horrocks, 1998] y Hermit [Shearer et al., 2008].

2.5 Calidad de Ontologías

En términos generales, la calidad ha sido definida como la capacidad para ajustarse a las especificaciones o conformidades de un conjunto de requisitos [Crosby, 1979].

La calidad de las ontologías, en contraste con los modelos conceptuales, tienen que satisfacer requerimientos computacionales también como requerimientos representacionales. La calidad se caracteriza por la capacidad para responder a preguntas de competencia, las cuales diseñadas antes de construir la ontología y sirve como un tipo de definición de requerimientos de ontologías. *“Un aspecto pragmático de la calidad concierne a la utilidad y usabilidad que es determinada por el lenguaje en el que está especificada la ontología y la herramienta software disponible para su manipulación”* [Evermann and Fang, 2010].

La calidad de una ontología es un factor importante para determinar su utilidad. Para asegurar la calidad de una ontología, se requiere evaluar si está construida adecuadamente y si el resultado refleja la tarea de construcción [Ohta et al., 2011].

2.6 Evaluación y métricas de Ontologías

En los últimos años se han desarrollado varios enfoques de evaluación de ontologías y se han aplicado a diferentes dominios. Estos enfoques se pueden dividir en tres categorías principales de acuerdo al objetivo particular de la evaluación: clasificación, corrección y calidad.

2.6.1 Clasificación

Los enfoques de esta categoría combinan criterios cuantitativos y cualitativos para medir (automática o manualmente), comparar y seleccionar una ontología. Estos enfoques abarcan desde la clasificación de ontologías genéricas hasta la selección de la ontología más adecuada para una tarea particular. Algunos de ellos son descritos a continuación.

Tartir define un método para seleccionar la ontología más adecuada. Para ello evalúa el contenido de la ontología a través de la distribución de sus clases, el esquema del árbol de herencia, la distribución de las instancias de clase y la conectividad entre las instancias de diferentes clases. La evaluación se hace con base en los términos que ingresa el usuario, los cuales son extendidos adicionando términos de WordNet [Miller, 1995], para luego determinar clases y relaciones que contienen alguno de los términos extendidos. Finalmente se calculan las métricas de base de conocimiento (*KB*) y las métricas para todas las clases y relaciones para obtener una puntuación. La puntuación se mide como el peso promedio de las métricas de esquema, sobre todas las métricas *KB* y las métricas de las clases y relaciones [Tartir and Arpinar, 2007].

OntoQA es un método para evaluar la calidad de ontologías en dos dimensiones diferentes: una orientada al diseño de la ontología y la otra orientada a la efectividad del diseño y la cantidad de conocimiento en la ontología. El autor clasifica las métricas en métricas de esquema y métricas de instancia. Las métricas de esquema evalúan el diseño ontología y su potencial para la representación “rica” del conocimiento y las métricas de instancia evalúan la ubicación de las instancias dentro de la ontología y el uso eficaz de la ontología para representar el conocimiento modelado en ella. Este método se basa en las clases, atributos y relaciones de las ontologías para definir las métricas, las cuales son: riqueza de la clase, media, población, cohesión,

2.6 Evaluación y métricas de Ontologías

importancia, plenitud, conectividad, herencia y legibilidad [Tartir et al., 2005]. Yinglong Ma et al. [Ma et al., 2009] proponen un conjunto de métricas de cohesión semántica para evaluar la calidad de la evolución de las ontologías. Los autores argumentan que estos indicadores propuestos “*son estables y se pueden utilizar para medir la semántica de la ontología en lugar de su estructura*” .

Las métricas de cohesión presentadas fueron definidas por los autores como:

1. *Average Axiom fanouts per Clase (AAFC)*: indica si la ontología requiere ser corregida para que su diseño esté balanceado entre el conocimiento del dominio general y detallado.
2. *Number of Minimally Inconsistent subsets (NMIS)*: representa el valor absoluto del conjunto de todos los subconjuntos mínimamente inconsistentes en la base del conocimiento.
3. *Average Value of Axiom Inconsistencies (AVAI)*: representa la razón de la suma del impacto de las inconsistencias de todos los axiomas y afirmaciones sobre la cardinalidad de una base de conocimiento.

Los autores dicen que “*mientras más alto el valor de MI, más difícil de compartir la base del conocimiento*” y “*mientras más alto sea el valor de AVAI más inconsistente es la ontología*”. Valores altos de AVAI o NMIS indican que se deben extraer los axiomas que tienen el valor de impacto más alto de inconsistencia para borrarlos de la ontología. Se revisa y repara de forma iterativa la ontología, hasta que la resultante no presente incompatibilidades. Se presentó una herramienta en JAVA para calcular SMIS y AVAI.

Yao et al. [Yao et al., 2005] definen un conjunto de indicadores de cohesión para ontologías OWL. Estas métricas, similares a las métricas de cohesión de software definidas por Chidamber y Kemerer [Chidamber and Kemerer, 1994], miden la separación de responsabilidades y la independencia de los componentes de ontologías. Se definen tres métricas: Número de Clase Raíz, Número de Clases hoja y profundidad media del árbol de herencia de los nodos hoja [Yao et al., 2005].

ONTOMETRIC es un método que permite seleccionar la ontología más adecuada entre varias alternativas, o la más idónea para un proyecto específico. Para ello

2. ESTADO DEL ARTE

se diseñaron 160 características de comparación. El marco está diseñado taxonómicamente de forma jerárquica en la que se presentan muchos aspectos descritos por factores, los factores por características y estas por subcaracterísticas más específicas [Tello and Gómez-Pérez, 2004].

El usuario selecciona los criterios que considera oportunos para construir un nuevo árbol que depende de las particularidades del sistema que reutilizará la ontología. El autor define la ontología de referencia, basada en el método y en la que se definen los conceptos: Lenguaje, Metodología, Herramienta, Ontología, Clase, Atributo, Instancia, Relación y Axioma. La Ontología de Referencia reúne información de estos conceptos de 160 ontologías existentes y la Herramienta *OntoMetric* provee el árbol de criterios por defecto con las características identificadas en esta Ontología de referencia.

Finalmente, los autores desarrollan el método *ONTOMETRIC* para la reutilización de ontologías, adaptando el Proceso de Jerarquía Analítica *Analytic Hierarchy Process (AHP)* de Saaty, el cual es una técnica para toma de decisiones en problemas complejos multi-criterio. El método es explicado en más detalle en [Saaty, 1980, Saaty, 2000].

ONTOMETRIC consiste de cinco pasos:

- Especificar los objetivos del proyecto.
- Construir el árbol de decisión (objetivo, los criterios y las alternativas) con base en la herramienta *OntoMetric*.
- Para cada nivel construir la matriz de comparación de pesos entre nodos hermanos, con base en los criterios del árbol de decisión.
- Valorar cada alternativa con una escala fijada previamente.
- Combinar los vectores de pesos obtenidos en la matriz y los valores del paso anterior, con estos resultados se va ascendiendo en el árbol hasta obtener el valor del nodo raíz y seleccionar la ontología más adecuada.

Los autores concluyen: “*ONTOMETRIC es una adaptación del método AHP para ayudar a los ingenieros del conocimiento a escoger la ontología adecuada para un nuevo proyecto. Para ello, los ingenieros deben comparar la importancia*

de los objetivos y estudiar cuidadosamente las características de la ontología”.

En Malone *et al.* se definen perfiles de actividad, que son descripciones cualitativas en términos de cantidad de clases adicionadas, cambiadas o borradas de la ontología [Malone and Stevens, 2013]; el cálculo de los valores se hace a través de la herramienta *Bubastis*¹. *Bubastis* [Malone et al., 2010] ejecuta un análisis de diferencias sintácticas (*diff analysis*) entre versiones consecutivas de la ontología, donde se cuenta el número de clases adicionadas, borradas, número de cambios de axiomas que han ocurrido en clases existentes.

2.6.2 Corrección

En esta categoría se incluyen los enfoques que tratan de la corrección formal, sintáctica y lógica del conocimiento ontológico y las primitivas utilizadas. Estos enfoques se refieren a la taxonomía de la ontología ([Corcho et al., 2004, Guarino and Welty, 2004, Sleeman and Reul, 2006, Vrandečić, 2010]).

En Sleeman *et al.* se presenta un sistema de chequeo de ontologías basado en la ruta de los conceptos extraídos de *WordNet* [Sleeman and Reul, 2006]. Los autores definen cada concepto como una ruta que contiene el concepto, sus hijos y sus padres hasta la raíz. Este enfoque se basa en tres fases [Sleeman and Reul, 2006]:

- Buscar en *Wordnet* cada uno de los conceptos que aparecen en la ontología para reportar las rutas [Miller, 1995].
- Eliminar los enlaces inapropiados, los cuales son de dos tipos, aquellos cuyos nodos no están en el diccionario o aquellos que son inconsistentes; como resultado se obtienen algunos enlaces rotos y algunos conceptos de la ontología huérfanos. Un algoritmo elimina de la ontología los subárboles que no satisfacen las dos condiciones.
- Incluir los subárboles y los nodos huérfanos en la ontología para tener un árbol consistente y, si es necesario, revisar nuevamente las tres fases.

OntoClean es una metodología que permite detectar inconsistencias semánticas y formales en las propiedades definidas por la ontología. Para chequear la exactitud

¹ (<http://www.ebi.ac.uk/efo/bubastis/>)

2. ESTADO DEL ARTE

de las relaciones taxonómicas de una ontología hace una evaluación formal de las propiedades definidas en la ontología por medio de una estructura ideal taxonómica predefinida por metapropiedades. Se definen tres metapropiedades: rigidez, que indica que la propiedad es esencial para todas sus instancias; identidad, la capacidad de reconocer si las entidades individuales son las mismas o diferentes y unidad, que se refiere al problema de identificar la partes de un objeto, de forma que se sabe cuáles son las partes del objeto y cuáles no. La aplicación de *OntoClean* se inicia con un conjunto de clases dispuestas en una taxonomía, se asignan las metapropiedades (rigidez, identidad y unidad) y la dependencia de cada propiedad a la taxonomía. Luego se analizan las propiedades rígidas y se verifica que las restricciones impuestas por las metapropiedades se cumplan, realizando correcciones cuando no se cumplen. Finalmente se analizan los tres casos de propiedades no-rígidas para obtener una “*ontología limpia*” [Guarino and Welty, 2004].

2.6.3 Calidad

En esta categoría se incluyen los enfoques que evalúan diferentes criterios de calidad de la ontología.

En Gangemi *et al.* [Gangemi et al., 2006] se proporciona un catálogo de medidas cuantitativas y cualitativas para evaluar las ontologías. Se mide la ontología en varias dimensiones: estructural, enfocada en sintaxis y semántica; perfil de usabilidad, se enfoca en las anotaciones de la ontología; y funcional, relacionada con el uso de la ontología y sus componentes. *O2* y *oQua* permiten hacer un diseño paramétrico de evaluación y validación de tareas. “*Este método no sugiere validación prescriptiva sino una validación distribuida e interactiva contra tareas*” .

Otros enfoques de evaluación se basan en la comparación de ontologías contra una ontología de referencia *Gold Standard* predefinida, como métricas de *ontology learning* [Dellschaft and Staab, 2006] y similitud de ontologías [David and Euzenat, 2008, Maedche and Staab, 2002]. Los *Gold Standard* son un recurso importante pero generalmente no están disponibles, especialmente en aquellos dominios para los que se desarrollan nuevas ontologías.

Existen otros enfoques que son el uso de “*Competency Questions*”, que pueden ser entendidas como la formulación de los requisitos de la ontología en forma de

2.7 Estándar de evaluación de calidad del software

preguntas, para los que se crean las respuestas correctas por expertos en el dominio. Una buena ontología debe proporcionar la respuesta correcta a estas preguntas con la ayuda de razonamiento automatizado [Fox et al., 1997]. Esta técnica también ha sido mencionada por enfoques de evaluación [Obrst et al., 2007] y construcción de ontologías [Gruninger and Fox, 1995].

Los métodos de desarrollo y evaluación de ontologías están relacionados. El informe final del *Ontology Summit 2013* identifica la necesidad de un método para desarrollar y evaluar ontologías [Neuhaus et al., 2013]. El método de centrarse en la trazabilidad de requisitos desde el ciclo de vida del desarrollo hasta la evaluación de la calidad. “*Se identificaron cinco características de calidad de ontologías y se combinaron en un modelo para ciclo de vida de la ontología: Desarrollo de Requisitos, Análisis de la Ontología, Diseño del Sistema, Desarrollo de la Ontología, Desarrollo del Sistema e Integración*”. Para cada fase del modelo se propone un conjunto de criterios de evaluación de la ontología en términos de cuestiones de competencia.

2.7 Estándar de evaluación de calidad del software

La calidad del software según la norma *ISO 25000:2005* es el “*grado en el que el producto software satisface necesidades implícitas y explícitas cuando se usa bajo condiciones específicas*” [ISO, 2005]. Se han creado estándares para evaluación de la calidad del software, inicialmente el *ISO 9126* [ISO/IEC, 2001], un modelo de calidad que clasifica la calidad en un grupo de características y subcaracterísticas. Este estándar fue reemplazado por la norma *ISO 25000:2005* [ISO, 2005], *Software Quality Requirements and Evaluation SQuaRE*.¹

ISO 25000:2005 (Software Quality Requirements and Evaluation SQuaRE) es una serie de normas basadas en *ISO 9126* y en *ISO 14598* (Evaluación del software). Uno de los principales objetivos de la serie *SQuaRE* es la coordinación y armonización del contenido de *ISO 9126* y de *ISO 15939:2002 (Measurement Information Model)*. *ISO 15939* incluye un modelo de información que determina lo que se debe especificar durante la planificación, ejecución y evaluación de la medición. *SQuaRE* consta de 5 divisiones de calidad (Figura 2.3) [ISO, 2005].

¹<http://iso25000.com/index.php/normas-iso-25000/iso-25010>

2. ESTADO DEL ARTE



Figura 2.3: Divisiones del Estándar de evaluación de la calidad del software *SQaRE*

1. División de Administración de la Calidad: incluye la guía para *SQaRE* con la estructura y terminología, y la guía para planificar y administrar las evaluaciones del software.
2. División de Modelo de Calidad: describe el modelo de calidad, que categoriza las propiedades de la calidad interna y externa del producto software en ocho características (funcionalidad, usabilidad, mantenibilidad, portabilidad, compatibilidad y seguridad). Cada característica se compone de un conjunto de subcaracterísticas, e incluye el modelo de calidad en uso, que es el grado en el que un producto o sistema puede ser usado por usuarios específicos para cumplir sus necesidades y alcanzar sus objetivos específicos con efectividad, eficiencia, libre de riesgos y satisfacción en un contexto de uso específico. El modelo de calidad se representa en la Figura 2.4.
3. División de Medidas de Calidad: describe las métricas de calidad para la evaluación de la calidad interna, externa y en uso, incluyendo:
 - Modelo de referencia de las métricas: presenta el modelo de referencia común para los elementos de medición de la calidad y proporciona una

2.7 Estándar de evaluación de calidad del software



Figura 2.4: Modelo del estándar de evaluación de la calidad del software *SQuaRE*

guía de selección, desarrollo y aplicación de las medidas propuestas por las normas *ISO*.

- Medidas primitivas: definen y especifican las medidas base y derivadas a utilizar durante el desarrollo del software. Algunos ejemplos de estas medidas son: número de líneas de código, número de puntos de función, tiempo de tareas, número de tareas, número de fallos. Este conjunto de primitivas se utilizarán como entrada para el cálculo de los otros tres tipos de medidas.
 - Medidas de calidad en uso: define específicamente las métricas para realizar la medición de la calidad en uso del producto.
 - Medidas de calidad del sistema y producto software: define específicamente las métricas para realizar la medición de la calidad de productos y sistemas software.
 - Medidas de calidad de datos: define específicamente las métricas para realizar la medición de la calidad de datos.
4. División de Requerimientos de Calidad: normas que forman este apartado ayudan a especificar requisitos de calidad que pueden ser utilizados en el proceso de elicitación de requisitos de calidad del producto software a desarrollar o como entrada del proceso de evaluación. Para ello, este apartado se compone de la división de requisitos ISO/IEC 25030 - Quality requirements, que proporciona un conjunto de recomendaciones para realizar la especificación de los requisitos de calidad del producto software.

2. ESTADO DEL ARTE

5. División de Evaluación de la Calidad: define los procesos para desarrolladores, compradores, evaluadores y la documentación del módulo de evaluación. El proceso de evaluación para evaluadores detalla requisitos y recomendaciones para la valuación de software de forma que dicha evaluación sea fiable y comprensible; las características del proceso de evaluación son: repetibilidad, imparcialidad, reproductividad y objetividad.

En estos términos, el estándar de calidad *ISO/IEC 25000:2005* o *Software Product Quality Requirements and Evaluation (SQuaRE)* es un estándar que define la evaluación de la calidad de un producto software, cubriendo dos procesos principales: especificación de requisitos de calidad del software y evaluación de la calidad del software. *SQuaRE* provee la guía general, términos y definiciones, planificación y gestión, medidas y evaluación para definir todos los elementos requeridos para la evaluación de productos software.

2.8 Métricas de calidad del software

La representación de objetos en OWL es sintáctica y semánticamente muy similar a la descripciones de objetos, clases e instancias en Programación Orientada a Objetos. En la práctica, los programadores tienden a explicar la construcción de ontologías con la analogía de análisis de sistemas en ingeniería de software [Koide et al., 2005].

Se ha demostrado además que lenguajes de modelado orientado a objetos pueden estar fundamentados en teoría ontológica [Opdahl and Henderson-Sellers, 2001]. La similitud entre el diseño orientado a objetos y el desarrollo de ontologías permite usar los principios y métodos de métricas de software para el diseño de métricas de ontologías [Zhang et al., 2010].

Shyam R. Chidamber y Chris F. Kemerer, definen un conjunto de métricas para medir la calidad de aplicaciones orientadas a objetos [Chidamber and Kemerer, 1994]:

- *Weighted Methods Per Class (WMC)* es el número de métodos definidos en la clase. Se relaciona directamente con la complejidad de un objeto, donde los métodos son propiedades de objetos y la complejidad está definida por

la cardinalidad de su conjunto de propiedades, esta métrica es un indicador del tiempo y esfuerzo que se requiere para desarrollar y mantener un objeto. *Weighted Methods Per Class (WMC_{Onto})* = $\sum |c_i|$ donde $|c_i|$ es la complejidad estática de los métodos.

- *Depth of Inheritance Tree (DIT)* es la altura de la clase en el árbol de herencia. La altura de un nodo en el árbol se refiere a la longitud de la ruta más larga desde el nodo a la raíz del árbol. Indica cuántos ancestros de clase pueden potencialmente afectarla.
- *Number of children (NOC)* es el número de sub-clases inmediatas de una clase, equivalente al número de clases hijas inmediatas derivadas de una clase base.
- *Coupling between objects (CBO)* es el número de relaciones (no de herencia) acopladas con otras clases. Dos clases están acopladas cuando los métodos de una usan métodos o variables de instancias de la otra.
- *Response For a Class (RFC)* es el conjunto de respuesta a una clase, el conjunto de métodos que pueden potencialmente ser ejecutados en respuesta a un mensaje recibido por un objeto de la clase. RFC es el número de métodos en el conjunto. $RFC = |RS|$ donde RS es el conjunto de respuesta a una clase. El conjunto de respuesta puede ser expresado como $RS = M_i \cup \forall n R_i$ donde M_i son todos los métodos de la clase y R_i es el conjunto de métodos llamados por M_i .
- *Lack of Cohesion in Methods (LCOM)* considera una clase C_1 con métodos M_1, M_2, \dots, M_n . Sea $I_i =$ el conjunto de variables de instancia usadas por el método M_i , habiendo por tanto n conjuntos I_1, I_2, \dots, I_n . *LCOM* es el número de conjuntos disjuntos formado por la intersección de los n conjuntos. Esta métrica usa la noción del grado de similitud de los métodos. El grado de similitud en la clase C_1 está dado por $\sigma() = I_1 \cap I_2 \dots \cap I_n$. Si no hay instancias comunes, el grado de similitud es cero. *LCOM* está relacionado con las variables de instancia y los métodos de un objeto, por lo tanto es una medida de los atributos de un objeto.

2. ESTADO DEL ARTE

2.9 Clasificación de los métodos existentes de evaluación de ontologías, por subcaracterísticas de calidad

Existen numerosos enfoques de evaluación y definiciones de métricas de ontologías creadas por diferentes autores, los cuales abordan una o varias de las subcaracterísticas de calidad definidas en SQuaRE. La tabla 2.1 muestra las características y los principales autores que las tratan.

Autor \ Subcaracterística	Estruc.	Ad. Func.	Mante.	Opera.	Confia.	Transfe.	Compa.
[Tello and Gómez-Pérez, 2004]	X			X			
[Netzer et al., 2009]		X					
[Park et al., 2011]	X						
[Guarino and Welty, 2004]	X						
[Völker et al., 2008]	X						
[Corcho et al., 2004]	X						
[Wang et al., 2007]	X						
[Vrandečić and Sure, 2007]	X	X				X	
[Ma et al., 2009]		X					
[Gangemi et al., 2006]	X	X		X	X		
[Rogers, 2006]	X	X		X			

Tabla 2.1: Autores por Característica

Otros autores que han creado enfoques de evaluación de ontologías donde se tratan algunas de las subcaracterísticas definidas en SQuaRE:

- **Estructura:** Consistencia y redundancia ([Grimm and Wissmann, 2011, Baumeister and Seipel, 2006])
- **Adecuación funcional:** clasificación de instancias y análisis de texto ([Kepler et al., 2006, Bloehdorn et al., 2005]); clúster y similitud ([Maedche and Staab, 2002, Ding and Peng, 2004]); ; árboles de guía y decisión([Gajderowicz and Sadeghian, 2009]); índices y enlaces [Kara et al., 2012, Bao et al., 2006], Reutilización del conocimiento [Bao et al., 2007, Gruber, 1991], representación de resultados [Whetzel et al., 2006], esquema y valor de reconciliación [Hameed et al., 2004], búsqueda y consulta [Dragoni et al., 2010, Strasunskas and Tomassen, 2008].

2.9 Clasificación de los métodos existentes de evaluación de ontologías, por subcaracterísticas de calidad

- **Mantenibilidad:** estabilidad de la modificación [Orme et al., 2007], modularidad [Oh et al., 2011, Wang et al., 2007], Mutabilidad ([Flouris et al., 2008, Klein, 2004]); reuzabilidad [Simperl, 2009, Gruber, 1991].
- **Transferibilidad:** portabilidad [Gruber, 1993],
- **Fiabilidad:** disponibilidad ([Benjamins et al., 2004]), detección de errores ([Gómez et al., 2000]).
- **Compatibilidad:** interoperabilidad [Klein et al., 2002]

*The mind that opens up to a new idea
never returns to its original size.*

Albert Einstein

CAPÍTULO

3

Objetivos y metodología

3.1 Motivación

Los recientes progresos de la Web Semántica han generado un gran interés en el desarrollo y uso de ontologías en distintos dominios de aplicación. Consecuentemente se ha incrementado el número de ontologías desarrolladas para un mismo fin, generando dificultades a la hora de seleccionar la ontología adecuada para un propósito particular, debido a que no existen mecanismos de asistencia en la toma de decisiones informadas. Según Brank et al. [Brank et al., 2005], los objetivos principales de la evaluación de ontologías son *“asistir a los usuarios en la selección de la ontología que mejor se ajuste a los requerimientos y ayudar a los desarrolladores a evaluar las ontologías resultantes de un proceso de desarrollo con el fin de generar confianza para ser compartidas y reutilizadas”*.

Sin embargo, evaluar la calidad de la ontología es una tarea difícil debido a los múltiples criterios que requieren ser evaluados. Según Vrandečić, *“ningún método de los existentes puede garantizar que la ontología es de buena calidad”* [Vrandečić, 2010]. Del análisis del estado del arte se identificaron algunas limitaciones en los métodos de evaluación de ontologías:

3. OBJETIVOS Y METODOLOGÍA

- Los métodos existentes no incluyen todos los criterios necesarios para la evaluación de la ontología. Algunos se centran en la evaluación de la estructura, otros en el conocimiento del dominio y la cobertura, algunos combinan los dos aspectos y definen algunos criterios de evaluación de la calidad. Otros métodos definen métricas de calidad de la ontología.
- De los métodos estudiados, pocos han sido definidos, diseñados, implementados y validados experimentalmente.
- Los métodos que usan métricas no incluyen una trazabilidad entre los requisitos de una ontología y las métricas que se requieren para ser medidos.
- Ninguno de los métodos de evaluación es estándar ni se basa en estándares de calidad.

3.1.1 Hipótesis

La evaluación de la calidad de las ontologías puede ser mejorada a través de la adaptación de estándares de calidad del software, en los que se incluyan todos los criterios que deban ser evaluados en la ontología y la trazabilidad de sus requisitos a sus métricas.

3.1.2 Formulación de preguntas de investigación

¿Es posible adaptar un estándar de calidad del software para evaluar la calidad de las ontologías de forma objetiva y reproducible, y asistir así a los usuarios en la toma de decisiones informadas?

¿Se pueden incluir en un sólo método de evaluación objetivo, todos los criterios necesarios para evaluar la calidad de una ontología y hacer la trazabilidad entre sus requisito y sus métricas?

3.2 Objetivo

Crear un método de evaluación de la calidad de ontologías adaptando el estándar de calidad del software ISO/IEC 25000:2005 *Software Quality Requirements and*

Evaluation SQuaRE, que permita la trazabilidad entre requisitos y métricas de una ontología, con el fin de valorar de forma objetiva y reproducible sus características, y brindar así un mecanismo de asistencia a los usuarios y desarrolladores para la toma de decisiones informadas.

3.2.1 Objetivos específicos

- Realizar un análisis exploratorio del estado del arte de los métodos de evaluación de ontologías existentes.
- Adaptar el estándar de calidad del software ISO 25000:2005 o SQuaRE a las ontologías.
- Definir las divisiones de SQuaRE para la evaluación de ontologías.
- Diseñar e implementar una aplicación software que facilite la utilización del framework, la cual permitirá hacer evaluaciones de ontologías de diferentes dominios de aplicación
- Aplicar el framework desarrollado para la evaluación de ontologías en diferentes dominios de aplicación.
- Validar experimentalmente el framework.

3.3 Metodología

La metodología que se propone para llevar a cabo el objetivo anteriormente descrito se define a continuación:

3.3.1 Análisis del estado del arte

El desarrollo de la tesis se inició con el análisis exploratorio del estado del arte, en el que se analizaron los enfoques de evaluación, métricas y calidad de ontologías existentes. Dichos enfoques se pueden dividir en tres categorías: clasificación de la ontología, la exactitud formal de los contenidos representados en las ontologías y marcos de calidad de la ontología.

3. OBJETIVOS Y METODOLOGÍA

El análisis del estado del arte evidenció la falta de un método estandarizado que abarque un conjunto más amplio de criterios de calidad, y que logre la trazabilidad entre los requisitos y las métricas de la ontología para evaluar su calidad de una manera objetiva y reproducible.

El uso de estándares como modelos de referencia para definir la evaluación de la calidad es razonable, debido a los beneficios que proporciona como son: una especificación y un modelo comprensible de evaluación de la calidad, un lenguaje común de especificación de requisitos, entendible por usuarios, desarrolladores y evaluadores, una evaluación de la calidad basada en la observación y un proceso de evaluación reproducible.

Se propone la adaptación del estándar de evaluación de calidad del software ISO/IEC 25000:2005 *Software Quality Requirements and Evaluation (SQuaRE)*, un estándar ya existente en la comunidad de Ingeniería de Software para evaluar la calidad de ontologías, con el fin de buscar una estandarización.

3.3.2 Definición del *framework*

Para la definición del *framework* de evaluación de calidad de ontologías OQuaRE, se estudiaron las diferentes divisiones del estándar de calidad del software ISO/IEC 25000:2005 *Software Quality Requirements and Evaluation SQuaRESQuaRE* y su posible aplicación a las ontologías, dichas divisiones se expresaron en términos ontológicos por medio de las siguientes tareas:

1. Definición del modelo de calidad de OQuaRE
2. Definición de las métricas de calidad
3. Definición de la matriz de asociaciones entre métricas y subcaracterísticas de calidad
4. Definición del escalado de métricas al rango [1-5]

La aplicación del método requirió el diseño y desarrollo de una herramienta software que soportara la evaluación de la calidad de las ontologías de forma automática; la herramienta fue diseñada en JAVA.

3.3.3 Aplicaciones del *framework*

El *framework* fue aplicado a ontologías de diferentes dominios:

1. Se aplicó a dos ontologías de tipos de células con el objetivo de comparar las diferencias entre una evaluación manual y el *framework* OQuaRE aplicado de forma automática.
2. Se aplicó a 8 ontologías de unidades de medida para identificar las fortalezas y debilidades que presentaban en términos de subcaracterísticas de calidad.
3. Se usó para evaluar la utilidad de una guía de buenas prácticas de desarrollo de ontologías, la guía *Good Ontology Design (GoodOD)* [Boeker et al., 2012]. Para ello, se evaluaron un conjunto de ontologías desarrolladas por un grupo de estudiantes, antes y después de ser entrenados con la guía.
4. Se aplicó para analizar los cambios, en términos de calidad, de varias versiones de la ontología biomédica EDAM.
5. Se hizo una redefinición del método para ontologías financieras, específicamente la *Financial Business Ontology (FIBO)*, identificando las características y subcaracterística aplicables a este tipo de ontologías y creando las asociaciones métrica-subcaracterística.

3.3.4 Evaluación y validación del *framework*

Con el fin de validar el *framework* para identificar su utilidad se realizó un experimento con tres expertos en ontologías, quienes evaluaron una ontología de forma manual y luego con el *framework* OQuaRE de forma automática.

A lo largo del trabajo doctoral se validó el método OQuaRE a través de la comparación de sus resultados con los de otros métodos de evaluación:

1. Se compararon los resultados del *framework* OQuaRE para un conjunto de ontologías, con los resultados de dos métodos de evaluación basados en *Gold Standard* y *Competency Questions*.

3. OBJETIVOS Y METODOLOGÍA

2. Se compararon los resultados de OQuaRE con la evaluación manual realizada por un grupo de expertos en ontologías.
3. Se hizo una comparación de los resultados de OQuaRE con los resultados de *Bubastis*, una herramienta que analiza los cambios en las versiones de una ontología y entrega información de clases cambiadas, adicionadas y borradas.

Measurement is the first step that leads to control and eventually to improvement. If you can't measure something, you can't understand it. If you can't understand it, you can't control it. If you can't control it, you can't improve it.

H. James Harrington

CAPÍTULO

4

Framework OQuaRE (Ontology Quality Requirements and Evaluation)

4.1 Framework OQuaRE

Ontology Quality Requirements and Evaluation (OQuaRE) es un *framework* de evaluación de la calidad de las ontologías que adapta el estándar ISO/IEC 25000:2005 de calidad del software (SQuaRE). OQuaRE define todos los elementos necesarios para la evaluación de ontologías, especificando *el modelo de calidad* y *las métricas de calidad*. *El modelo de calidad* incluye los criterios de calidad de alto nivel (características) y los divide en criterios de más bajo nivel (subcaracterísticas), para luego ser medidos de forma cuantitativa por medio de *las métricas de calidad*.

Un trabajo previo con base en el estándar calidad del software ISO/IEC 9126 [Fernández-Breis et al., 2009b] definió parte de las subcaracterísticas relacionadas con el *framework*, y continuando con estas investigaciones se definió OQuaRE [Duque-Ramos et al., 2011].

4. FRAMEWORK OQUARE (ONTOLOGY QUALITY REQUIREMENTS AND EVALUATION)

En términos del estándar ISO/IEC 25000:2005, la calidad de la ontología se define como el *grado en el que la ontología satisface necesidades implícitas y explícitas cuando se usa bajo condiciones específicas*".

4.2 Modelo de Calidad

El modelo de calidad de OQuaRE consta de un grupo de características que se dividen en subcaracterísticas, siendo algunas de éstas adaptaciones del modelo de SQuaRE, mientras que otras han sido definidas por la comunidad investigadora de evaluación de ontologías.

Las características de OQuaRE son ocho: *estructura, fiabilidad, operabilidad, mantenibilidad, compatibilidad, transferibilidad, calidad en uso y adecuación funcional*.

La característica *estructura* no es tenida en cuenta en SQuaRE, pero ha sido añadida al modelo dado que la propia naturaleza de las ontologías, y el estado del arte en cuanto a su evaluación, indican que este criterio es importante para determinar su calidad.

Las otras siete características son una adaptación de características de SQuaRE. Este proceso de adaptación requirió el análisis de las subcaracterísticas de calidad de software y la selección de aquellas que aplicaban a las ontologías.

Las subcaracterísticas de OQuaRE son 44, las cuales están asociadas a cada una de las características:

- En las características *fiabilidad, operabilidad, mantenibilidad, compatibilidad, transferibilidad*, se incluyen un grupo de subcaracterísticas definidas en el modelo de SQuaRE. Algunas de ellas fueron descartadas por no aplicar a ontologías, tales como: *seguridad en el uso*, debido a que las ontologías son "sistemas" de uso público; *salud y seguridad de operador y público*, debido a que las ontologías no son un **producto** de riesgo; *daños comerciales*, debido a que no es un "**sistema**" que se comercie.
- En la característica *Adecuación funcional*, el modelo *SQuaRE* indica que sus subcaracterísticas son específicas del producto y deben ser definidas por los

expertos del dominio. Por lo tanto, para esta característica se incluyeron las subcaracterísticas definidas por Stevens y Lord [Stevens and Lord, 2009].

- En la característica *Estructura*, las subcaracterísticas se definen teniendo en cuenta la literatura y el conocimiento sobre la estructura de la ontología.

La Figura 4.1 muestra el modelo de calidad de OQuRE con todas sus características y subcaracterísticas.

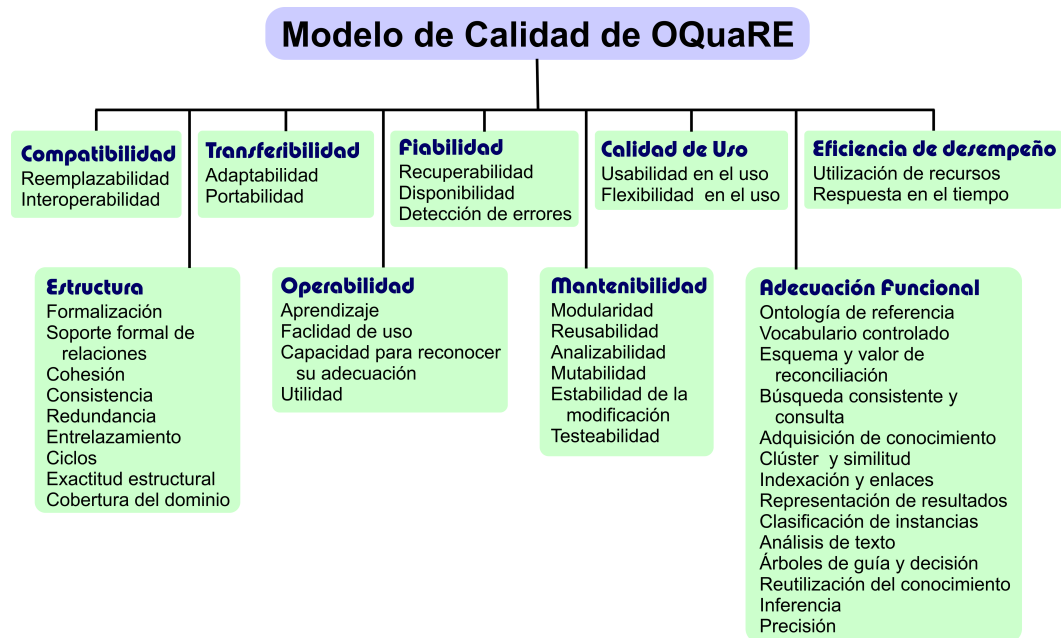


Figura 4.1: Modelo de Calidad de OQuRE

Las definiciones de las características, subcaracterísticas y las asociaciones entre ellas se presentan a continuación.

- *Estructura*: propiedades formales y semánticas importantes para la evaluación de la calidad de ontologías.
 - *Formalización*: una ontología tiene que estar construida sobre un modelo formal para apoyar el razonamiento.

4. FRAMEWORK OQUARE (ONTOLOGY QUALITY REQUIREMENTS AND EVALUATION)

- *Soporte formal de relaciones*: la mayoría de las ontologías solo tienen soporte formal para taxonomías. La utilización de otras teorías formales sería un indicador positivo para la ontología.
 - *Cohesión*: una ontología tiene una fuerte cohesión si las clases están fuertemente relacionadas.
 - *Consistencia*: una ontología es consistente si no contiene un conjunto de elementos descritos en ella que sean contradictorios o conflictivos.
 - *Redundancia*: todos los elementos de conocimiento de la ontología deben ser informativos, por lo que los elementos que no aportan información deberían ser eliminados.
 - *Entrelazamiento*: Esta subcaracterística mide la distribución de categorías con múltiples padres.
 - *Ciclos*: la existencia de ciclos en las relaciones semánticas son un signo de mal diseño, ya que pueden generar inconsistencias.
 - *Exactitud estructural*: trata sobre el uso correcto de los términos en la ontología.
 - *Cobertura del dominio*: el grado en el que la ontología cubre el dominio especificado.
- *Adecuación Funcional*: mide el grado de cumplimiento de los requisitos funcionales descritos por Stevens y Lord [Stevens and Lord, 2009].
- *Ontología de referencia*: las personas utilizan el conocimiento como un recurso de referencia de un dominio particular. Las ontologías modulan estos dominios, y las máquinas pueden explotar mejor el conocimiento de referencia de una ontología cuando tiene una estructura más explícita.
 - *Vocabulario controlado*: las etiquetas de las entidades ontológicas se utilizan para evitar la heterogeneidad, que complicaría el procesamiento y análisis de datos. Las ontologías proporcionan gestión de su terminología, unificando los términos para referirse a las entidades de conocimiento incluidas.

- *Esquema y valor de reconciliación*: las ontologías pueden proporcionar un modelo de datos común que puede aplicarse en distintos puntos de vista, favoreciendo su integración y reconciliación. Las ontologías facilitan que sea posible la interoperabilidad semántica, proporcionando un contexto semántico para datos e información.
- *Búsqueda consistente y consulta*: el carácter formal de las ontologías permite mejores métodos de consulta y búsqueda. La estructura de las ontologías puede guiar los procesos de búsqueda, ya que proporcionan un contexto semántico para evaluar cuáles son los datos buscados por los usuarios. Este contexto semántico no solo es proporcionado por los conceptos, sino también por todas las propiedades y axiomas computables de la máquina.
- *Adquisición de conocimiento*: las ontologías pueden ser vistas como plantillas para generar formularios mediante los cuales se adquieren las instancias.
- *Cluster y similitud*: las anotaciones de los datos con relación a la ontología se pueden utilizar para agrupar esos datos contra los aspectos de dicha ontología.
- *Indexación y enlaces*: las clases incluidas en las ontologías pueden servir como índice para recuperar la información rápidamente.
- *Representación de resultados*: las clases de la ontología sirven para analizar resultados complejos como los de los experimentos de *microarrays*.
- *Clasificación de instancias*: grado en el cual las instancias de una ontología pueden ser reconocidas como miembros de una cierta clase.
- *Análisis de texto*: la estructura de las ontologías ayuda a detectar asociaciones entre palabras o conceptos, y a clasificar tipos de palabra.
- *Árboles de guía y decisión*: las restricciones en una ontología reducen el espacio de posibilidades, lo cual es muy útil en dominios grandes y complejos como el de la biología y la bioinformática.
- *Reutilización del conocimiento*: es el grado con el que el conocimiento de una ontología puede ser usado para construir otras ontologías.

4. FRAMEWORK OQUARE (ONTOLOGY QUALITY REQUIREMENTS AND EVALUATION)

- *Inferencia*: es el grado con el que el modelo formal de una ontología puede ser utilizado por los razonadores para hacer explícito el conocimiento implícito.
- *Precisión*: es el grado en que la ontología provee los resultados correctos o especificados, con el grado necesario de exactitud.
- *Fiabilidad*: capacidad de una ontología de mantener su nivel de rendimiento en las condiciones estipuladas durante un periodo de tiempo determinado.
 - *Recuperabilidad*: es el grado con el que una ontología puede reestablecer un nivel de rendimiento específico y recuperar los datos afectados en caso de fallo.
 - *Disponibilidad*: es el grado con el que una ontología está operativa y disponible cuando se necesita.
 - *Detección de errores*: es el grado en el que la ontología permite a los usuarios para detectar fallos.
- *Operabilidad*: esfuerzo necesario y estimado para el uso de una ontología por un grupo de usuarios establecido o implícito.
 - *Aprendizaje*: es el grado con el que las ontologías permiten comprender su contenido y aplicación.
 - *Facilidad de uso*: es el grado en el que la ontología permite ser controlada y operada con facilidad.
 - *Capacidad para reconocer su adecuación*: el grado en el que la ontología permite a los usuarios reconocer si es apropiada para sus necesidades.
 - *Utilidad*: el grado en el que la ontología proporciona ayuda cuando los usuarios necesitan asistencia.
- *Mantenibilidad*: capacidad de las ontologías para ser modificadas debido a cambios en los entornos, en los requisitos o en las especificaciones funcionales.

- *Modularidad*: es el grado en el que las ontologías están constituidas por componentes separados permitiendo que el reemplazamiento de uno de los componentes tenga un impacto mínimo en el resto.
 - *Reusabilidad*: es el grado con el que parte de una ontología puede ser utilizado en más de una ontología, o en la construcción de otras partes.
 - *Analizabilidad*: es el grado con el que a la ontología se le puede diagnosticar deficiencias o inconsistencias, o se pueden identificar las partes modificadas.
 - *Mutabilidad*: la facilidad con la que una ontología puede ser modificada.
 - *Estabilidad de la modificación*: el grado con el que las ontologías pueden evitar efectos inesperados debido a modificaciones del software o del conocimiento.
 - *Testeabilidad*: el grado con el que las ontologías modificadas pueden ser validadas.
- *Transferibilidad*: capacidad de una ontología de ser transferida de un entorno a otro.
 - *Adaptabilidad*: el grado con el que una ontología se puede adaptar a distintos entornos, sin aplicar más medidas que las establecidas para el propósito particular de la ontología considerada.
 - *Portabilidad*: la facilidad con que una ontología puede ser transferida de un ambiente hardware o software a otro.
 - *Compatibilidad*: capacidad de dos o más ontologías para intercambiar información y/o realizar sus funciones al compartir el mismo hardware o software.
 - *Reemplazabilidad*: el grado con el que una ontología se puede utilizar en lugar de otra con el mismo propósito en el mismo entorno.
 - *Interoperabilidad*: el grado en que la ontología puede ser operable combinando su conocimiento con otras ontologías.
 - *Eficiencia de desempeño*:

4. FRAMEWORK OQUARE (ONTOLOGY QUALITY REQUIREMENTS AND EVALUATION)

- *Utilización de recursos*: grado en que la ontología proporciona respuestas apropiadas cuando ejecuta su función (consultas o razonamiento), bajo condiciones específicas.
- *respuesta en el tiempo*: el grado en que la ontología usa una cantidad apropiada de recursos cuando ejecuta su función bajo condiciones específicas.
- *Calidad en el uso*:
 - *Usabilidad en el uso*: se divide en: (1) efectividad en el uso, que es el grado en que los usuarios pueden alcanzar los objetivos específicos con exactitud y completitud en un contexto de uso específico; (2) *eficiencia en el uso*, que es el grado en el que los usuarios invierten la cantidad apropiada de recursos en relación con la efectividad alcanzada en un contexto de uso específico. (3) *satisfacción en uso*, el grado en el cual los usuarios están satisfechos con el contexto de uso especificado.
 - *Flexibilidad en el uso*: se divide en: (1) *conformidad en el contexto de uso* que es el grado de usabilidad en uso en el dominio para el que fue diseñada la ontología; (2) *extensibilidad en el contexto de uso*, que es el grado de usabilidad en uso en dominios más allá de los previstos inicialmente.

4.3 Métricas de Calidad

En SQuaRE se definen dos tipos de métricas de calidad, las métricas básicas y las métricas derivadas, de esta misma forma se plantea en OQuaRE la definición de las métricas para ontologías.

4.3.1 Métricas básicas

Las métricas básicas son aquellas que provienen directamente de la ontología, es decir, aquellas que pueden ser obtenidas por conteo manual, o de manera automática sin necesidad de hacer operaciones adicionales. Con el fin de calcular estas métricas, se representa la ontología como un grafo en términos matemáticos.

En las ontologías, las métricas básicas se dividen en tres grupos:

4.3.1.1 Métricas no jerárquicas

Corresponden a las métricas relacionadas con los elementos de la ontología como el número de clases, número total de instancias, total de anotaciones, número de usos de las propiedades, número de clases definidas, número de clases disjuntas y número de clases hoja.

4.3.1.2 Métricas jerárquicas (*asserted vs inferred*)

Son las métricas que se refieren a las relaciones de la ontología como padres directos de clases hoja, padres directos de clases, número de clases con más de un padre directo, número de relaciones, número de relaciones del nodo raíz y número de atributos.

4.3.1.3 Métricas de rutas

Son las métricas de los recorridos de la ontología vista como un grafo, como el número de rutas totales (todas las rutas desde las hojas hasta la raíz) y las longitudes de todas las rutas desde las hojas hasta raíz.

4.3.2 Métricas derivadas

Son aquellas que se calculan a partir de las métricas básicas. Para el caso de las ontologías se definen diferentes tipos de métricas, algunas provenientes de la ingeniería de software, específicamente de Programación Orientada a Objetos (POO), debido a que los conceptos de dicho paradigma de programación tales como: clases, propiedades, herencia, etc, son similares a los conceptos de ontologías; también se definen métricas que han sido desarrolladas exclusivamente para la Ingeniería de Ontologías por los expertos y se crean nuevas métricas que se consideran necesarias a la hora de evaluar una ontología.

4. FRAMEWORK OQUARE (ONTOLOGY QUALITY REQUIREMENTS AND EVALUATION)

4.3.2.1 Métricas adaptadas desde la Programación Orientada a Objetos (POO)

Las métricas adaptadas desde POO fueron:

- Acoplamiento - *Coupling between objects (CBO)*
- Profundidad del árbol de herencia - *Depth of inheritance tree (DIT)*
- Número de hijos - *Number of children (NOC)*
- Respuesta a una clase - *Response for a class (RFC)*
- Conteo ponderado de los métodos - *Weighted method count (WMC)*
- Número de métodos locales - *Number of local methods (NOM)*

Las cinco primeras han sido adaptadas de [Chidamber and Kemerer, 1994] y la última de [Li and Henry, 1993].

4.3.2.2 Métricas desarrolladas por la comunidad de ontologías

Otras métricas definidas en OQuaRE provienen de las desarrolladas por la comunidad de ontologías, principalmente aquellas que miden propiedades estructurales tales como:

- Riqueza de las relaciones (*RROnto*)
- Riqueza de los Atributos (*AROnto*)
- Riqueza de las propiedades (*PROnto*)
- Riqueza de las clases (*CROnto*)
- Cohesion - *Lack of Cohesion in Methods (LCOMOnto)*

Las cuatro primeras han sido adaptadas de [Tartir and Arpinar, 2007] y la última de [Yao et al., 2005].

Para el cálculo de las métricas se definió la siguiente notación:

- $C_1; C_2; \dots; C_n$ Clases de la ontología
- $R_{C_1}; R_{C_2}; \dots; R_{C_n}$ Relaciones de la clase C_i

- $P_{C_1}; P_{C_2}; \dots ; P_{C_n}$ Propiedades de la clase C_i
- $I_{C_1}; I_{C_2}; \dots ; I_{C_n}$ Instancias de la clase C_i
- $Sup_1; Sup_2; \dots ; Sup_n$ Superclases directas de la clase C_i

Estos conceptos pueden ser asociados con las primitivas del lenguaje OWL así: owl:Class se corresponde con las clases de la ontología, owl:DatatypeProperty y owl:ObjectProperty con las propiedades en general y, más específicamente, owl:ObjectProperty se corresponde con las relaciones. También se usarán otros conceptos de OWL como owl:Thing, el cual se corresponde con la clase raíz de la ontología. las clases que contengan en el modelo inferido owl:Nothing se corresponderán con las clases hoja.

A continuación se describen las métricas de OQuARE:

- *Lack of cohesion in methods* (LCOMOnto): la relación semántica y conceptual de las clases se puede utilizar para medir la separación de responsabilidades e independencia de los componentes de las ontologías. Se calcula mediante la siguiente fórmula:

LCOMOnto = $\sum path[|C(Leaf)_i|] / m$ donde *path* es la ruta desde la hoja $C(Leaf)_i$ hasta la raíz del árbol *Thing* y *m* el número total de rutas.

- *Weighted method count* (WMCOnto): promedio de la longitud de caminos hasta Thing. Se calcula mediante la siguiente fórmula:

WMCOnto = $\sum path[|C(Leaf)_i|] / \sum |C(Leaf)_i|$

- *Weighted method count 2* (WMCOnto2): promedio de la longitud de caminos hasta Thing. Se calcula mediante la siguiente fórmula:

WMCOnto 2 = $m / \sum |C(Leaf)_i|$ donde *m* es el número total de rutas.

- *Depth of subsumption hierarchy* (DITOnto): longitud del camino más largo desde la clase Thing a una clase hoja. Se calcula mediante la siguiente fórmula:

DITOnto = $Max(\sum D|C_i|)$

- *Number of ancestor classes* (NACOnto): promedio de clases ancestro por clase hoja. Trabaja con las superclases directas de las clases hoja. Se calcula mediante la siguiente fórmula:

NACOnto = $\sum |Sup_{C(Leaf)_i}| / \sum |C(Leaf)_i|$

4. FRAMEWORK OQUARE (ONTOLOGY QUALITY REQUIREMENTS AND EVALUATION)

- *Number of children* (NOCOnto): promedio de subclases directas. Es el número de relaciones dividido por el número de clases menos las relaciones de la clase Thing. Se calcula mediante la siguiente fórmula:

$$NOCOnto = \sum |R_{C_i}| / (\sum |C_i| - |Sup_{C(Leaf)_i}|)$$

- *Coupling between objects* (CBOOnto): número de clases relacionadas. Es la media de padres directos por clase menos las relaciones de la clase Thing. Se calcula mediante la siguiente fórmula:

$$CBOOnto = \sum |Sup_{C_i}| / (\sum |C_i| - |R_{Thing}|)$$

- *Response for a class* (RFCOnto): número de propiedades que pueden ser accedidas directamente desde una clase. Se calcula mediante la siguiente fórmula:

$$RFCOnto = (\sum (|P_{C_i}| + \sum |Sup_{C_i}|) / (\sum |C_i|)$$

- *Number of properties* (NOMOnto): promedio de uso de propiedades por clase. Se calcula mediante la siguiente fórmula:

$$NOMOnto = \sum |P_{C_i}| / \sum |C_i|$$

- *Properties Richness* (PROnto): número relaciones definidas en la ontología dividido por el número de relaciones y propiedades. Se calcula mediante la siguiente fórmula:

$$PROnto = \sum |R_{C_i}| / \sum (|R_{C_i}| + |P_{C_i}|)$$

- *Relation Richness* (RROnto): número de propiedades definidos en la ontología dividido por el número de relaciones y propiedades. Se calcula mediante la siguiente fórmula:

$$RROnto = \sum |P_{C_i}| / \sum (|R_{C_i}| + |P_{C_i}|)$$

- *Attribute richness* (AROnto): promedio de atributos por clase. Se calcula mediante la siguiente fórmula:

$$AROnto = \sum |Att_{C_i}| / \sum |C_i|$$

- *Relationships per class* (INROnto): promedio de relaciones por clase. Se calcula mediante la siguiente fórmula:

$$INROnto = \sum |R_{C_i}| / \sum |C_i|$$

4.4 Asociación de métricas de calidad al modelo de calidad

- *Class richness* (CROnto): promedio de instancias por clase. Se calcula mediante la siguiente fórmula:

$$CROnto = \sum |I_{C_i}| / \sum |C_i|$$

- *Annotation richness* (ANOnto): promedio de anotaciones por clase. Se calcula mediante la siguiente fórmula:

$$ANOnto = \sum |A_{C_i}| / \sum |C_i|$$

- *Tangledness* (TMOnto): promedio de clases con múltiples padres. Se calcula mediante la siguiente fórmula:

$$TMOnto = \sum |C(MDP)_i| / (\sum |C_i| - 1) \text{ donde } C(MDP)_i \text{ es el número de clases con más de un padre directo}$$

- *Tangledness 2* (TMOnto2): promedio de padres directo por clase, de las clases que tienen más de un padre directo. Se calcula mediante la siguiente fórmula:

$$TMOnto2 = \sum |(MDP)_i| / \sum |C(MDP)_i| \text{ donde } C(MDP)_i \text{ es el número de clases con más de un padre directo}$$

- *Direct Parent average* (DPOnto): promedio de padres directos por clase. Se calcula mediante la siguiente fórmula:

$$DPOnto = \sum |(MDP)_i| / \sum |C_i|$$

4.4 Asociación de métricas de calidad al modelo de calidad

4.4.1 Asociación de métricas y subcaracterísticas

Cada característica de OQuARE tiene asociada un conjunto de subcaracterísticas. La evaluación de cierta característica depende de sus subcaracterísticas asociadas. De manera similar, la evaluación de una subcaracterística depende de sus métricas asociadas. Las asociaciones característica-subcaracterística provienen del modelo SQuARE y las asociaciones métrica-subcaracterística han sido definidas con base en las recomendaciones y prácticas incluidas en el compendio de estándares y métricas de calidad de software [Lincke and Lowe, 2007]. Para aquellas características y subcaracterísticas no definidas en SQuARE como *estructura* y *adecuación funcional*,

4. FRAMEWORK OQUARE (ONTOLOGY QUALITY REQUIREMENTS AND EVALUATION)

Estructura	
Formalización	
Soporte para Relaciones Formales	RROnto
Redundancia	ANOnto
Exactitud Estructural	
Consistencia	
Entrelazamiento	TMOnto
Cohesión	LCOMOnto
Adecuación Funcional	
Ontología de Referencia	
Vocabulario Controlado	ANOnto
Esquema y Valor de Reconciliación	RROnto, AROnto
Búsqueda y Consulta Consistente	ANOnto, RROnto, AROnto, INROnto
Adquisición de Conocimiento	ANOnto, RROnto, NOMOnto
Clúster y Similitud	ANOnto, RROnto, AROnto
Indexado y Enlaces	RROnto, AROnto, INROnto
Representación de Resultados	AROnto, CROnto
Árboles de Guía y Decisión	INROnto, AROnto, TMOnto
Reutilización del Conocimiento	ANOnto, AROnto, INROnto, NOMOnto, LCOMOnto
Inferencia	RROnto, CROnto

Tabla 4.1: Asociación de subcaracterísticas y métricas de calidad(parte 1)

las asociaciones de las características han sido definidas por expertos. Las tablas 4.1 y 4.2 muestran las asociaciones existentes entre las subcaracterísticas y métricas de OQuaRE.

Las subcaracterísticas se asocian exclusivamente con una característica, sin embargo, las métricas se encuentran asociadas con múltiples subcaracterísticas. Por ejemplo, la métrica *CBOOnto* contribuye a las subcaracterísticas *modularidad* y *reusabilidad* de la característica *mantenibilidad*, pero también a las subcaracterística *adaptabilidad* de la característica *transferibilidad*, y *aprendizaje* de la característica *operabilidad*.

La tabla 4.3 muestra asociaciones entre subcaracterística y métricas, Cada celda contiene un signo más (+), menos (-) o estará vacía según se indica a continuación:

- Signo más (+): la contribución de la métrica es directamente proporcional a la puntuación de la subcaracterística, es decir, un valor mayor en la métrica

4.4 Asociación de métricas de calidad al modelo de calidad

Mantenibilidad	
Modularidad	WMCOnto, CBOnto
Reusabilidad	WMCOnto, DITOnto, NOMOnto, -NOCOnto, RFCOnto, CBOnto
Analizabilidad	WMCOnto, DITOnto, NOMOnto, LCOMOnto, RFCOnto, CBOnto
Mutabilidad	WMCOnto, DITOnto, NOMOnto, LCOMOnto, RFCOnto, CBOnto, NOCOnto
Capacidad de ser modificado	WMCOnto, LCOMOnto, RFCOnto, CBOnto, NOCOnto
Capacidad de ser probado	WMCOnto, DITOnto, NOMOnto, LCOMOnto, RFCOnto, CBOnto
Compatibilidad	
Reemplazabilidad	WMCOnto, DITOnto, NOMOnto, NOCOnto
Transferibilidad	
Adaptabilidad	WMCOnto, DITOnto, RFCOnto, CBOnto
Operabilidad	
Aprendizaje	WMCOnto, LCOMOnto, NOCOnto, CBOnto, NOMOnto
Capacidad de recuperación	WMCOnto, DITOnto, NOMOnto, LCOMOnto
Disponibilidad	LCOMOnto

Tabla 4.2: Asociación de subcaracterísticas y métricas de calidad (parte 2)

corresponde a una puntuación mayor en la subcaracterística.

- Signo menos (-): la contribución de la métrica es inversamente proporcional a la puntuación de la subcaracterística, es decir, un valor mayor de la métrica contribuye a una puntuación menor de la subcaracterística.
- Casilla vacía: no existe relación entre la subcaracterística y la métrica.

Según la tabla 4.3, si una métrica afecta de manera positiva (+) al cálculo de una subcaracterística se usará su puntuación obtenida, pero si la métrica afecta de manera negativa (-) se aplica el valor opuesto en la tabla, es decir, si toma el valor de 1 se aplica el 4, si toma 2, se aplica 3 y así sucesivamente.

Por ejemplo, en la tabla 4.3, *NOCOnto* influye positivamente en la puntuación de las subcaracterísticas *mutabilidad* y *estabilidad de la modificación*, pero lo hace de manera negativa a la subcaracterística *reusabilidad*.

4. FRAMEWORK OQUARE (ONTOLOGY QUALITY REQUIREMENTS AND EVALUATION)

Métrica \ Subcar.	WMCOnto	DITOnto	NOCOnto	RFCOnto	NOMOnto	LCOMOnto	CBOOnto
Modularidad	+						+
Reusabilidad	+	+	-	+	+		+
Analizabilidad	+	+		+	+	+	+
Mutabilidad	+	+	+	+	+	+	+
Estabilidad de la modificación	+		+	+		+	+
Testeabilidad	+	+		+	+	+	+

Tabla 4.3: Aporte (+ ó -) de las métricas a la subcaracterística mantenibilidad

Calcular las características y subcaracterísticas con base en los valores numéricos de las métricas requiere de la combinación de varias métricas que se presentan en diferentes rangos. Por ejemplo, métricas tales como *LCOMOnto* o *WMCOnto* producen un valor absoluto, mientras que métricas como *RROnto* y *AROnto* generan valores relativos.

Métrica \ Puntuación	1	2	3	4	5
LCOMOnto	> 8	(6-8]	(4,6]	(2,4]	≤ 2
WMCOnto	> 15	(11,15]	(8,11]	(5,8]	≤ 5
DITOnto	> 8	(6-8]	(4,6]	(2,4]	[1,2]
NACOnto	> 8	(6-8]	(4,6]	(2,4]	[1,2]
NOCOnto	> 12	(8-12]	(6,8]	(3,6]	[1,3]
CBOOnto	> 8	(6-8]	(4,6]	(2,4]	[1,2]
RFCOnto	> 12	(8-12]	(6,8]	(3,6]	[1,3]
NOMOnto	> 8	(6-8]	(4,6]	(2,4]	≤ 2
RROnto	[0,20] %	(20-40] %	(40-60] %	(60-80] %	>80 %
AROnto	[0,20] %	(20-40] %	(40-60] %	(60-80] %	>80 %
INROnto	[0,20] %	(20-40] %	(40-60] %	(60-80] %	>80 %
CROnto	[0,20] %	(20-40] %	(40-60] %	(60-80] %	>80 %
ANOnto	[0,20] %	(20-40] %	(40-60] %	(60-80] %	>80 %
TMOnto	> 8	(6-8]	(4,6]	(2,4]	(1,2]

Tabla 4.4: Escala 1 a 5 de las métricas de calidad

Las puntuaciones de las subcaracterísticas y características de calidad de SQua-

4.4 Asociación de métricas de calidad al modelo de calidad

RE son presentadas en rangos de 1 a 5, donde "1 significa no aceptable, 3 mínimamente aceptable y 5 excede los requerimientos" [ISO, 2005]. Por lo tanto, en OQuaRE se mapea el rango de valores de las métricas al rango 1 a 5 mediante los criterios establecidos en las recomendaciones de buenas prácticas de la comunidad de Ingeniería de Software para las métricas adaptadas del software, y las métricas de evaluación de ontologías. La escala propuesta se presenta en la tabla 4.4. Es de anotar que un valor alto en la métrica no corresponde directamente a un valor alto en la escala.

Finalmente, las métricas pueden tener diferente peso de acuerdo a la aportación a la subcaracterística específica, por lo tanto una subcaracterística se ve afectada en diferente medida por las métricas que la componen.

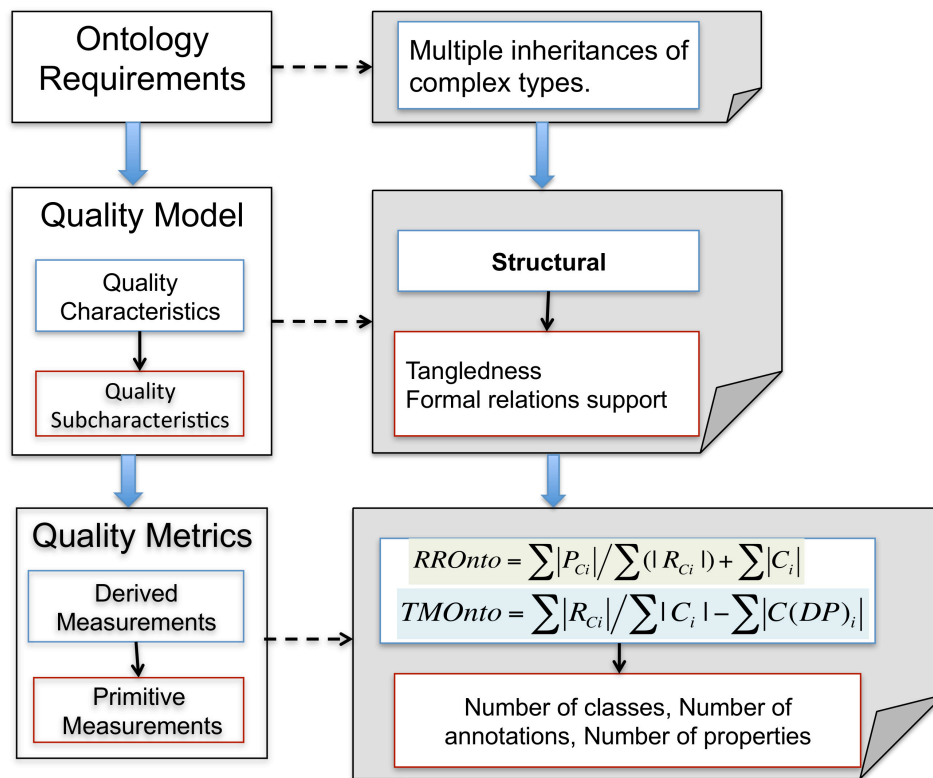


Figura 4.2: Ejemplo de OQuaRE

En la figura 4.2 se muestra la trazabilidad del Framework OQuaRE. En el lado izquierdo están las divisiones de requisitos, el modelo de calidad y las métricas de

4. FRAMEWORK OQUARE (ONTOLOGY QUALITY REQUIREMENTS AND EVALUATION)

calidad y en la parte derecha se ve un ejemplo de la trazabilidad, partiendo desde la división de requisitos hasta llegar a las métricas de calidad.

4.5 Ejemplo de OQuaRE

Partiendo de lo visto en los anteriores subapartados, con el fin de obtener una visión global y más clara del funcionamiento de OQuaRE, se presenta el proceso de evaluación de la calidad para la versión uno de la ontología EDAM ¹ a modo de ejemplo. Los pasos que sigue la herramienta para calcular de forma automática los resultados de las métricas, subcaracterísticas y características son:

- Se calculan las métricas básicas. Algunas de ellas son *número de clases*: 2078, *número de anotaciones*: 2318, *número de relaciones (SubClassOf)*: 2205, *número de propiedades (Número de uso de propiedades)*: 6473, *longitud de caminos hasta Thing* 7156, *número de caminos hasta Thing*: 1847.
- A partir de las métricas básicas se calculan las métricas derivadas con la fórmula correspondiente. Para el ejemplo, los resultados se presentan en la segunda columna de la tabla 4.5:
- Para obtener el valor de la métrica escalada se aplica la escala correspondiente definida en la tabla 4.4. Por ejemplo, LCOMOnto obtuvo un valor de 3,87, está en el intervalo (2, 4], por lo que le corresponde el valor “4” en la escala. Para el caso del ejemplo, los valores correspondientes en la escala se muestran en la tercera columna de la tabla 4.5.
- Los valores para las subcaracterísticas se calculan con los valores de las métricas escaladas multiplicadas por el peso asignado, dependiendo del aporte de la métrica a la subcaracterística así: $\text{Aprendizaje} = (K_1(WMCOnto) + K_2(LCOMOnto) + K_3(NOCOnto) + K_4(CBOnto) + K_5(NOMOnto) = 3,83$ donde K_5 representa el peso de la métrica para la subcaracterística específica. Para el ejemplo se asume que todos los pesos son iguales:

¹(<http://edamontology.org/page>)

4.5 Ejemplo de OQuaRE

Métrica	Valor real	Valor escalado
LCOMOnto	3.87	4
WMCOnto	4.18	5
DITOnto	11	1
NACOnto	1.10	5
NOCOnto	1.10	5
CBOnto	1.14 8	5
RFCOnto	4.38 12	4
NOMOnto	3.12	4
RROnto	0.75	4
AROnto	16.0	5
INROnto	1.06	5
CROnto	0.00	1
ANOnto	1.12	5
TMOnto	0.10	5

Tabla 4.5: Métricas de EDAM escaladas

Subcaracterística
Soporte para Relaciones Formales= $RROnto = 4$
Entrelazamiento= $TMOnto = 5$
Cohesión= $LCOMOnto = 1$
Operabilidad= 5
Aprendizaje= $(WMCOnto + LCOMOnto + NOCOnto + CBOnto + NOMOnto)/5 = 3,83$
Capacidad de recuperación= $(WMCOnto + DITOnto + NOMOnto + LCOMOnto)/4 = 2,50$
Esquema y Valor de Reconciliación= $(RROnto + AROnto)/2 = 4,5$
Vocabulario Controlado = $ANOnto = 5$
Adquisición de Conocimiento= $(ANOnto + RROnto + NOMOnto)/3 = 4,25$
Representación de Resultados= $(AROnto + CROnto)/2 = 3$
Árboles de Guía y Decisión = $(INROnto + AROnto + TMOnto)/3 = 5$
Inferencia= $(RROnto + CROnto)/2 = 2,5$

Tabla 4.6: Cálculo de subcaracterísticas de EDAM

4. FRAMEWORK OQUARE (ONTOLOGY QUALITY REQUIREMENTS AND EVALUATION)

- Se calculan las puntuaciones de las características: la puntuación de una característica concreta se calcula como la media ponderada de las puntuaciones de sus subcaracterísticas asociadas. En este ejemplo asumimos que los pesos son iguales. Para el ejemplo, se calculan las siguientes características:
 - Estructura = $(4 + 5 + 1)/3 = 3,33$, correspondientes a Soporte para Relaciones Formales, Entrelazamiento y Cohesión.
 - Adecuación funcional = $4,5 + 5 + 4,25 + 3 + 5 + 2,5 = 4,04$ correspondiente a Esquema y Valor de Reconciliación, Vocabulario Controlado, Adquisición de Conocimiento, Representación de Resultados, Árboles de Guía y Decisión e Inferencia. De igual forma se calculan el resto de los valores para las características.
- Finalmente, se calcula la puntuación global del modelo de calidad de OQuARE: la puntuación del modelo se calcula como la media ponderada de la puntuaciones de las características, siendo los pesos de dicha media iguales. En el ejemplo, tendríamos la suma de las características estructura y Adecuación funcional $3,33 + 4,04 = 3,7$ sería el valor global de la ontología.

*Somewhere, something incredible is
waiting to be known.*

Carl Sagan

CAPÍTULO

5

Aplicación del *framework* OQuaRE

En este capítulo se describe cómo se aplicó el *framework* en distintos contextos de evaluación de ontologías, y a las ontologías de varios dominios.

5.1 Ontologías de tipos de células

Objetivo: identificar la capacidad de OQuaRE para detectar diferencias entre la ontología original y su versión normalizada.

Descripción de las ontologías:

- Ontología de tipos de células *Cell Type Ontology* - (*CL*): fue diseñada como un vocabulario controlado estructurado para tipos de células, se construyó para ser utilizada por organismos modelo y otras bases de datos de bioinformática donde hay una necesidad de un vocabulario controlado de tipos de células.¹ En este experimento se usó la ontología original (oCTO) [Bard et al., 2005], que es una versión convertida de OBO a OWL.

¹<http://bioportal.bioontology.org/ontologies/CL>

5. APLICACIÓN DEL *FRAMEWORK* OQUARE

- *Normalized Cell Type Ontology (nCTO)*: la ontología normalizada es una versión axiomáticamente rica, creada por medio del proceso de normalización definido en ¹, donde muchas características se hacen explícitas. En la ontología normalizada, el razonador mantiene la jerarquía de afirmación múltiple con el fin de crear una ontología tal que, se define un eje de clases primitivas, con una superclase por clase como máximo, y hermanos disjuntos. [Fernández-Breis et al., 2009a].

Descripción del método: para este experimento se realizó la evaluación de las dos ontologías de forma automática con la herramienta de aplicación OQuaRE, de la que se obtuvo el grupo de valores cuantitativos de las características y subcaracterísticas de calidad luego de ser escaladas. Para la evaluación manual se tomaron datos de un estudio previo realizado en Fernández-Breis *et al* [Fernández-Breis et al., 2009b], en el cual la evaluación fue llevada a cabo por ocho expertos que dieron su calificación a las subcaracterísticas de la ontología con base en su experiencia y conocimiento. Finalmente, se crearon las tablas comparativas y los histogramas de frecuencia con los resultados de ambas evaluaciones.

Resultados: para ambas evaluaciones, manual y automática, los valores de las características de la ontología normalizada fueron más altos que los de la no normalizada, pero la evaluación automática (Figura 5.2) presentó valores más altos que la manual (Figura 5.1). En la ontología normalizada nCTO, la evaluación con OQuaRE presentó resultados entre 3.05 y 3.82, mientras que la evaluación de la ontología no normalizada oCTO presentó algunos valores por debajo de 3.0, como *operabilidad* con 2.92. Esto indica que la ontología requiere más esfuerzo para ser usada, entendida y aprendida, sin embargo, la versión normalizada mejoró esta puntuación obteniendo valores mayores que 3.0. ²

Conclusiones: el estudio indica que la ontología normalizada obtiene una puntuación más alta para las características de calidad que la original. Características como *mantenibilidad* incrementaron su valor, posiblemente debido a que nCTO es más modular y, por lo tanto, los componentes podrían ser modificados con mínimo impacto sobre el resto de la ontología. Sin embargo, la ontología original obtuvo

¹(<http://odps.sourceforge.net/odp/html/Normalisation.html>)

²Información adicional de este experimento se puede en el capítulo 7, sección 7.1

5.1 Ontologías de tipos de células

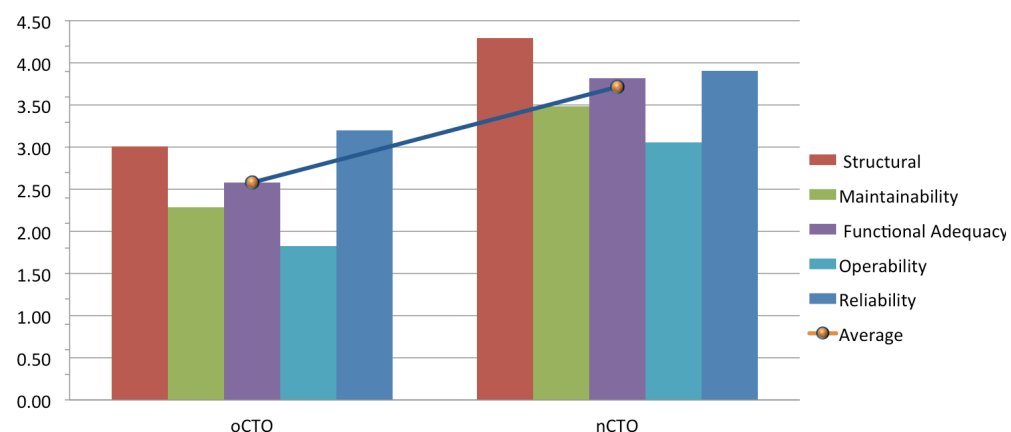


Figura 5.1: Evaluación manual de las ontologías de tipos de células

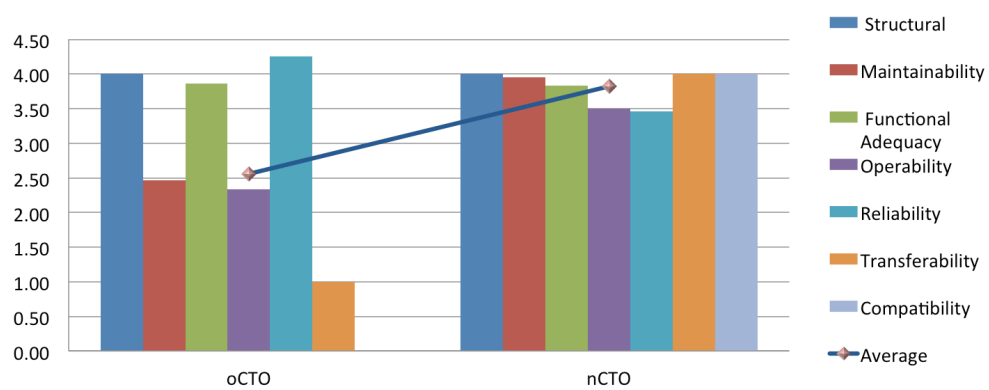


Figura 5.2: Evaluación automática de las ontologías de tipos de células

5. APLICACIÓN DEL *FRAMEWORK* OQUARE

una puntuación buena respecto a la *herencia múltiple* (cantidad de padres por clase) y a la *fiabilidad*.

5.2 Ontologías de unidades de medida

Objetivo: identificar fortalezas y debilidades de un conjunto de ontologías de unidades de medida, en términos de características y subcaracterísticas de calidad de OQuaRE.

Descripción de las ontologías: para este experimento y, teniendo en cuenta la estructura, el tamaño y el lenguaje en el que estaba desarrollada la ontología, se seleccionaron ocho ontologías de unidades de medida de distintos repositorios:

- *Measurement Units Ontology (MUOVOCAB)*: define clases y propiedades, proporcionando el vocabulario esencial para definir la semántica de las mediciones en las ontologías ¹.
- *Unified Code for Units of Measure (UCUM)*: el objetivo principal de esta ontología es acuñar URI para las unidades de medida más comunes, cualidades físicas y prefijos que pueden ser compartidas y reutilizadas. Cada unidad de medida está vinculada a una calidad física².
- *Gist Units of Measure Ontology (GISTUM)*: es una ontología diseñada principalmente para uso en negocios que incluye unidades de medida ³.
- *SWEET 2.0 Scientific Units Ontology (SCIUNITS)*, ontología para unidades científicas que es útil para la mayoría de aplicaciones de las ciencias físicas ⁴.
- *OpenMath (OPENMATH)*: representa la semántica de los objetos matemáticos, incluyendo unidades de medida⁵.

¹<http://idi.fundacionctic.org/muo/muo-vocab.html>

²<http://idi.fundacionctic.org/muo/ucum-instances.owl>

³<http://www.gist-ont.com/>

⁴<http://sweet.jpl.nasa.gov/2.0/sciUnits.owl>

⁵<http://www.openmath.org/index.html>

5.2 Ontologías de unidades de medida

- *OBO units of measurement (UNITPATO)*: es la ontología desarrollada por el OBO Foundry¹ para representar las unidades de medida².
- *Ontology of units of measure and related concepts (WURVOC)*: modela conceptos y relaciones importantes con un fuerte enfoque en las unidades, cantidades, medidas y dimensiones³.
- *Quantities Units Dimensions Values (QUDVSI)*: es una ontología de cantidades, unidades, dimensiones y valores⁴.

Descripción del método: para este experimento, la evaluación se realizó en dos fases utilizando el método comparativo: en la primera fase, las ontologías seleccionadas fueron evaluadas de forma manual, siguiendo unos criterios de evaluación predefinidos; en la segunda fase, se evaluaron las ontologías de forma automática. Con estos datos se generaron las tablas y las gráficas de histogramas de frecuencia y valores medios de las características, tanto para la evaluación manual (Figura 5.3) como para la automática (Figura 5.4).

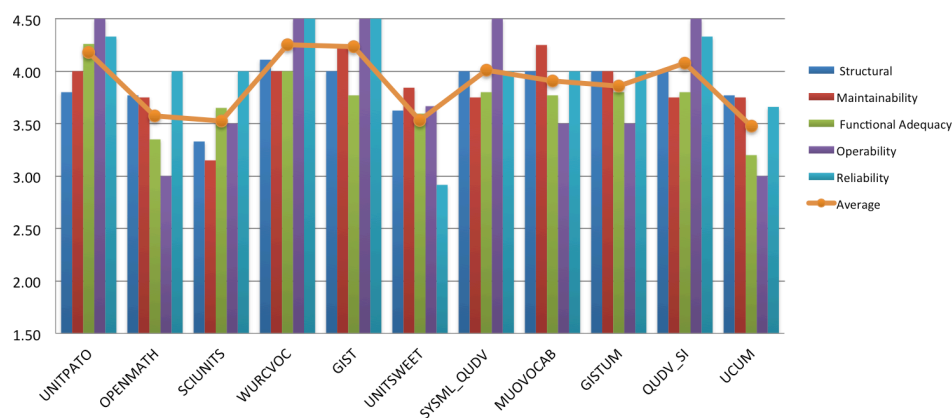


Figura 5.3: Evaluación manual de las ontologías de unidades de medida

¹<http://www.obofoundry.org/>

²<http://purl.org/obo/owl/UO>

³<http://www.wurvoc.org/vocabularies/om-1.6/>

⁴http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-qudv:qudv_owl

5. APLICACIÓN DEL *FRAMEWORK* OQUARE

Resultados: el experimento mostró que las ontologías con mejores valores fueron *MUOVOCAB* y *GISTUM*, con promedio cercano a 4.0. Las características *Estructura* y *Adecuación Funcional* presentaron el mejor valor (aproximadamente 4.0) para las ocho ontologías estudiadas. En particular, *QUDVSI* obtuvo las puntuaciones más bajas en la evaluación automática (Figura 5.4), con un promedio aproximado de 3.2, aunque la representación del conocimiento fue buena, indicando que su contenido podría ser usado para reemplazar contenido de otra ontología. *GISTUM* podría ser fácilmente evaluada y validada, su conocimiento podría ser reutilizado y adaptado a diferentes ambientes.

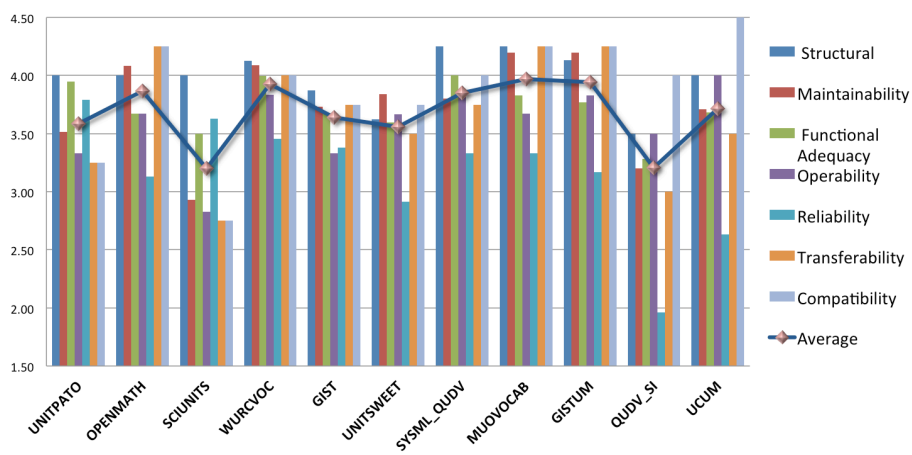


Figura 5.4: Evaluación automática de las ontologías de unidades de medida

Conclusiones: OQuaRE permite analizar fortalezas y debilidades de las ontologías en términos de *reutilización*, *modularidad*, *formalización* y otras subcaracterísticas de calidad. En este experimento se presentaron conclusiones acerca de la reutilización de las ontologías, la distribución de sus clases, la capacidad para ser cambiadas, testada, entre otros¹.

¹Información adicional de este experimento se puede encontrar en el capítulo 7, sección 7.1

5.3 Ontologías desarrolladas por medio de una guía de buenas prácticas

Objetivo: identificar la utilidad de OQuaRE para evaluar métodos y guías de buenas prácticas de construcción de ontologías.

Descripción de las ontologías: para este experimento se utilizaron 216 ontologías biomédicas de diferentes tópicos (*Process and Participation (PRO)*, *Immaterial object (IMM)* and *the Closure ODP (CLO)*, *Collective material entity (CME)*, *Information object (INF)* and *the Spatial disjointness ODP (SPA)*), desarrolladas por dos grupos de estudiantes, uno de ellos entrenado con la guía de buenas prácticas de desarrollo de ontologías *Good Ontology Design Guideline (GoodOD)*, y el otro no entrenado, y seis ontologías Gold Standard (una por tópico) desarrolladas por los creadores de la guía *GoodOD*. El entrenamiento de los estudiantes se describe en Boeker *et al.* [Boeker et al., 2013b]

Descripción del método: se aplicó el *framework* OQuaRE para evaluar las ontologías creadas por los estudiantes y por los autores de la guía, obteniéndose los valores de las subcaracterísticas escaladas. Luego se realizó el análisis estadístico para identificar el efecto del entrenamiento sobre las ontologías creadas, las diferencias entre medias de los grupos entrenados y no entrenados y las diferencias entre las ontologías creadas por los estudiantes y la ontología *Gold Standard*. Para el análisis se usaron varias técnicas estadísticas como *ANOVA de doble vía*, prueba *t-Student*, *análisis de clúster* y *componentes principales*.

Resultados: los resultados mostraron el efecto del entrenamiento con la guía *GoodOD* sobre la calidad de las ontologías, donde el 59 % de las subcaracterísticas de OQuaRE presentaron un efecto significativo (Figura 5.5). Varias métricas y subcaracterísticas de OQuaRE asociadas con puntos claves del entrenamiento mostraron cambios significativos.

Conclusiones: con este estudio se pudo ver que OQuaRE permitió obtener resultados más detallados sobre la calidad de las ontologías, mostrando mejoras en las ontologías creadas por los estudiantes entrenados, debido a cambios asociados al entrenamiento con la guía. Subcaracterísticas como *recuperabilidad*, *cohesión*

5. APLICACIÓN DEL *FRAMEWORK* OQUARE

y *disponibilidad* incrementaron su distancia al estándar del caso no entrenado al entrenado.¹

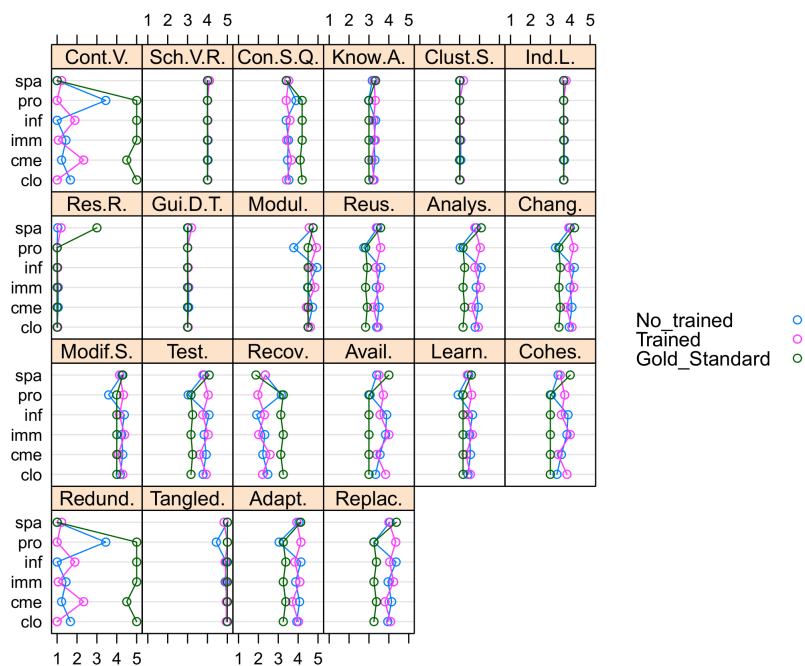


Figura 5.5: Diferencias significativas entre las ontologías de estudiantes entrenados, no entrenados y el *Gold Standard*

5.4 Versionado de Ontologías

Objetivo: evaluar el impacto de la evolución de una ontología en su calidad, identificando el efecto de los cambios entre versiones [Quesada-Martínez et al., 2015]².

Descripción de las ontologías: la ontología usada en este experimento fue *EDAM* (originalmente de *EMBRACE Data and Methods*), la cual es una ontología de conceptos familiares y bien establecidos que son importantes en bioinformática,

¹Información adicional de este experimento se puede en el capítulo 7, sección 7.2

²Este trabajo fue presentado en el ICBO 2015 [Quesada-Martínez et al., 2015], en esta sección se describirá brevemente

incluyendo tipos de datos e identificadores de datos, formatos de datos, operaciones y tópicos¹. *EDAM* contiene aproximadamente 2600 clases y cuenta con 18 versiones.

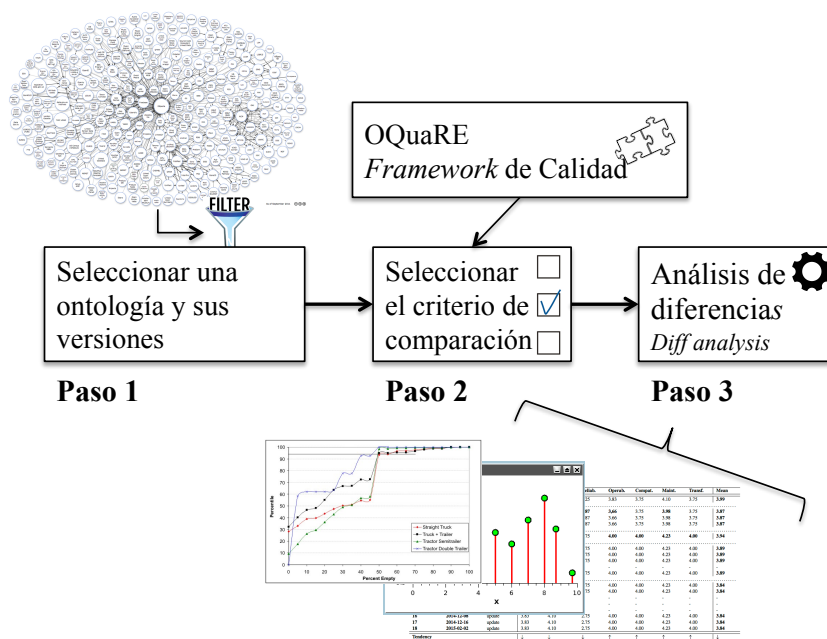


Figura 5.6: Pasos para evaluar la calidad de las versiones de una ontología con base en el *framework* OQuaRE.

Descripción del método: el método seguido para este experimento se muestra en la Figura 5.6 y se describe a continuación:

- Se seleccionaron del corpus las 18 versiones existentes de la ontología *EDAM* en orden secuencial.
- Se definieron los criterios de comparación, los cuales se miden a nivel de características, subcaracterísticas y métricas después de ser escaladas a valores entre 1 y 5, según la escala definida en OQuaRE.
- Se ejecutó el análisis de diferencias (*Diff analysis*) con el *framework* OQuaRE como criterio de comparación.

¹(<http://edamontology.org/page>)

5. APLICACIÓN DEL *FRAMEWORK* OQUARE

- Se analizaron los cambios entre versiones, tanto a nivel de métrica, como a nivel de subcaracterística.

Resultados: los resultados con OQuaRE mostraron que, de la versión v.4 a la v.5, los valores de *Operabilidad*, *Compatibilidad*, *Mantenibilidad* y *Transferabilidad* pasaron del nivel 3 a 4 en la escala; en las siguientes versiones se mantuvo el valor de 4. Este incremento se debió principalmente a la variación que presentaron las métricas *NOMOnto* y *RFCOnto* (Figura 5.7), relacionadas con el uso de propiedades. La subcaracterística que presentó el mayor decremento (de 4 a 1) fue *Soporte de relaciones formales*. Esto se debe principalmente a la métrica *RROnto*, la cual cambió dos niveles en la escala a lo largo de las versiones debido al aumento en el número de relaciones (18 %) entre versiones v.10 y v.11, y la disminución en el uso de propiedades (86 %) entre versiones v.4 y v.6. La característica *Estructura* decrementó por la métrica *TMOnto*, relacionada con la distribución de padres en la ontología, que para la v.4 fue de 10 %, mientras que para la v.5 fue del 24 %. Los valores de las características fueron más estables entre las versiones v.11 a v.18 con una desviación estándar de 0.04, mientras que en las versiones v1.0 a v4.0 la desviación fue de 0.17.

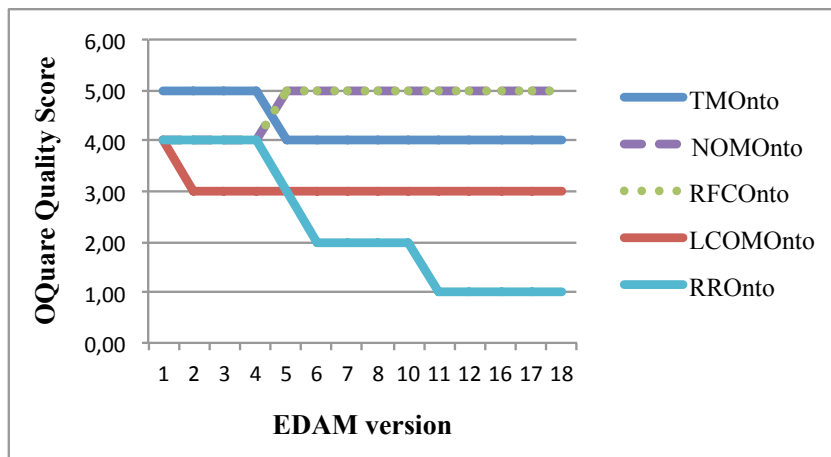


Figura 5.7: Representación de métricas que han cambiado su valor en la escala.

Finalmente, se aplicó *Bubastis* a las versiones de la ontología *EDAM*, para obtener el número de clases borradas, cambiadas y adicionadas por cada par de

versiones consecutivas, ver Figura 5.7 en la que se incluye el número de clases medida por OQuaRE. Con este resultado, se clasificaron las versiones en los cinco perfiles de actividad definidos por Malone et. al. [Malone and Stevens, 2013]: *Inicial, Ad hoc, Expandido, Refinado, Optimizado, Maduro e Inactivo*.

Conclusiones: la ontología *EDAM* inició en un estado *Refinado* (muchas clases existentes cambiadas, pero pocas clases adicionadas y borradas), luego bajó a *Expandido* (v.4 a v.7) y finalmente, evolucionó a un estado “*Maduro y Optimizado*” (estable en términos de calidad), según la clasificación definida por Malone *et al.* [Malone and Stevens, 2013]. Este resultado va en línea con los resultados obtenidos con OQuaRE para los niveles de calidad, los cuales revelaron estabilidad entre las versiones.

5.5 Ontologías financieras

Objetivo: identificar las características, subcaracterísticas y métricas aplicables a las ontologías financieras.

Descripción de las ontologías: la ontología financiera *Financial Industry Business Ontology (FIBO)*, creada y administrada por el *EDM council (Enterprise Data Management)* es un estándar de la ontología conceptual de negocios, que proporciona una descripción de la estructura y de las obligaciones contractuales de los instrumentos financieros, las personas jurídicas, los datos del mercado y los procesos financieros¹. En este experimento se analizaron las ontologías de negocios conceptuales *Business Conceptual Ontologies BCO*, las cuales son los estándares FIBO; y las ontologías operacionales derivadas de éstas, *Operational Ontologies (OO)*, las cuales se usan para desarrollo en aplicaciones de tecnología semántica.

Descripción del método: para este experimento se realizó un estudio de las subcaracterísticas, métricas y las asociaciones subcaracterística-métrica definidas en OQuaRE; con el apoyo del Líder de Arquitectura y estándar de la ontología *FIBO* se redefinieron las subcaracterísticas para ontologías financieras, se incluyeron los *pitfalls* de *Ontology Pitfall Scanner OOPS!* [Poveda-Villalón et al., 2010, Poveda-Villalón et al., 2012] como métricas de calidad asociadas a cada subcaracterística de

¹<http://www.edmcouncil.org/financialbusiness>

5. APLICACIÓN DEL *FRAMEWORK* OQUARE

OQuARE y, adicionalmente, se incluyeron las asociaciones subcaracterística-métrica específicas para ontologías de negocios.

Resultados: este experimento dio como resultado la lista de subcaracterísticas-métricas con los *pitfalls* y métricas de OQuARE aplicables para ontologías de negocios. Algunas de las asociaciones se presentan a continuación ¹

■ Estructura:

- *Formalización:* aplica para ontologías conceptuales y operacionales, el *pitfall* asociado es el *P23*, que está relacionado con el uso incorrecto de los elementos de la ontología.
- *Soporte formal de relaciones:* *RROnto* sólo aplica para ontologías operacionales.
- *Consistencia:* aplica para ambos tipos de ontologías, conceptual y operacional, el *pitfall* asociado es el *P22*, que se refiere a la convención de nombres usada en la ontología.
- *Redundancia:* aplica sólo para ontología operacional, y tiene asociados los *pitfalls* *P2* y *P32*, que se refieren a las etiquetas de las clases y a las clases sinónimas.
- *Corrección estructural:* aplican los *pitfalls* *P5*, *P14*, *P15*, *P16*, *P17*, *P19*, *P23*, *P25*, relacionados con: uso incorrecto de restricción universal, definición errónea de relaciones inversas, uso incorrecto de los elementos de la ontología, etc. Aplican para las ontologías conceptuales.

■ Adecuación funcional:

- *Vocabulario Controlado:* aplica para ontologías conceptuales, está asociada con *ANOnto* de OQuARE y *P1*, *P2* y *P9 de OOPs!*. Estas métricas se refieren a clases sinónimas y a falta de inclusión de información relevante para la ontología.

¹La tabla completa puede ser vista en <https://docs.google.com/document/d/1ErbZV0IFj890lHFcnygsw6n93dxublAamOu9oBnHdOo/edit#>

- *Esquema y valor de reconciliación*: aplica para ontologías operacionales y conceptuales, las métricas de OQuaRE asociadas son *RROnto* y *AROnto* y no tiene *pitfalls* asociados.
 - *Adquisición de conocimiento*: las métricas asociadas de OQuaRE son *ANOnto*, *NOMOnto* y *ANOnto*, y los *pitfalls* son *P9*, *P14-18* y *P23*, relacionados con información importante no incluida en la ontología, uso incorrecto de los elementos de la ontología, etc.
 - *Indexación y enlaces*: aplica para ontologías conceptuales y operacionales, las métricas asociadas de OQuaRE son *RROnto*, *AROnto* e *INROnto* y los relacionados con las relaciones equivalentes definidas erróneamente, clases equivalentes no definidas o erróneas. Los *pitfalls* asociados son *P21*, *P30*, *P31* y *P27*.
 - *Árboles de guía y decisión*: las métricas de OQuaRE que aplican son *AROnto*, *INROnto*, *TMOnto*. En el análisis se identificaron otras tres métricas de OQuaRE aplicables: *RROnto*, *CROnto*, *NOMOnto*. Se pueden aplicar todos los *pitfalls* excepto los que están relacionados con anotaciones.
 - *Precisión*: se identificaron dos métricas de OQuaRE que aplican a esta subcaracterística, *DITOnto* e *INROnto*, y los *pitfalls* *P23*, *P27-29* y *P5* aplican tanto para ontologías conceptuales como para operacionales.
- *Mantenibilidad*
 - *Analizabilidad*: aplica para ambos tipos de ontologías, operacionales y conceptuales. El *pitfall* asociado es *P19*, que está relacionado con el dominio y rango de las propiedades. Las métricas de OQuaRE aplicables son: *LCOMOnto*, *WMCOnto*, *CBOnto*, *DITOnto*, *RFCOnto*, *NOMOnto*,
 - *Estabilidad de la modificación*: aplica para ontologías operacionales y conceptuales, las métricas de OQuaRE asociadas son *WMCOnto*, *CBOnto*, *RFCOnto*, *COMOnto*, *NOCOnto*, pero no tiene *pitfalls* asociados.
 - *Operabilidad*

5. APLICACIÓN DEL *FRAMEWORK OQUARE*

- *Aprendizaje*: aplica para ontologías conceptuales y sus métricas asociadas son *LCOMOnto*, *WMCONto*, *CBOnto*, *RFCOnto*, *NOMOnto*, *AROnto*, *ANOnto*, y el *pitfall P8* que se refiere a las anotaciones perdidas.
- *Pertinencia del desempeño*: aplica para ontologías conceptuales y está asociada a la métrica *ANOnto* y a los *pitfalls P8, P11 Y P20*, los cuales están relacionados con las anotaciones y el dominio y rango de las propiedades.

Conclusiones: las características, subcaracterísticas y métricas de OQuaRE son aplicables a ontologías de negocios. Para hacer una medición efectiva de estas ontologías se requiere generar las asociaciones métricas-subcaracterística, describir algunas de las subcaracterísticas en términos de ontologías de negocios y descartar algunas de las métricas y subcaracterísticas por no aplicar a ontologías de negocios.

Este trabajo permitió identificar la utilidad de OQuaRE en otros contextos de aplicación y, además, la inclusión de las métricas de *OOPS!* mostró que se pueden asociar las métricas semánticas a las subcaracterísticas permitiendo combinar en un sólo método, las métricas estructurales con las métricas semánticas.

Este trabajo fue ejecutado en el marco del *Hackaton Clinic* presentado en el *Ontology Summit 2013* (Evento anual de la comunidad de ontologías), coorganizado por *Ontolog*, *NIST*, *NCOR*, *NCBO*, *IAOA* y *NCO-NITRD*, donde se obtuvo el primer premio *IAOA* al mejor *Hackaton Clinic* (2013),

The logic of validation allows us to move between the two limits of dogmatism and skepticism.

Paul Ricoeur

CAPÍTULO

6

Validación del *framework* OQuaRE

La validación se realizó a través de la comparación de resultados con otros métodos de evaluación. En el primer estudio se compararon los resultados con la evaluación manual; en el segundo se compararon con los resultados de otros dos métodos de evaluación, uno basado en un *Gold Standard* y otro en técnicas de *Competency Questions*.

La evaluación de OQuaRE se realizó por medio de un experimento en el que participaron tres expertos en ontologías biomédicas. Esta evaluación permitió identificar los aspectos positivos y negativos del método en términos de la relevancia y utilidad del *modelo de calidad* y *las métricas de calidad* y de la utilidad y adecuación de las asociaciones subcaracterística-métrica.

6.1 Comparación con evaluaciones humanas

Objetivo: comparar los resultados de dos evaluaciones realizadas por humanos con la evaluación realizada con OQuaRE.

Descripción de las ontologías: esta validación se llevó a cabo como parte de los experimentos descritos en las *secciones 5.1 y 5.2*, en los que se evaluaron dos ontologías de tipos de células: oCTO y nCTO, y ocho ontologías de unidades de Medida: *Measurement Units Ontology (MUOVOCAB)*, *Quantities Units Dimensions Values (QUDVSI)*, *OBO units of measurement (UNITPATO)*, *Ontology of units of measure and related concepts (WURVOC)*, *Gist Units of Measure Ontology (GISTUM)*, *Unified Code for Units of Measure (UCUM)*, *SWEET 2.0 Scientific Units Ontology (SCIUNITS)*, *OpenMath (OPENMATH)*.

Descripción del método: las ontologías de tipos de células se evaluaron con una aplicación JAVA diseñada para OQuaRE y los resultados se compararon con una evaluación manual realizada anteriormente en Fernández-Breis *et al.* [Fernández-Breis et al., 2009b]. En esta evaluación, las personas calificaron cada una de las subcaracterísticas de acuerdo a su experiencia y conocimiento. La evaluación manual de las ontologías de unidades de medida se realizó como parte del experimento descrito en la *sección 5.2*. Se evaluaron ocho ontologías de unidades de medida de forma manual y a continuación de forma automática.

Resultados: para las ontologías de tipos de células, los resultados de la evaluación automática con OQuaRE obtuvieron valores mayores a los de la evaluación de los humanos en un grado aceptable de diferencia, y las características evaluadas tuvieron, en su mayoría, un comportamiento similar en ambas evaluaciones, como se muestra en la Figura 6.1. Para las ontologías de unidades de medida, los resultados obtenidos para el método manual estaban entre 3.4 y 4.2, mientras que en el método automático estaban entre 3.3 y 4.0. Aunque el método automático obtuvo valores superiores al manual, éstos fueron muy similares (Figura 6.2).

Conclusiones: los resultados de las evaluaciones manuales de los dos experimentos fueron consistentes con los resultados de la evaluación automática de OQuaRE y por lo tanto, OQuaRE presenta una buena aproximación para la evaluación de ontologías ¹.

¹Información adicional de este experimento se puede en el capítulo 7, sección 7.1

6.1 Comparación con evaluaciones humanas

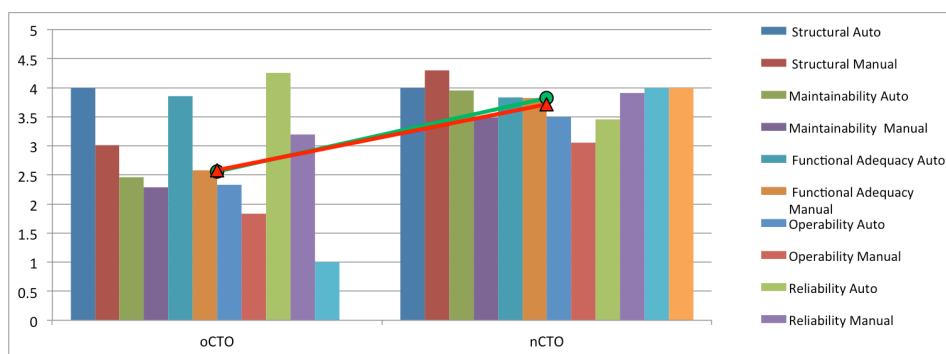


Figura 6.1: Comparación de la evaluación manual y automática para ontologías de tipos de células

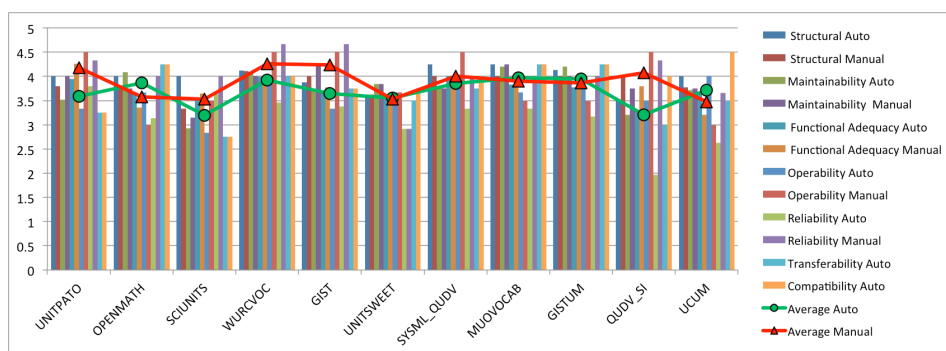


Figura 6.2: Comparación de la evaluación manual y automática para ontologías de unidades de medida

6.2 Comparación con resultados de otras técnicas

6.2.1 Validación con *Gold Standard*

Objetivo: comparar los resultados de OQuaRE con los de una evaluación de similitud entre las ontologías desarrolladas por los estudiantes y la ontología *Gold Standard* desarrollada por los creadores de la guía GoodOD [Boeker et al., 2012].

Descripción del método: la metodología usada para el experimento fue aplicar el *framework* OQuaRE a la ontología *Gold Standard* y a las ontologías desarrolladas por dos grupos de estudiantes, uno de ellos entrenado con la guía GoodOD y otro sin entrenamiento. Con los valores obtenidos se hizo un análisis multivariado de distancias medias de los estudiantes entrenados y no entrenados al *Gold Standard* para los seis tópicos (*PRO, CME, CLO, IMM, INF and SPA*). Luego se aplicó el método de clasificación y ordenación para identificar las similitudes entre las subcaracterísticas de los estudiantes entrenados y no entrenados con respecto a las distancias al *Gold Standard*. Para la clasificación se usó clúster jérarquico con base en la distancia euclídea entre subcaracterísticas y el método de varianza mínima de Ward [Ward, 1963]. Estos resultados se compararon con los resultados del estudio realizado por los creadores de la guía [Grewe et al., 2012], quienes calcularon el efecto del entrenamiento como la diferencia de las distancias medias del grupo al *Gold Standard*; estas diferencias significativas fueron testeadas por medio del *T-test* para muestras independientes.

Resultados: los resultados mostraron que no existe evidencia de que las ontologías desarrolladas por los estudiantes entrenados fueran más similares al *Gold Standard* que aquellas desarrolladas por los no entrenados. Sin embargo, al analizar cada uno de los tópicos, sí se encontraron diferencias significativas en ambas direcciones (positivas y negativas) para 5 tópicos; *PRO, IMM, CLO, CME e INF*. El análisis de clúster mostró que 19 subcaracterísticas fueron más similares al *Gold Standard* después del entrenamiento y 10 fueron menos similares para todos los tópicos excepto para *SPA*, para el cual, 18 subcaracterísticas presentaron valores más similares al *Gold Standard* y 11 menos similares.

Conclusiones: los resultados de la evaluación con OQuaRE concordaron con los obtenidos en el estudio previo, en el que se evaluó la similitud entre las ontologías desarrolladas por los estudiantes y el *Gold Standard*. En dicho estudio no se

6.2 Comparación con resultados de otras técnicas

confirmó que las ontologías mejoraran con el entrenamiento a nivel general, igual que en OQuaRE, pero sí se detectó un posible efecto del tópico y de la tarea de evaluación en el entrenamiento [Boeker et al., 2013b]. El análisis por tópico en la evaluación con OQuaRE confirmó este efecto, donde el *framework* OQuaRE mostró resultados similares a los presentados por el método de similitud al *Gold Standard*, y además permitió profundizar más en el estudio, logrando demostrar diferencias que no se habían encontrado con el estudio previo ¹.

6.2.2 Validación con *Competency Questions*

Objetivo: comparar los resultados de OQuaRE con los resultados de una evaluación basada en *Competency Questions*, la cual fue diseñada por los desarrolladores de la guía *GoodOD* [Boeker et al., 2013a].

Descripción del método: se aplicó OQuaRE a las ontologías desarrolladas por ambos grupos de estudiantes, entrenados y no entrenados con la guía *GoodOD* y se ejecutó el análisis estadístico descrito en la sesión anterior. Luego se compararon los resultados con los obtenidos en el estudio previo basado en *Competency Questions* [Boeker et al., 2013a] desarrollado por los creadores de la guía.

Resultados: en la evaluación realizada con *Competency Questions* se encontró un efecto del entrenamiento para los tópicos *PRO* y *CME*; en el estudio con OQuaRE se encontró que el efecto del entrenamiento fue significativo para 17 subcaracterísticas en *PRO*, 11 en *CME* y seis en *INF*.

Conclusiones: la evaluación con OQuaRE arrojó resultados similares a los de la evaluación con *Competency Questions*, siendo OQuaRE más específico en la información; ambas evaluaciones mostraron efecto del entrenamiento en los mismos tópicos, *PRO* y *CME* pero en la evaluación con *Competency Questions*, el efecto fue significativo solamente para *PRO*, mientras que en OQuaRE fue significativo para los dos tópicos *PRO*, *CME* siendo más significativas para *PRO*. Además, *PRO* presentó más homogeneidad en los datos de estudiantes entrenados que en los no entrenados. Con OQuaRE también se identificó un pequeño efecto en el tópico *INF*.

¹Información adicional de este experimento se puede en el capítulo 7, sección 7.2

6.3 Evaluación externa del modelo de calidad

Objetivo: evaluar completitud, usabilidad y adecuación del *modelo de calidad* y las *métricas de calidad* de OQuaRE, además, medir la utilidad del proceso de evaluación automática.

Descripción del método: la metodología usada fue *Goal Question Metric (GQM)*, que es un método de evaluación basado en objetivos que va desde los objetivos de la evaluación hasta el análisis de resultados [van Solingen and Egon, 1999]. Con base en este método se diseñó el experimento, se generaron los artefactos requeridos para él, se recolectaron los datos y se hizo el análisis estadístico de los resultados. Para la evaluación se contó con la participación de tres expertos en ingeniería de ontologías biomédicas. El experimento se llevó cabo en dos fases: en la primera, se evaluó la dificultad y adecuación de cada una de las subcaracterísticas de calidad; en la segunda, se identificaron la relevancia y utilidad de las métricas asociadas a una subcaracterística específica. Adicionalmente, se evaluaron las diferencias entre los métodos manual, manual asistido y automático. En el método manual, la evaluación se hace solamente con base en el conocimiento del experto quien define los valores de cada métrica; en el método asistido, la evaluación se hace de forma manual pero se cuenta con las definiciones de las métricas y de las subcaracterísticas y los valores de las métricas como soporte a evaluación; y finalmente, en el método automático, la evaluación se hace a través de la herramienta JAVA desarrollada para calcular los valores de las métricas y características de OQuaRE, sin intervención de los expertos.

Resultados: los resultados mostraron que cuatro subcaracterísticas fueron difíciles y no adecuadas, 23 fueron apropiadas pero difíciles y 19 subcaracterísticas apropiadas y fáciles (Figura 6.3). A nivel de métricas, se encontró que todas eran relevantes y útiles para alguna de las subcaracterísticas a las que pertenecían: 11 asociaciones se encontraron muy útiles y relevantes, 32 asociaciones fueron de utilidad y relevancia media y las otras 30 asociaciones tenían relevancia y utilidad bajas.¹

Las medias de los tres métodos manual, manual asistido y automático no fueron significativamente diferentes a un nivel de significación del 0.05 (ANOVA de una

¹Información complementaria se puede ver en el capítulo 7, sección 7.3

6.3 Evaluación externa del modelo de calidad

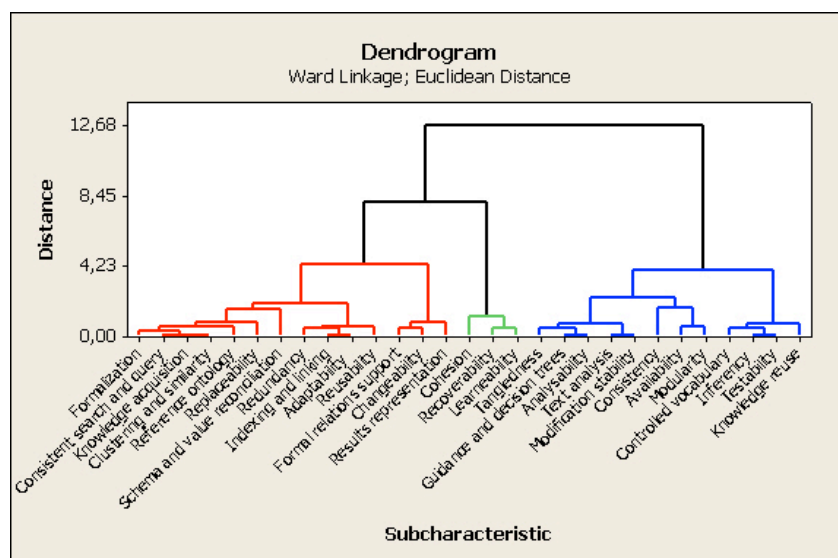


Figura 6.3: Clasificación de subcaracterísticas en un clúster

vía, con $p\text{-valor}=0.05$)

Conclusiones: el experimento mostró que algunas métricas fueron difíciles de entender por los expertos, debido a que no eran específicas de ontologías OWL. Sin embargo, el objetivo de OQuARE es evaluar ontologías de distintos formatos y diferente grado de formalización. Los expertos hicieron algunas sugerencias para la mejora del método, tales como: la creación de perfiles para evaluar ontologías en diferentes contextos de uso. Esta sugerencia está en línea con el método SQuARE, el cual define los requisitos de calidad que serían los perfiles indicados por los expertos. Otras sugerencias apuntaban a la inclusión de nuevas asociaciones subcaracterística métrica y a la eliminación de algunas existentes ¹.

¹Información complementaria se puede ver en el capítulo 7, sección 7.3

Publicaciones que componen esta tesis

7.1 *OQuaRE: A SQuaRE based Approach for Evaluating the Quality of Ontologies*

Título	OQuaRE: A SQuaRE based Approach for Evaluating the Quality of Ontologies
Autores	Astrid Duque Ramos y Jesualdo Tomás Fernández Breis
Páginas	41-58
Volumen, Número	Vol. 43, No. 2
Tipo	Artículo científico
Revista	Journal of Research and Practice in Information Technology
Factor de Impacto (2011)	0.105
Editor	Australian Computer Society Inc.
Año	2011
ISSN	443-458X
DOI	
URL	http://ws.acs.org.au/jrpit/JRPITVolumes/JRPIT43/JRPIT43.2.159.pdf
Estado	Publicado

7. PUBLICACIONES QUE COMPONEN ESTA TESIS

ABSTRACT

The development of the Semantic Web has provoked an increasing interest in the development of ontologies. There are, however, few mechanisms for guiding users in making informed decisions on which ontology to use under given circumstances. In this paper, we propose a framework for evaluating the quality of ontologies based on the SQuaRE standard for software quality evaluation. This method requires the definition of both a quality model and quality metrics for evaluating the quality of the ontology. The quality model is divided into a series of quality dimensions or characteristics, such as structure or functional adequacy, which are organized into subcharacteristics, such as cohesion or tangledness. Thus, each subcharacteristic is evaluated by applying a series of quality metrics, which are automatically measured. Finally, each characteristic is evaluated by combining values of its subcharacteristics. This work also includes the application of this framework for the evaluation of ontologies in two application domains.

7. PUBLICACIONES QUE COMPONEN ESTA TESIS

7.2 *Evaluation of the OQuaRE framework for ontology quality*

Título	Evaluation of the OQuaRE framework for ontology quality
Autores	Astrid Duque-Ramos, Jesualdo Tomás Fernández-Breis, Miguela Iniesta, Michel Dumontier, Mikel Egaña Aranguren, Stefan Schulz, Nathalie Aussenac-Gilles, Robert Stevens
Páginas	2696-2703
Volumen, Número	Vol. 40, No. 13
Tipo	Artículo científico
Revista	Expert Systems with Applications
Factor de Impacto (2013)	1.965
Editor	Elsevier Ltd.
Año	2012
ISSN	0957-4174
DOI	10.1016/j.eswa.2012.11.004
URL	http://www.sciencedirect.com/science/article/pii/S0957417412012146
Estado	Publicado

ABSTRACT

The increasing importance of ontologies has resulted in the development of a large number of ontologies in both coordinated and non-coordinated efforts. The number and complexity of such ontologies make hard to ontology and tool developers to select which ontologies to use and reuse. So far, there are no mechanism for making such decisions in an informed manner. Consequently, methods for evaluating ontology quality are required. OQuaRE is a method for ontology quality evaluation which adapts the SQuaRE standard for software product quality to ontologies. OQuaRE has been applied to identify the strengths and weaknesses of different ontologies but, so far, this framework has not been evaluated itself. Therefore, in this paper we present the evaluation of OQuaRE, performed by an international panel of experts in ontology engineering. The results include the positive and negative aspects of the current version of OQuaRE, the completeness and utility of the quality metrics included in OQuaRE and the comparison between the results of the manual

7.2 Evaluation of the OQuaRE framework for ontology quality

evaluations done by the experts and the ones obtained by a software implementation of OQuaRE.

7. PUBLICACIONES QUE COMPONEN ESTA TESIS

7.3 Evaluating the Good Ontology Design Guideline (GoodOD) with the Ontology Quality Requirements and Evaluation Method and Metrics (OQuaRE)

Título	Evaluating the Good Ontology Design Guideline (GoodOD) with the Ontology Quality Requirements and Evaluation Method and Metrics (OQuaRE)
Autores	Astrid Duque-Ramos, Martin Boeker, Ludger Jansen, Stefan Schulz, Miguela Iniesta, Jesualdo Tomaás Fernández-Breis
Páginas	e104463-1 a e104463-14
Volumen/ Número	Vol. 9, No. 8
Tipo	Artículo científico
Revista	PLOS ONE
Factor de Impacto (2014)	3.234
Editor	Public Library of Science
Año	2011
ISSN	1932-6203
DOI	10.1371/journal.pone.0104463
URL	http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0104463
Estado	Publicado

ABSTRACT

Objective: To (1) evaluate the GoodOD guideline for ontology development by applying the OQuaRE evaluation method and metrics to the ontology artefacts that were produced by students in a randomized controlled trial, and (2) informally compare the OQuaRE evaluation method with gold standard and competency questions based evaluation methods, respectively.

Background: In the last decades many methods for ontology construction and ontology evaluation have been proposed. However, none of them has become a standard and there is no empirical evidence of comparative evaluation of such methods. This paper brings together GoodOD and OQuaRE. GoodOD is a guideline for developing robust ontologies. It was previously evaluated in a randomized controlled trial employing metrics based on gold standard ontologies and competency questions as

7.3 Evaluating the Good Ontology Design Guideline (GoodOD) with the Ontology Quality Requirements and Evaluation Method and Metrics (OQuaRE)

outcome parameters. OQuaRE is a method for ontology quality evaluation which adapts the SQuaRE standard for software product quality to ontologies and has been successfully used for evaluating the quality of ontologies.

Methods: In this paper, we evaluate the effect of training in ontology construction based on the GoodOD guideline within the OQuaRE quality evaluation framework and compare the results with those obtained for the previous studies based on the same data.

Results: Our results show a significant effect of the GoodOD training over developed ontologies by topics: (a) a highly significant effect was detected in three topics from the analysis of the ontologies of untrained and trained students; (b) both positive and negative training effects with respect to the gold standard were found for five topics.

Conclusion: The GoodOD guideline had a significant effect over the quality of the ontologies developed. Our results show that GoodOD ontologies can be effectively evaluated using OQuaRE and that OQuaRE is able to provide additional useful information about the quality of the GoodOD ontologies.

*Now this is not the end. It is not even
the beginning of the end. But it is,
perhaps, the end of the beginning.*

Winston Churchill

CAPÍTULO

8

Conclusiones y trabajo futuro

8.1 Contribución científica

- En esta tesis se combina la Ingeniería Ontológica con la Ingeniería de Software para crear el *framework* de evaluación de calidad de ontologías (OQuaRE) con base en el estándar de evaluación de la calidad del software *SQuaRE*. OQuaRE adapta *SQuaRE* a través de la definición de las características y subcaracterísticas del modelo de calidad en términos ontológicos y de la inclusión de las característica Estructura, no definida en el estándar, y las subcaracterísticas de la característica Adecuación funcional que son específicas para ontologías.
- Las ontologías pueden ser consideradas artefactos de software, sus elementos son similares a los componentes de la programación orientada a objetos, es por ello que este método adapta un conjunto de métricas usadas en Programación Orientada a Objetos para la medición de la calidad de las ontologías, además recoge un conjunto de métricas de calidad de ontologías para asociarlas a las subcaracterísticas, incluyendo además las escalas predefinidas para convertir

8. CONCLUSIONES Y TRABAJO FUTURO

las métricas en valores de 1-5, donde 1 indica una calificación baja y 5 una alta.

- OQuaRE proporciona 49 criterios de calidad, reutiliza un conjunto de métricas de ontologías y adapta un conjunto de métricas de software a ontologías, creando asociaciones de métricas con subcaracterísticas y éstas a su vez asociadas a características, permitiendo a la comunidad de expertos en ontologías tener una idea más clara del significado de las métricas y su aporte a la calidad de las ontologías.

8.2 Limitaciones identificadas

- OQuaRE permite identificar el grado en que una ontología cumple con los requisitos, sin embargo, en la versión actual no se tienen aún definidos los perfiles o configuraciones que incluyen los requisitos de calidad, pero se han hecho algunas aproximaciones en las publicaciones presentadas como parte de esta tesis doctoral (Capítulo 7).
- En el estudio de versiones de la ontología EDAM, se detectó que la escala actual en algunas de las métricas no respresenta los cambios reales de la ontología de una versión a otra y por lo tanto, algunos cambios en los valores reales de la métrica no generaban un cambio en la escala, es por ello que se está estudiando la posibilidad de mejorar la escala, en los trabajos futuros se explican las opciones para ello.
- Algunas de las subcaracterísticas no se tienen asignadas aún métricas y por lo tanto, con la versión actual de OQuaRE, aunque se mide un gran número de subcaracterísticas, no es posible medir todas las subcaracterísticas del modelo, la inclusión de nuevas métricas o métricas existentes en la comunidad de ontología se incluye como trabajo futuro, una aproximación ya se ha hecho al incluir las métricas de *POO* al modelo.

8.3 Trabajos futuros

Los trabajos futuros están encaminados a la evaluación de las ontologías y mejora del método.

- En la línea de evaluación, se está trabajando en la utilización de OQuaRE para evaluar la calidad de diferentes versiones de una misma ontología con el fin de identificar los cambios de una versión respecto a su versión anterior en términos de calidad. En esta línea, ya se presentó un trabajo en el ICBO 2015 [Quesada-Martínez et al., 2015], en el que se identificaron los cambios de calidad en la ontología EDAM. Actualmente, se está trabajando en las versiones de las ontologías del *OBO Foundry* para identificar los cambios entre versiones y comparar estos resultados con trabajos realizados de versionado en los que se califican las ontologías en términos de madurez.
- En la línea de mejora del método, se está trabajando en el rediseño de las escalas de valores. Para ello se está estudiando la posibilidad de crear una escala basada en métodos estadísticos de partición de clases o creación de grupos clúster. El trabajo actual se está haciendo con división por vecindad, en la cual la escala viene dada por los valores más cercanos, es decir, el vecino más próximo. Adicionalmente, se está estudiando la posibilidad de crear una escala basada en aprendizaje, con el fin de encontrar un escalado dependiente del repositorio de ontologías. Inicialmente, se trabajará con las ontologías del *Bioportal* y se mirará el manejo de la escala para otros repositorios.
- Otros trabajos futuros van encaminados a la inclusión de nuevas métricas y de métricas semánticas en el *framework* OQuaRE. Una primera aproximación se ha hecho para las ontologías de negocios FIBO, en el cual se incluyeron las métricas de POO a la tabla de características y subcaracterísticas creando nuevas asociaciones métricas-subcaracterísticas.
- Se incluirán configuraciones de calidad en las que se combinen varias subcaracterísticas que cumplan con un requerimiento específico y como resultado, se tenga un valor que indique si la ontología cumple o no dicho requisito. En el artículo de GoodOD ya se hizo una primera aproximación a esta trazabilidad.

8. CONCLUSIONES Y TRABAJO FUTURO

- Se buscará cómo el contenido y los cambios en calidad se ajustan al dominio representado por la ontología.
- OQuaRE se ha usado para evaluar ontologías ya desarrolladas, pero servirá también para definir lo que requiere una ontología antes de ser creada ya que, de acuerdo a los requisitos de ésta, se puede indicar la forma en que se desarrolla.

8.4 Conclusiones

- OQuaRE es un *framework* de evaluación de ontologías que integra las medidas de varios criterios de la ontología y las agrupa en un modelo de calidad basado en el estándar de calidad del software *SQuaRE ISO 25000:2005*. OQuaRE evalúa la calidad de las ontologías de manera objetiva y reproducible, e incluye trazabilidad entre características, subcaracterísticas y métricas, partiendo desde un alto nivel, hasta llegar a un nivel más bajo de granularidad.
- Las asociaciones características-subcaracterísticas-métricas permiten un mejor entendimiento de las conexiones entre las características de una ontología y los criterios de calidad, además de una visión más clara de las métricas y su utilidad en la evaluación de las ontologías.
- OQuaRE sirve como base para evaluar la estructura de una ontología, desde su aspecto funcional, hasta su aspecto más técnico y puede ser de aplicación para ontologías de diferentes dominios.
- OQuaRE contribuye a la toma de decisiones informadas porque presenta las fortalezas y debilidades de una ontología en términos de características y subcaracterísticas de calidad.
- OQuaRE es un *framework* adaptable que permite incorporar nuevas métricas y asociaciones-subcaracterística-métrica y crear o redefinir las escalas para cada repositorio particular.
- En los experimentos se encontró que el nivel de granularidad de las características es muy alto, permitiendo solamente tener una idea global de la

calidad de la ontología, el de las métricas es muy bajo y por lo tanto muy específico y poco entendible para los usuarios. Es por ello que el análisis de la calidad de las ontologías se hace a nivel de subcaracterística, ya que contiene el nivel de granularidad adecuado para el análisis de los resultados de la evaluación de las ontologías y la facilidad para ser entendido.

8.5 Contribuciones

8.5.1 Publicaciones

- Astrid Duque-Ramos, Martin Boeker, Ludger Jansen, Stefan Schulz, Miguela Iniesta, Jesualdo Tomás Fernández-Breis, *Evaluating the Good Ontology Design guideline (GoodOD) with the Ontology Quality Requirements and Evaluation method and metrics (OQuaRE)*, PLOS One 9(8), e104463 (factor de impacto 2014:3,234)
- Astrid Duque Ramos, Jesualdo Tomás Fernández-Breis, Miguela Iniesta, Michel Dumontier, Mikel Egaña Aranguren, Stefan Schulz, Nathalie Aussenac-Gilles, Robert Stevens. *Evaluation of the OQuaRE framework for ontology quality*, Expert Systems with Applications 40 (2013) pp. 2696-2703 (factor de impacto 2013:1,965)
- Astrid Duque-Ramos, Jesualdo Tomás Fernández-Breis, Robert Stevens, Nathalie Aussenac-Gilles. *OQuaRE: a SQuaRE-based approach for evaluating the quality of ontologies*, Journal of Research and Practice in Information Technology, vol 43 No 2:159-173 (factor de impacto 2010:0,205)

8.5.2 Conferencias Internacionales y Workshops

- Manuel Quesada Martínez, Astrid Duque Ramos and Jesualdo Tomás Fernández Breis, *Analysis of the evolution of ontologies using OQuaRE: application to EDAM*, ICBO 2015

8. CONCLUSIONES Y TRABAJO FUTURO

- A. Duque Ramos, JesualdoTomas Fernandez Breis, Robert Stevens, Nathalie Aussenac Gilles. *The OQuaRE Framework for Ontology Evaluation*¹. Ontology Summit 2013: Ontology Evaluation Across the Ontology Lifecycle,
- A. Duque Ramos, U. López, J.T. Fernández Breis and R. Stevens. *Towards a SQuaRE based Quality Evaluation Framework for Ontologies*, Workshop on Ontology Quality, EKAW 2010

8.5.3 Premios

First IAOA Prize for the Best Hackathon-Clinic Project@ Ontology Summit 2013:“Evaluation of OOPS!, OQuaRE and OntoQA for FIBO Ontologies”. Este premio fue evaluado por un tribunal comprendido por Leo Orbst, Mathew West y Michael Gruninger.

¹ (http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_01_31)

Bibliografía

- [Alani et al., 2006] Alani, H., Brewster, C., and Shadbolt, N. (2006). Ranking ontologies with aktiverank. In *The 5th International Semantic Web Conference (ISWC)*, pages 1–15. Springer Berlin Heidelberg. 3
- [Antoniou and Harmelen, 2003] Antoniou, G. and Harmelen, F. V. (2003). Web ontology language: Owl. In *Handbook on Ontologies in Information Systems*, pages 67–92. Springer. 9
- [Ashburner et al., 2000] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene Ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29. 2
- [Baader and Nutt, 2003] Baader, F. and Nutt, W. (2003). Basic description logics. In Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors, *The Description Logic Handbook*, pages 43–95. Cambridge University Press, New York, NY, USA. 10, 11, 12
- [Bao et al., 2006] Bao, J., Caragea, D., and Honavar, V. (2006). On the semantics of linking and importing in modular ontologies. *The Semantic Web - ISWC 2006*, pages 72–86. 24
- [Bao et al., 2007] Bao, J., Slutzki, G., and Honavar, V. (2007). A semantic importing approach to knowledge reuse from multiple ontologies. In *AAAI*, pages 1304–1309. AAAI Press. 24

BIBLIOGRAFÍA

- [Bard et al., 2005] Bard, J., Seung, Y. R., and Michael, A. (2005). An ontology for cell types. *Genome Biology*, 6(2):R21. 53
- [Baumeister and Seipel, 2006] Baumeister, J. and Seipel, D. (2006). Verification and refactoring of ontologies with rules. In *EKAW'06: Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management*, pages 82–95. 24
- [Beisswanger et al., 2008] Beisswanger, E., Schulz, S., Stenzhorn, H., and Hahn, U. (2008). Biotop: An upper domain ontology for the life sciences: A description of its current structure, contents and interfaces to obo ontologies. *Appl. Ontol.*, 3(4):205–212. 2
- [Benjamins et al., 2004] Benjamins, R., Contreras, J., Corcho, O., and Gómez-Pérez, A. (2004). Six challenges for the semantic web. *AIS SIGSEMIS Bulletin*, 1(1):24–25. 25
- [Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):28–37. 1
- [Bloehdorn et al., 2005] Bloehdorn, S., Cimiano, P., Hotho, A., and Staab, S. (2005). An ontology-based framework for text mining. *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, 20(1):87–112. 24
- [Boeker et al., 2013a] Boeker, M., Grewe, N., Röhl, J., Schober, D., Schulz, S., Seddig-Raufie, D., and Jansen, L. (2013a). Measuring the effect of a guideline-based training on ontology design with a competency questions based evaluation approach. In *Informatik 2013, 43. Jahrestagung der Gesellschaft für Informatik e.V. (GI), Informatik angepasst an Mensch*, Lecture Notes in Informatics, pages 1783–1795. Matthias Horbach, GI. 71
- [Boeker et al., 2013b] Boeker, M., Jansen, L., Grewe, N., Röhl, J., Schober, D., Seddig-Raufie, D., and Schulz, S. (2013b). Effects of guideline-based training on the quality of formal ontologies: A randomized controlled trial. *PLOS One*, 8(5):e61425. 59, 71

- [Boeker et al., 2012] Boeker, M., Schober, D., Raufie, D., Grewe, N., Röhl, J., Jansen, L., and Schulz, S. (2012). Teaching good biomedical ontology design. In *ICBO*. 3, 31, 70
- [Borst, 1997] Borst, W. N. (1997). *Construction of engineering ontologies for knowledge sharing and reuse*. PhD thesis, Universiteit Twente, Enschede, The Netherlands. CTIT Ph.D-thesis series No.97-14. 7
- [Brank et al., 2005] Brank, J., Grobelnik, M., and Mladenic, D. (2005). A survey of ontology evaluation techniques. In *In Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005)*. 2, 27
- [Brewster et al., 2004] Brewster, C., Alani, H., and Dasmahapatra, A. (2004). Data driven ontology evaluation. In *Int. Conf. on Language Resources and Evaluation*.
- [Buggenhout and Ceusters, 2005] Buggenhout, C. V. and Ceusters, W. (2005). A novel view on information content of concepts in a large ontology and a view on the structure and the quality of the ontology. *I. J. Medical Informatics*, 74(2-4):125–132.
- [Chidamber and Kemerer, 1994] Chidamber, S. and Kemerer, C. (1994). A metric suite for object oriented design. *IEEE Transactions on Software Engineering*, pages 467–493. 15, 22, 42
- [Chikofsky and Cross II, 1990] Chikofsky, E. J. and Cross II, J. H. (1990). Reverse engineering and design recovery: A taxonomy. *IEEE Softw.*, 7(1):13–17. 3
- [Cimiano and Völker, 2005] Cimiano, P. and Völker, J. (2005). Text2onto. In *Natural Language Processing and Information Systems*, pages 227–238. Springer.
- [Cimino and Zhu, 2006] Cimino, J. and Zhu, X. (2006). The practical impact of ontologies on biomedical informatics. *Methods of Information in Medicine*, 45 Suppl 1:124–135.
- [Corcho et al., 2004] Corcho, O., Gómez-Pérez, A., González-Cabero, R., and Suárez-Figueroa, M. C. (2004). ODEVAL: A Tool for Evaluating RDF(S), DAML+OIL and OWL Concept Taxonomies. In *Artificial Intelligence Applications and Innovations*, pages 369–382. Springer US. 3, 17, 24

BIBLIOGRAFÍA

- [Crosby, 1979] Crosby, P. (1979). *Quality is free: the art of making quality certain*. New American Library, New York. 13
- [D' Aquin and Gangemi, 2011] D' Aquin, M. and Gangemi, A. (2011). Is there beauty in ontologies? *Applied Ontology*, 6(3):165–175.
- [David and Euzenat, 2008] David, J. and Euzenat, J. (2008). Comparison between ontology distances (preliminary results). In Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., and Thirunarayan, K., editors, *The Semantic Web - ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 245–260. Springer Berlin Heidelberg. 2, 18
- [Davies et al., 2003] Davies, J., Fensel, D., and Harmelen, F. V. (2003). *Towards the semantic web*. Wiley Online Library.
- [Dellschaft and Staab, 2006] Dellschaft, K. and Staab, S. (2006). On how to perform a gold standard based evaluation of ontology learning. In Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., and Aroyo, L., editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 228–241. Springer Berlin Heidelberg. 2, 18
- [Ding and Peng, 2004] Ding, Z. and Peng, Y. (2004). A probabilistic extension to ontology language OWL. In *Proceedings of the 37th Hawaii International Conference on System Sciences*. 24
- [Dragoni et al., 2010] Dragoni, M., da Costa Pereira, C., and Tettamanzi, A. (2010). An ontological representation of documents and queries for information retrieval systems. In Melucci, M., Mizzaro, S., and Pasi, G., editors, *IIR*, volume 560 of *CEUR Workshop Proceedings*, pages 83–87. CEUR-WS.org. 24
- [Duque-Ramos et al., 2013] Duque-Ramos, A., Fernández-Breis, J., Iniesta, M., Dumontier, M., Aranguren, M. E., Schulz, S., Aussenac-Gilles, N., and Stevens, R. (2013). Evaluation of the oquare framework for ontology quality. *Expert Systems with Applications*, 40(7):2696–2703.

- [Duque-Ramos et al., 2011] Duque-Ramos, A., Fernández-Breis, J. T., Stevens, R., and Aussenac-Gilles, N. (2011). Oquare: a square-based approach for evaluating the quality of ontologies. *Journal of Research and Practice in Information Technology*, 43(2):159–73. 33
- [Duque-Ramos et al., 2010] Duque-Ramos, A., Lopez, U., Fernández-Breis, J. T., and Stevens, R. (2010). Towards an square-based quality evaluation framework for ontologies. In *Workshop on Ontology Quality, EKAW*.
- [Euzenat et al., 2009] Euzenat, J., Allocca, C., David, J., dÁquin, M., Le Duc, C., and Sváb-Zamazal, O. (2009). NeOn:Lifecycle Support for Networked Ontologies - D3.3.4: Ontology distances for contextualization.
- [Everitt and Hothorn, 2011] Everitt, B. S. and Hothorn, T. (2011). *An Introduction to Applied Multivariate Analysis with R*. Springer-Verlag, New York.
- [Evermann and Fang, 2010] Evermann, J. and Fang, J. (2010). Evaluating ontologies: Towards a cognitive measure of quality. *Information Systems*, 35(4):391 – 403. Vocabularies, Ontologies and Rules for Enterprise and Business Process Modeling and Management. 13
- [Fensel and Musen, 2001] Fensel, D. and Musen, M. A. (2001). The semantic web: A brain for humankind. *IEEE Intelligent Systems*, 16(2):24–25. 1
- [Fernández-Breis, 2003] Fernández-Breis, J. T. (2003). *Un Entorno de Integración de Ontologías par el Desarrollo de Sistemas de Gestión de Conocimiento*. PhD thesis, Universidad de Murcia. 7, 8
- [Fernández-Breis et al., 2009a] Fernández-Breis, J. T., Aranguren, M. E. n., and Stevens, R. (2009a). A quality evaluation framework for bio-ontologies. *ICBO*, page 127. 54
- [Fernández-Breis et al., 2009b] Fernández-Breis, J. T., Egana-Aranguren, M., and Stevens, R. (2009b). A quality evaluation framework for bio-ontologies. In *ICBO 2009: Proceedings of the 2009 International Conference on Biomedical Ontology*, pages 136–139, University at Buffalo, NY. Nature Precedings. 33, 54, 68

BIBLIOGRAFÍA

- [Flouris et al., 2008] Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., and Antoniou, G. (2008). Ontology change: classification and survey. *Knowledge Eng. Review*, 23(2):117–152. 25
- [Fox et al., 1997] Fox, M., Barbuceanu, M., Gruninger, M., and Lin, J. (1997). An organization ontology for enterprise modelling. In *Modeling, In: International Conference on Enterprise Integration Modelling Technology 97*. Springer. 19
- [Fox, 1992] Fox, M. S. (1992). The tove project towards a common-sense model of the enterprise. In *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 25–34. Springer.
- [Gajderowicz and Sadeghian, 2009] Gajderowicz, B. and Sadeghian, A. (2009). Ontology granulation through inductive decision trees. In Bobillo, F., da Costa, P. C. G., d’Ámato, C., Fanizzi, N., Laskey, K. B., Laskey, K. J., Lukasiewicz, T., Martin, T., Nickles, M., Pool, M., and Smrz, P., editors, *URSW*, volume 527 of *CEUR Workshop Proceedings*, pages 39–50. CEUR-WS.org. 24
- [Gangemi et al., 2006] Gangemi, A., Catenacci, C., Ciaramita, M., and Lehmann, J. (2006). Modelling ontology evaluation and validation. In *ESWC*, pages 140–154. 3, 18, 24
- [Gómez et al., 2000] Gómez, A., Moreno, A. M., Pazos, J., and Sierra-Alonso, A. (2000). Knowledge maps: An essential technique for conceptualisation. *Data Knowl. Eng.*, 33(2):169–190. 25
- [Gómez-Pérez, 2004] Gómez-Pérez, A. (2004). Ontology evaluation. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 251–273. Springer Berlin Heidelberg.
- [Gómez-Pérez and Rojas-Amaya, 1999] Gómez-Pérez, A. and Rojas-Amaya, M. (1999). Ontological reengineering for reuse. In Fensel, D. and Studer, R., editors, *Knowledge Acquisition, Modeling and Management*, volume 1621 of *Lecture Notes in Computer Science*, pages 139–156. Springer Berlin Heidelberg. 3

- [Grau et al., 2008] Grau, B. C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., and Sattler, U. (2008). Owl 2: The next step for owl. *Web Semant.*, 6(4):309–322. 10
- [Grewe et al., 2012] Grewe, N., Schober, D., and Boeker, M. (2012). Comparing closely related, semantically rich ontologies: The GoodOD Similarity Evaluator. In *4th Workshop Ontologies in Biomedicine and Life Sciences*. 70
- [Grimm and Wissmann, 2011] Grimm, S. and Wissmann, J. (2011). Elimination of redundancy in ontologies. In Antoniou, G., Grobelnik, M., Simperl, E. P. B., Parsia, B., Plexousakis, D., Leenheer, P. D., and Pan, J., editors, *Proceedings of the 8th Extended Semantic Web Conference (ESWC 2011)*, volume 6643 of *Lecture Notes in Computer Science*, pages 260–274. Springer-Verlag. 24
- [Gruber, 1991] Gruber, T. R. (1991). The role of common ontology in achieving sharable, reusable knowledge bases. In Allen, J. F., Fikes, R., and Sandewall, E., editors, *KR*, pages 601–602. Morgan Kaufmann. 24, 25
- [Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220. 1, 7, 8, 25
- [Gruninger and Fox, 1995] Gruninger, M. and Fox, M. (1995). Methodology for the design and evaluation of ontologies. In *IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing, April 13, 1995*. 19
- [Gruninger and Fox, 1994] Gruninger, M. and Fox, M. S. (1994). The design and evaluation of ontologies for enterprise engineering. 3
- [Guarino and Welty, 2004] Guarino, N. and Welty, C. (2004). An overview of ontoclean. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, pages 151–159. Springer, New York. 3, 17, 18, 24
- [Hameed et al., 2004] Hameed, A., Preece, A., and Sleeman, D. (2004). Ontology reconciliation. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, volume 10 of *International Handbooks on Information System*, chapter 12, pages 231–250. Springer. 24

BIBLIOGRAFÍA

- [Hepp, 2008] Hepp, M. (2008). GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In *Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns*, EKAW '08, pages 329–346, Berlin, Heidelberg. Springer-Verlag. 2
- [Hofweber, 2009] Hofweber, T. (2009). Logic and ontology. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Stanford University, spring 2009 edition. 7
- [Horrocks, 1998] Horrocks, I. (1998). The FaCT system. In *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX '98)*, volume 1397 in *Lecture Notes in Artificial Intelligence*, pages 307–312. Springer. 13
- [Horrocks et al., 2002] Horrocks, I. et al. (2002). Daml+oil: a description logic for the semantic web. *IEEE Data Eng. Bull.*, 25(1):4–9.
- [Horrocks et al., 2003] Horrocks, I., Patel-Schneider, P. F., and Harmelen, F. V. (2003). From shiq and rdf to owl: The making of a web ontology language. *Journal of Web Semantics*, 1:2003. 9, 10
- [Horrocks et al., 2007] Horrocks, I., Patel-schneider, P. F., McGuinness, D. L., and Welty, C. A. (2007). OWL: a Description Logic Based Ontology Language for the Semantic Web. In Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors, *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2nd edition edition. 12
- [ISO, 2005] ISO (2005). Iso 2005, software engineering - software product quality requirements and evaluation (square) - guide to square (iso/iec 25000), geneva, switzerland: International organization for standardization. 19, 49
- [ISO/IEC, 2001] ISO/IEC (2001). International standard iso/iec 9126, information technology product quality part1: Quality model. Technical report, International Organization for Standardization. 19

- [Jobson, 1992] Jobson, J. (1992). *Volume II: Categorical and Multivariate Methods*, chapter Applied Multivariate Data Analysis. Springer Texts in Statistics. Springer.
- [Kalfoglou and Schorlemmer, 2003] Kalfoglou, Y. and Schorlemmer, M. (2003). Ontology mapping: The state of the art. *Knowl. Eng. Rev.*, 18(1):1–31. 3, 4
- [Kara et al., 2012] Kara, S., Alan, A., Sabuncu, O., Akpınar, S., Cicekli, N. K., and Alpaslan, F. N. (2012). An ontology-based retrieval system using semantic indexing. *Inf. Syst.*, 37(4):294–305. 24
- [Kepler et al., 2006] Kepler, F., Paz-Trillo, C., Riani, J., Ribeiro, M. M., Valdivia-Delgado, K., Barros, L. N., and Wassermann, R. (2006). Classifying ontologies. In *Proceedings of the Second Workshop on Ontologies and their Applications (WONTO 2006)*. 24
- [Klein, 2004] Klein, M. (2004). *Change Management for Distributed Ontologies*. PhD thesis, Vrije Universiteit Amsterdam. 25
- [Klein et al., 2002] Klein, M., Fensel, D., Kiryakov, A., and Ognyanov, D. (2002). Ontology versioning and change detection on the web. In *Proc. of the 13th European Conf. on Knowledge Engineering and Knowledge Management (EKAW02) (Sigüenza, Spain)*, pages 197–212. 25
- [Koide et al., 2005] Koide, S., Aasman, J., and Haflich, S. (2005). Owl vs. object orientated programming. In *International Semantic Web Conference, Workshop1: Semantic Web Enabled Software Engineering. 2005*. 22
- [Legaz-García, 2015] Legaz-García, M. C. (2015). *Integración de Información Biomédica Basada en Tecnologías Semánticas Avanzadas*. PhD thesis, Universidad de Murcia. 9, 13
- [Li and Henry, 1993] Li, W. and Henry, S. (1993). Object-oriented metrics that predict maintainability. *J. Sys. Software*, 23:11–122. 42
- [Lincke and Lowe, 2007] Lincke, R. and Lowe, W. (2007). Compendium of software quality standards and metrics – version 1.0. <http://www.arisa.se/compendium/>. 45

BIBLIOGRAFÍA

- [López et al., 1999] López, M. F., Gómez-Pérez, A., Sierra, J. P., and Sierra, A. P. (1999). Building a chemical ontology using methontology and the ontology design environment. *IEEE Intelligent Systems*, 14(1):37–46. 3
- [López et al., 1997] López, M. F., Pérez, A. G., and Juristo, N. (1997). MET-HONTOLOGY: from Ontological Art towards Ontological Engineering. In *Proceedings of the AAAI97 Spring Symposium*, pages 33–40, Stanford, USA.
- [Ma et al., 2009] Ma, Y., Ma, X., Liu, S., and Jin, B. (2009). A proposal for stable semantic metrics based on evolving ontologies. *Artificial Intelligence, International Joint Conference on*, 0:136–139. 15, 24
- [Maedche and Staab, 2002] Maedche, A. and Staab, S. (2002). Measuring similarity between ontologies. In *EKAW '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 251–263, London, UK. Springer-Verlag. 2, 18, 24
- [Malone et al., 2010] Malone, J., Holloway, E., Adamusiak, T., Kapushesky, M., Zheng, J., Kolesnikov, N., Zhukova, A., Brazma, A., and Parkinson, H. (2010). Modeling sample variables with an experimental factor ontology. *Bioinformatics*, 26(8):1112–1118. 17
- [Malone and Stevens, 2013] Malone, J. and Stevens, R. (2013). Measuring the level of activity in community built bio-ontologies. *Journal of Biomedical Informatics*, 46(1):5 – 14. 17, 63
- [Manly, 1986] Manly, B. F. (1986). *Multivariate Statistical Methods: A Primer*. Chapman & Hall, Ltd., London, UK, UK, second edition edition.
- [Miller, 1995] Miller, G. A. (1995). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41. 14, 17
- [Moher et al., 2010] Moher, D., Hopewell, S., Schulz, K., Montori, V., Goetzsche, P., Devereaux, P., Elbourne, D., Egger, M., Altman, D., and Consolidated Standards of Reporting Trials Group (2010). CONSORT 2010 Explanation and Elaboration: Updated guidelines for reporting parallel group randomised trials. *Journal of Clinical Epidemiology*, 63(8):e1–37.

- [Neches et al., 1991] Neches, R., Fikes, R. E., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W. R. (1991). Enabling technology for knowledge sharing. *AI magazine*, 12(3):36. 7
- [Netzer et al., 2009] Netzer, Y. D., Gabay, D., Adler, M., Goldberg, Y., and Elhadad, M. (2009). Ontology evaluation through text classification. In 0002, L. C., Liu, C., Zhang, X., Wang, S., Strasunskas, D., Tomassen, S. L., Rao, J., Li, W.-S., Candan, K. S., Chiu, D. K. W., Zhuang, Y., Ellis, C. A., and Kim, K.-H., editors, *APWeb/WAIM Workshops*, volume 5731 of *Lecture Notes in Computer Science*, pages 210–221. Springer. 24
- [Neuhaus et al., 2013] Neuhaus, F., Vizedom, A., Baclawski, K., Bennett, M., Dean, M., Denny, M., Grüniger, M., Hashemi, A., Longstreth, T., Obrst, L., Ray, S., Sriram, R. D., Schneider, T., Vegetti, M., West, M., and Yim, P. (2013). Towards ontology evaluation across the life cycle the ontology summit 2013. *Applied Ontology*, 8(3):179–194. 19
- [Object Management Group, 2009] Object Management Group (2009). Ontology definition metamodel (omg) version 1.0. Technical Report formal/2009-05-01, Object Management Group.
- [Obrst et al., 2007] Obrst, L., Ceusters, W., Mani, I., Ray, S., and Smith, B. (2007). The evaluation of ontologies. In *Semantic Web*, pages 139–158. Springer US. 19
- [Oh et al., 2011] Oh, S., Yeom, H. Y., and Ahn, J. (2011). Cohesion and coupling metrics for ontology modules. *Information Technology and Management*, 12(2):81–96. 25
- [Ohta et al., 2011] Ohta, M., Kozaki, K., and Mizoguchi, R. (2011). A quality assurance framework for ontology construction and refinement. In Mugellini, E., Szczepaniak, P., Pettenati, M., and Sokhn, M., editors, *Advances in Intelligent Web Mastering*, volume 86 of *Advances in Intelligent and Soft Computing*, pages 207–216. Springer Berlin Heidelberg. 13
- [Opdahl and Henderson-Sellers, 2001] Opdahl, A. L. and Henderson-Sellers, B. (2001). Grounding the oml metamodel in ontology. *Journal of Systems and Software*, 57(2):119–143. 22

BIBLIOGRAFÍA

- [Orme et al., 2007] Orme, A. M., Yao, H., and Etkorn, L. H. (2007). Indicating ontology data quality, stability, and completeness throughout ontology evolution. *Journal of Software Maintenance*, 19(1):49–75. 25
- [Park et al., 2011] Park, J., Oh, S., and Ahn, J. (2011). Ontology selection ranking model for knowledge reuse. *Expert Syst. Appl.*, 38(5):5133–5144. 24
- [Patil et al., 1997] Patil, R., Knight, K., and Russ, T. (1997). Towards distributed use of large-scale ontologies. In *In AAAI Spring Symposium on Ontological Engineering*. 1
- [Porzel and Malaka, 2004] Porzel, R. and Malaka, R. (2004). A task-based approach for ontology evaluation. In *Proc. of ECAI 2004 Workshop on Ontology Learning and Population*, Valencia, Spain.
- [Poveda-Villalón et al., 2014] Poveda-Villalón, M., Gómez-Pérez, A., and Suárez-Figueroa, M. C. (2014). Oops! (ontology pitfall scanner!): An on-line tool for ontology evaluation. *Int. J. Semantic Web Inf. Syst.*, 10(2):7–34.
- [Poveda-Villalón et al., 2010] Poveda-Villalón, M., Suárez-Figueroa, M. C., and Gómez-Pérez, A. (2010). A double classification of common pitfalls in ontologies. *Ontology Quality Workshop at EKAW*. 63
- [Poveda-Villalón et al., 2012] Poveda-Villalón, M., Suárez-Figueroa, M. C., and Gómez-Pérez, A. (2012). Validating ontologies with oops! In *Proceedings of the 18th International Conference on Knowledge Engineering and Knowledge Management, EKAW'12*, pages 267–281, Berlin, Heidelberg. Springer-Verlag. 63
- [Quesada-Martínez et al., 2015] Quesada-Martínez, M., Duque-Ramos, A., and Fernández-Breis, J. T. (2015). Analysis of the evolution of ontologies using OQuaRE : Application to EDAM. In *Proceedings of the International Conference on Biomedical Ontology 2015*, pages 62–66, Lisbon. 60, 125
- [Quine, 1961] Quine, W. V. O. (1961). *From a logical point of view: 9 logico-philosophical essays*. Harvard University Press. 7

- [Rijgersberg et al., 2011] Rijgersberg, H., Wigham, M., and Top, J. (2011). How semantics can improve engineering processes: A case of units of measure and quantities. *Advanced Engineering Informatics*, 25(11):276–287.
- [Rogers, 2006] Rogers, J. E. (2006). Quality assurance of medical ontologies. *Methods Inf Med*, 45(3):267–274. 3, 24
- [Saaty, 1980] Saaty, T. (1980). *The Analytic Hierarchy Process, Planning, Priority Setting, Resource Allocation*. McGraw-Hill, New york. 16
- [Saaty, 2000] Saaty, T. (2000). *Fundamentals of the Analytic Hierarchy Process*. RWS Publications, 4922 Ellsworth Avenue, Pittsburgh, PA 15413. 16
- [Sabou et al., 2006] Sabou, M., Lopez, V., Motta, E., and Uren, V. (2006). Ontology selection: Ontology evaluation on the real semantic web. 2
- [Sachs, 1982] Sachs, L. (1982). *Applied Statistics. A Handbook of Techniques*. Springer-Verlag, second edition edition.
- [Sánchez-Vera et al., 2012] Sánchez-Vera, M. M., Fernández-Breis, J. T., Castellanos-Nieves, D., Frutos-Morales, F., and Prendes-Espinosa, M. P. (2012). Semantic web technologies for generating feedback in online assessment environments. *Knowledge-Based Systems*, 33:152–165. 2
- [Shearer et al., 2008] Shearer, R., Motik, B., and Horrocks, I. (2008). HermiT: A Highly-Efficient OWL Reasoner. In *OWLED*, volume 432. 13
- [Simperl, 2009] Simperl, E. P. B. (2009). Reusing ontologies on the semantic web: A feasibility study. *Data Knowl. Eng.*, 68(10):905–925. 25
- [Sirin et al., 2007] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. (2007). Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53. 13
- [Sleeman and Reul, 2006] Sleeman, D. and Reul, Q. (2006). Cleanonto: Evaluating taxonomic relationships in ontologies. In *Workshop on Evaluating Ontologies*. 3, 17

BIBLIOGRAFÍA

- [Smith et al., 2007] Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L. J., Eilbeck, K., Ireland, A., Mungall, C. J., et al. (2007). The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25(11):1251–1255.
- [Stevens and Lord, 2009] Stevens, R. and Lord, P. (2009). Application of ontologies in bioinformatics. In *Handbook on Ontologies*, pages 735–756. Springer. 35, 36
- [Stevens et al., 2008] Stevens, R., Wroe, C., Gobel, C., , and Lord, P. (2008). Applications of ontologies in bioinformatics. In Staab, S. and Studer, R., editors, *Handbook on Ontologies in Information Systems*, pages 635–658. Springer.
- [Strasunskas and Tomassen, 2008] Strasunskas, D. and Tomassen, S. L. (2008). Empirical insights on a value of ontology quality in ontology-driven web search. In Meersman, R. and Tari, Z., editors, *OTM Conferences (2)*, volume 5332 of *Lecture Notes in Computer Science*, pages 1319–1337. Springer. 24
- [Studer et al., 1998] Studer, R., Benjamins, V. R., and Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1):161–197. 8
- [Stvilia, 2007] Stvilia, B. (2007). A model for ontology quality evaluation. *First Monday*, 12(12). 3
- [Sure and Domingue, 2006] Sure, Y. and Domingue, J., editors (2006). *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings*, volume 4011 of *Lecture Notes in Computer Science*. Springer.
- [Tartir and Arpinar, 2007] Tartir, S. and Arpinar, I. B. (2007). Ontology evaluation and ranking using ontoqa. In *ICSC '07: Proceedings of the International Conference on Semantic Computing*, pages 185–192, Washington, DC, USA. IEEE Computer Society. 14, 42

- [Tartir et al., 2005] Tartir, S., Arpinar, I. B., Moore, M., Sheth, A. P., and Alemanmeza, B. (2005). Ontoqa: Metric-based ontology quality analysis. In *IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*. 3, 15
- [Tello and Gómez-Pérez, 2004] Tello, A. L. and Gómez-Pérez, A. (2004). Onto-metric: A method to choose the appropriate ontology. *J. Database Manag.*, 15(2):1–18. 3, 16, 24
- [Tian and Huang, 2012] Tian, Y. and Huang, M. (2012). Enhance discovery and retrieval of geospatial data using soa and semantic web technologies. *Expert Syst. Appl.*, 39(16):12522–12535. 2
- [Uschold et al., 1996] Uschold, M., Gruninger, M., Uschold, M., and Gruninger, M. (1996). Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11:93–136. 3
- [van Solingen and Egon, 1999] van Solingen, R. and Egon, B. (1999). *The Goal-Question/Metric Method*. McGraw-Hill Education. 72
- [Völker et al., 2008] Völker, J., Vrandečić, D., Sure, Y., and Hotho, A. (2008). Aeon - an approach to the automatic evaluation of ontologies. *Appl. Ontol.*, 3(1-2):41–62. 24
- [Vrandečić, 2010] Vrandečić, D. (2010). *Ontology Evaluation*. PhD thesis, Institute of Applied Informatics and Formal Description Methods AIFB. 3, 17, 27
- [Vrandečić and Sure, 2007] Vrandečić, D. and Sure, Y. (2007). How to design better ontology metrics. In *Proceedings of the 4th European conference on The Semantic Web: Research and Applications*, ESWC '07, pages 311–325, Berlin, Heidelberg. Springer-Verlag. 24
- [W3C, 2004a] W3C (2004a). Inference. 11
- [W3C, 2004b] W3C (2004b). OWL Web Ontology Language. 9, 10
- [W3C, 2009] W3C (2009). OWL 2 Web Ontology Language. 10

BIBLIOGRAFÍA

- [Wang et al., 2007] Wang, Y., Peter, H., and Jie, B. (2007). A survey of formalisms for modular ontologies. In *Proceedings of the International Joint Conference on Artificial Intelligence 2007 (IJCAI'07) Workshop SWeCKa*. 24, 25
- [Ward, 1963] Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244. 70
- [Whetzel et al., 2006] Whetzel, P. L., Parkinson, H., Causton, H. C., Fan, L., Fostel, J., Fragoso, G., Game, L., Heiskanen, M., Morrison, N., Rocca-Serra, P., Sansone, S.-A., Taylor, C., White, J., and Stoeckert, C. J. (2006). The MGED ontology: A Resource for semantics-based description of microarray experiments. *Bioinformatics*, 22(7):866–873. 24
- [Yao et al., 2005] Yao, H., Orme, A. M., and Etzkorn, L. (2005). Cohesion metrics for ontology design and application. *Journal of Computer science*, 1(1):107–113. 15, 42
- [Zhang et al., 2010] Zhang, H., Li, Y.-F., and Tan, H. B. K. (2010). Measuring design complexity of semantic web ontologies. *J. Syst. Softw.*, 83(5):803–814. 22