

Herramientas Útiles para la Evaluación de la Calidad de Ontologías Durante su Ciclo de Vida

Damián Forzani¹, María Laura Taverna¹, Mariela Rico¹

¹Centro de Investigación y Desarrollo de Ingeniería en Sistemas de Información (CIDISI) – Universidad Tecnológica Nacional – Facultad Regional Santa Fe
Lavaise 610 – S3004EWB – Santa Fe – SF – Argentina

damianforzani@gmail.com, ltaverna@fiq.unl.edu.ar,
mrico@frsf.utn.edu.ar

Abstract. *The Software Engineering (SE) community is using ontologies to solve different problems and their quality must be evaluated like any other SE artifact. Ontology evaluation methods generally quantify the quality of the final product. In this paper, based on the measures proposed in OntoQualitas, new guidelines are inferred to evaluate the quality of an ontology in different phases of its life cycle. Evaluating possible errors in these phases allows a better quality, achieved through feedback.*

Resumen. *La comunidad de Ingeniería de Software (IS) está utilizando ontologías para resolver diferentes problemas y su calidad debe evaluarse como cualquier otro artefacto de IS. Los métodos de evaluación de ontologías generalmente cuantifican la calidad del producto final. En este trabajo, a partir de las medidas propuestas en OntoQualitas, se infieren nuevas pautas para evaluar la calidad de una ontología en diferentes fases de su ciclo de vida. Evaluar posibles errores en estas fases permite una mejor calidad, lograda por medio de la retroalimentación.*

1. Introducción

La comunidad de la Ingeniería de Software ha reconocido la utilidad de las tecnologías de la Web Semántica, especialmente las ontologías, para resolver diferentes problemas [Bhatia et al. 2016], [Gašević et al. 2009], [Zhao et al. 2009]. Las ontologías se han utilizado para representar y comunicar el conocimiento aceptado dentro de algún subdominio, dando soporte a las distintas fases del ciclo de vida del software; y como artefactos de software, donde se distinguen propuestas en tiempo de desarrollo y en tiempo de ejecución, según el momento en el que se utilizan las ontologías [Ruiz and Hilera 2006]. Más aún, diferentes sinergias entre Ingeniería de Software y las ontologías han atraído la atención de cuerpos de estandarización. Uno de los más importantes resultados es el Metamodelo de Definición de Ontología (Ontology Definition Metamodel, ODM), una especificación del Object Management Group (OMG). La especificación permite la integración de lenguajes de ontologías en el proceso de desarrollo de software basado en los principios de ingeniería dirigida por modelos [Object Management Group 2014].

Uno de los objetivos de la Ingeniería de Software es generar un producto de alta calidad. En este sentido, se distinguen dos aspectos importantes: la calidad del producto

y la calidad del proceso de desarrollo. Relacionada con la calidad del producto, se han hecho esfuerzos para lograr estándares internacionales, entre ellos: ISO/IEC 9126 (Software Product Quality), ISO/IEC 14598 (Software Product Evaluation) e ISO/IEC 25000 (System and Software Quality Requirements and Evaluation, SQuaRE), que sustituye a las dos normas mencionadas anteriormente [ISO/IEC 2014]. Dado que “producto” refiere tanto a producto final como intermedio y también producto destinado a usuarios como desarrolladores y mantenedores, la calidad de las ontologías también debe ser evaluada.

Diferentes trabajos se han propuesto en función del tipo de ontologías que se evalúa y con qué propósito [Brank et al. 2005, Burton-Jones et al. 2005, Colomb 2002, Stvilia 2007, Supekar et al. 2004, Tartir et al. 2005]. Sin embargo, ninguno se ha convertido en un estándar. Un esfuerzo por adaptar la norma SQuaRE a la evaluación de ontologías es OQuaRE [Duque-Ramos et al. 2011] que incluye el modelo de calidad, compuesto por un conjunto de características de calidad y sub-características asociadas. Las sub-características de calidad tienen medidas relacionadas; algunas adaptadas de las medidas de programación orientada a objetos y otras de las desarrolladas por la comunidad de Ingeniería Ontológica, especialmente las propiedades estructurales.

OQuaRE puede ser visto como un paso hacia la estandarización del proceso de evaluación de ontologías. Sin embargo, un aspecto de importancia que no considera es el objetivo de la misma. La calidad no es una propiedad de algo, sino un juicio, por lo que debe estar en relación con algún propósito [Colomb 2002]. Tratar de evaluar la calidad de una ontología sin tener en cuenta sus usos previstos responde a un proceso incompleto. El conjunto de características, sub-características, medidas y los pesos respectivos deben estar en relación con el propósito de la ontología [Rico et al. 2014].

En 2013, usuarios y desarrolladores de sistemas, software y ontologías participaron de la Cumbre de Ontología, donde se crea una guía sobre cómo evaluar su calidad [Neuhaus et al. 2013]. Se argumenta cómo las mejores prácticas establecidas en Ingeniería de Sistemas y en Ingeniería de Software pueden ser utilizadas en el desarrollo de ontologías. Para un resultado más confiable en su desarrollo y uso, la comunidad participante de la Cumbre recomienda la evaluación en todas las fases de su ciclo de vida y llevarla a cabo en función de los requisitos cuidadosamente identificados, que dependen del uso previsto y de su entorno operativo.

Un framework propuesto que considera el propósito específico de las ontologías es OntoQualitas, que permite evaluar la calidad de ontologías implementadas en OWL 2 [Motik et al. 2012], cuyo objetivo es definir la semántica de la información intercambiada en un entorno de procesos de negocio colaborativo. Para ello se identifica un conjunto de requisitos que la ontología debe cumplir y un conjunto de preguntas que reflejan aspectos específicos relevantes para su evaluación. Para cada pregunta, se definen nuevas medidas y se adoptan otras ya conocidas, sus rangos de valores posibles y sus valores óptimos [Rico et al. 2014]. Tibaldo et al. [2015] calcularon manualmente las medidas de OntoQualitas sobre un conjunto de ontologías para estudiar su comportamiento respecto de la calidad. Los resultados parciales obtenidos en este trabajo mostraron que el proceso de evaluación conlleva una carga de trabajo excesiva y una alta dependencia del juicio humano, siendo necesario proporcionar una herramienta automatizada que brinde apoyo al evaluador.

En el presente trabajo se muestra cómo información de las medidas de OntoQualitas puede ser de utilidad para evaluar la calidad de las ontologías en diferentes fases de su ciclo de vida. También se discuten dificultades encontradas a la hora de automatizar el cálculo de algunas medidas de calidad.

El trabajo está organizado de la siguiente manera: en la Sección 2 se sitúa a OntoQualitas en relación a la Cumbre de Ontología. La Sección 3 presenta las medidas que pueden ser utilizadas para evaluar la calidad, aquellas pendientes de automatización y advertencias. La Sección 4 discute los beneficios que aportan las advertencias y los avances realizados sobre la automatización del cálculo de las medidas, mientras que la Sección 5 presenta brevemente la aplicación en desarrollo. Finalmente, la Sección 6 está dedicada a las conclusiones y trabajos futuros.

2. OntoQualitas y la Cumbre de Ontologías

La recomendación de la Cumbre de Ontología es incorporar la evaluación en todas las fases de su ciclo de vida, que incluye las siguientes: desarrollo de requisitos, análisis ontológico, diseño de ontologías, diseño de sistemas, desarrollo de ontologías, desarrollo e integración de sistemas, despliegue, y operación y mantenimiento. En el contexto de este modelo, se considera la evaluación de cinco aspectos de su calidad: inteligibilidad, fidelidad, artesanía, adecuación e implementación [Neuhaus et al. 2013].

OntoQualitas se estructura a partir del requisito que las ontologías deben permitir el intercambio de información entre diferentes contextos sin imponer un sentido global de la misma a todos los contextos involucrados [Rico et al. 2014]. De este se derivan tres requisitos específicos: (i) la representación de la información intercambiada debe ser formal, (ii) sólo la información estrictamente necesaria para el intercambio debe ser representada, y (iii) la representación debe permitir una correcta interpretación de la información intercambiada en todos los contextos.

El primer requisito refiere al hecho que las ontologías utilizadas para definir la semántica de la información intercambiada deben estar implementadas en un lenguaje ejecutable e interpretable por una máquina. El segundo requisito tiene dos aspectos: completitud y concisión. El tercer requisito tiene tres aspectos: corrección sintáctica, corrección semántica y corrección de representación, cuyo objetivo es evaluar la correspondencia entre las entidades de dominio, sus relaciones y características, y los elementos utilizados para su representación.

Los requisitos planteados en OntoQualitas, que dan sustento a las medidas de calidad, contemplan parte de los aspectos presentados por la Cumbre de Ontología.

3. Herramientas de Apoyo para la Evaluación de Ontologías

Si bien las medidas propuestas inicialmente en OntoQualitas fueron pensadas para evaluar la calidad de una ontología ya definida y con un propósito particular, en el presente trabajo se muestra que las mismas y sus derivadas, son de utilidad para evaluar la calidad en diferentes fases de su ciclo de vida.

Las medidas propuestas se pueden dividir en tres grupos:

- Medidas de calidad: se obtienen automáticamente mediante un cálculo de proporción e indican un aspecto de calidad.

- Medidas de calidad pendientes de automatización: requieren de un entendimiento semántico del lenguaje natural, un razonamiento humano o un marco de referencia automatizado para poder implementarlas e inferir sobre un aspecto de calidad.
- Advertencias: requieren de la interpretación humana para decidir sobre la calidad de una ontología. En este grupo se encuentran medidas cuyo valor, si es alto, en algunas circunstancias implica buena calidad y en otras lo contrario; solo una persona le dará la interpretación adecuada.

A continuación se describe, para cada aspecto a evaluar, el conjunto de medidas y advertencias.

3.1. Completitud

La completitud refiere a la extensión, grado, cantidad o cobertura para la cual la información en una ontología cubre la información del mundo real [Gómez-Pérez 2001]. Las medidas relativas a este aspecto se presentan en la Tabla 1.

Tabla 1. Medidas de completitud

Medida	
Condiciones necesarias y suficientes	$CNS=CHCNS/CH, CH \neq \emptyset$
CHCNS: N° de clases hoja con al menos un conjunto de condiciones necesarias y suficientes.	
CH: N° de clases hoja, sin considerar la clase raíz (Thing).	
Restricciones de existencia y universalidad	$REU=PREU/PRU, PRU \neq \emptyset$
PREU: N° de propiedades con restricciones de existencia y universalidad a lo largo de la misma propiedad.	
PRU: N° de propiedades con restricciones universales.	
Dominio y rango de las relaciones	$DRR=RNJDR/RNJ, RNJ \neq \emptyset$
RNJDR: N° de relaciones no-jerárquicas con dominio y rango especificado.	
RNJ: N° de relaciones no-jerárquicas.	
No omisión de partición de subclases	$NOPS=PSCRD/CSC, CSC \neq \emptyset$
PSCRD: N° de particiones de subclase definidas sobre clases con la respectiva restricción de disyunción.	
CSC: N° de clases con un conjunto de subclases directas identificadas, sin considerar la clase raíz (Thing).	
No omisión de partición exhaustiva de subclases	$NOPEs=CSAC/CSCD, CSCD \neq \emptyset$
CSAC: N° de clases con un conjunto de subclases directas disjuntas y con axioma de cobertura.	
CSCD: N° de clases con un conjunto identificado de subclases directas disjuntas.	
*Cobertura de clases	$Cobertura(O_c;MR_c)= O_c \cap MR_c / MR_c , MR_c \neq \emptyset$
O _c : Conjunto de clases en la ontología.	
MR _c : Conjunto de clases en el marco de referencia.	
*Cobertura de relaciones entre clases	$Cobertura(O_{rc};MR_{rc})= O_{rc} \cap MR_{rc} / MR_{rc} , MR_{rc} \neq \emptyset$
O _{rc} : Conjunto de relaciones entre clases en la ontología.	
MR _{rc} : Conjunto de relaciones entre clases en el marco de referencia.	

* Medida pendiente de automatización

Tabla 1. Medidas de completitud (continuación)

Medida	
*Cobertura de relaciones entre instancias	$Cobertura(O_{ri};MR_{ri})= O_{ri} \cap MR_{ri} / MR_{ri} $ O_{ri} : Conjunto de relaciones entre instancias en la ontología. MR_{ri} : Conjunto de relaciones entre instancias en el marco de referencia.
*Cobertura de instancias	$Cobertura(O_i;MR_i) = O_i \cap MR_i / MR_i $, $MR_i \neq \emptyset$ O_i : Conjunto de instancias en la ontología. MR_i : Conjunto de instancias en el marco de referencia.
*Cobertura de características de clases	$Cobertura(O_{cc};MR_{cc})= O_{cc} \cap MR_{cc} / MR_{cc} $ O_{cc} : Conjunto de características de clases en la ontología. MR_{cc} : Conjunto de características de clases en el marco de referencia.
*Cobertura de dimensiones	$Cobertura(O_{dcc};MR_{dcc})= O_{dcc} \cap MR_{dcc} / MR_{dcc} $, $MR_{dcc} \neq \emptyset$ O_{dcc} : Dimensiones usadas para especificar características contextuales de la entidad en la ontología. MR_{dcc} : Dimensiones usadas para especificar características contextuales de la entidad en el marco de ref.

* Medida pendiente de automatización

Respecto de la Completitud, se proponen las siguientes advertencias, de utilidad durante las distintas actividades en la fase de desarrollo de la ontología:

- A1. Clases hoja que no poseen al menos un conjunto de condiciones necesarias y suficientes
- A2. Restricciones universales sin las correspondientes restricciones existenciales.
- A3. Relaciones no-jerárquicas sin dominio y/o rango especificado.
- A4. Clases con un conjunto identificado de subclases directas, sin las correspondientes restricciones de disyunción.
- A5. Clases con un conjunto de subclases directas disjuntas y sin axioma de cobertura.

3.2. Concisión

La concisión refiere a si una ontología no almacena definiciones innecesarias o inútiles, si no existen redundancias explícitas entre las definiciones y las redundancias no se pueden deducir usando otras definiciones y axiomas [Gómez-Pérez 2001]. Las medidas de concisión se presentan en la Tabla 2.

Tabla 2. Medidas de concisión

Medida	
Clases semánticamente diferentes	$CSD = 1 - CMD / C$, $C \neq \emptyset$ CMD : N° de clases con la misma definición formal que otras clases en la ontología. C : N° de clases en la ontología, sin considerar la clase raíz (Thing).
Individuos semánticamente diferentes	$ISD = 1 - IMD / I$, $I \neq \emptyset$ IMD : N° de individuos con la misma definición formal que otros individuos en la ontología. I : N° de individuos en la ontología.

Tabla 2. Medidas de concisión (continuación)

Medida	
No redundancia de relaciones subclase-de RSCR: N° de relaciones subclase-de redundantes en la ontología. RJ: N° de relaciones jerárquicas en la ontología, sin considerar las relaciones jerárquicas que tienen a la clase raíz Thing como padre.	$NRRS = 1 - RSCR / RJ, RJ \neq \emptyset$
*Otras relaciones no redundantes RNJR: N de relaciones no-jerárquicas redundantes en la ontología. RNJ: N de relaciones no-jerárquicas.	$ORNR = 1 - RNJR / RNJ, RNJ \neq \emptyset$
No redundancia de relaciones individuo-de RIRO: Número de relaciones individuos-de redundantes en la ontología. RIO: Número de relaciones individuos-de en la ontología.	$NRRI = 1 - RIRO / RIO, RIO \neq \emptyset$
*Precisión de clases O _c : Conjunto de clases en la ontología, sin tener en cuenta la clase raíz (Thing). MR _c : Conjunto de clases en un marco de referencia.	$Precision(O_c;MR_c) = (O_c \cap MR_c) / O_c, O_c \neq \emptyset$
*Precisión de relaciones entre clases O _r : Conjunto de relaciones entre clases en la ontología. MR _r : Conjunto de relaciones entre clases en un marco de referencia	$Precision(O_r;MR_r) = (O_r \cap MR_r) / O_r, O_r \neq \emptyset$
*Precisión de características de clase O _{cc} : Conjunto de características de clases en la ontología. MR _{cc} : Conjunto de características de clases en un marco de referencia.	$Precision(O_{cc};MR_{cc}) = (O_{cc} \cap MR_{cc}) / O_{cc}, O_{cc} \neq \emptyset$
*Precisión de individuos O _i : Conjunto de individuos en la ontología. MR _i : Conjunto de individuos en un marco de referencia.	$Precision(O_i;MR_i) = (O_i \cap MR_i) / O_i, O_i \neq \emptyset$

* Medida pendiente de automatización

Con el propósito de ayudar a evaluar el aspecto concisión durante el desarrollo de la ontología, se proponen las siguientes advertencias:

- A6. Clases con la misma definición formal que otras clases.
- A7. Individuos con la misma definición formal que otros individuos.
- A8. Relaciones subclase-de redundantes.
- A9. Relaciones no-jerárquicas con el mismo dominio, rango y características.

3.3. Corrección Sintáctica

La corrección sintáctica intenta evaluar la calidad de la ontología de acuerdo a la forma en que está escrita, es decir, la corrección y la amplitud de la sintaxis utilizada [Burton-Jones et al. 2005]. Las medidas de corrección sintáctica se dan en la Tabla 3.

Tabla 3. Medidas de corrección sintáctica

Medida	
Legalidad Xb: Total de reglas sintácticas violadas. NS: N° total de sentencias en la ontología.	$L = 1 - Xb/NS, NS \neq \emptyset$

Tabla 3. Medidas de corrección sintáctica (continuación)

Medida	
Riqueza	$R=Z/Y, Y \neq \emptyset$
Z: N° de características sintácticas usadas en la ontología.	
Y: N° de características sintácticas disponibles en el lenguaje de la ontología.	

La siguiente advertencia puede guiar el desarrollo de la ontología para que sea sintácticamente correcta:

- A10. Reglas sintácticas violadas.

3.4. Corrección Semántica

La corrección semántica se ocupa del vocabulario utilizado para representar las entidades, relaciones y características, y la corrección de la representación de la información intercambiada en la ontología. Las medidas se dan en la Tabla 4.

Tabla 4. Medidas de corrección semántica

Medida	
*Interpretabilidad	$IN=SW/PCP, PCP \neq \emptyset$
SW: N° de palabras usadas para definir clases y propiedades que tienen al menos un sentido listado en WordNet.	
PCP: N° de palabras diferentes usadas para definir clases y propiedades en la ontología.	
*Claridad	$CL = SW / TSTP$
SW: N° de palabras usadas para definir clases y propiedades que tienen al menos un sentido listado en WordNet.	
TSTP: Total de sentidos que tienen las SW palabras en WordNet.	
Sin errores de circularidad en distancia 0	$SECD0 = 1 - Ciclos(0;0)/RJ, RJ \neq \emptyset$
Ciclos(0;0): N° de ciclos detectados entre una clase con ella misma.	
RJ: N° de relaciones jerárquicas directas, sin considerar las relaciones jerárquicas que tienen a la clase raíz Thing como padre.	
Sin errores de circularidad en distancia 1	$SECD1 = 1 - Ciclos(0;1)/RJ, RJ \neq \emptyset$
Ciclos(0;1): N° de ciclos detectados entre una clase y una clase adyacente.	
RJ: N° de relaciones jerárquicas directas, sin considerar las relaciones jerárquicas que tienen a la clase raíz Thing como padre.	
Sin errores de circularidad en distancia d	$SECDd = 1 - Ciclos(0;d)/RJ, RJ \neq \emptyset$
Ciclos(0;d): N° de ciclos detectados entre una clase y otra a d clases.	
RJ: N° de relaciones jerárquicas directas, sin considerar las relaciones jerárquicas que tienen a la clase raíz Thing como padre.	
Partición de subclases sin individuos comunes	$PSSIC = 1 - ICPS/I, I \neq \emptyset$
ICPS: N° de individuos que pertenecen a más de una subclase de una partición disyunta en la ontología.	
I: N° de individuos en la ontología.	
Partición de subclases sin clases comunes	$PSSCC = 1 - CCPS/C, C \neq \emptyset$
CCPS: N° de clases que pertenecen a más de una subclase de una partición disyunta en la ontología	
C: N° de clases en la ontología, sin tener en cuenta la clase raíz (Thing)	

* Medida pendiente de automatización

Tabla 4. Medidas de corrección semántica (continuación)

Medida
Partición exhaustiva de subclases sin individuos comunes $PESSIC=1-PESIC/I, I \neq \emptyset$ PESIC: N° de individuos que pertenecen a más de una subclase de una partición disyunta y exhaustiva en la ontología. I: N° de individuos en la ontología.
Partición exhaustiva de subclases sin clases comunes $PESSCC=1-CCPES/C, C \neq \emptyset$ CCPES: N° de clases que pertenecen a más de una subclase de una partición disyunta y exhaustiva en la ontología. C: N° de clases en la ontología, sin tener en cuenta la clase raíz (Thing).
Partición exhaustiva de subclases sin individuos externos $PESSIE=1-IEPES/I, I \neq \emptyset$ IEPES: N° de individuos de una clase base que no pertenecen a ninguna de las subclases de la partición disyunta y exhaustiva de la clase base. I: N° de individuos en la ontología.

El aspecto de corrección semántica puede ser evaluado durante la etapa de desarrollo de la ontología con las siguientes advertencias:

- A11. Palabras usadas para definir clases y propiedades que no tienen un sentido listado en WordNet [Princeton University 2010].
- A12. Palabras usadas para definir clases y propiedades que tienen más de un sentido listado en WordNet.
- A13. Clases con relaciones jerárquicas directas a ella misma.
- A14. Clases con ciclos detectados entre una clase y una clase adyacente.
- A15. Individuos que pertenecen a más de una subclase de una partición disyunta.
- A16. Clases que pertenecen a más de una subclase de una partición disyunta.
- A17. Individuos que pertenecen a más de una subclase de una partición disyunta y exhaustiva.
- A18. Clases que pertenecen a más de una subclase de una partición disyunta y exhaustiva
- A19. Individuos de una clase base que no pertenecen a ninguna de las subclases de la partición disyunta y exhaustiva de la clase base.

Las medidas Interpretabilidad y Claridad y las advertencias A11 y A12 pueden ser de utilidad, además, durante la fase de análisis ontológico, si durante esta fase se emplea una herramienta de edición de ontologías.

3.5. Corrección de Representación

La corrección de representación (Tabla 5) refiere a la calidad de las correspondencias de entidades, relaciones y características en los elementos de la ontología evaluada.

Tabla 5. Medidas de corrección de representación

Medida
* Principio de entidades $PE = \sum_k \alpha_k / E, E \neq \emptyset$ E: N° de entidades.

Tabla 5. Medidas de corrección de representación (continuación)

Medida	
* Principio uso previsto de entidades	$PU = \sum_k \alpha_k / U, U \neq \emptyset$
U: N° de usos previstos para todas las entidades.	
* Principio relaciones entre entidades	$PR = \sum_k \alpha_k / RE, RE \neq \emptyset$
RE: N° de relaciones identificadas para todas las entidades.	
* Principio características simples como entidades	$PCS = \sum_k \alpha_k / CS, CS \neq \emptyset$
CS: N° de características simples identificadas para todas las entidades.	
* Principio características simples medibles de entidades	$PCM = \sum_k \alpha_k / CM, CM \neq \emptyset$
CM: N° de características simples medibles identificadas para todas las entidades	
* Principio características complejas de entidades	$PCC = \sum_k \alpha_k / CC, CC \neq \emptyset$
CC: N° de características complejas identificadas para todas las entidades.	
* Principio de características comunes de entidades	$PCc = \sum_k \alpha_k / Cc \neq \emptyset$
Cc: N° de características comunes identificadas para todas las entidades.	

$\alpha_k = 0$ si el elemento k no está representado; $\alpha_k = 0,5$ si el elemento k está representado en forma incompleta; $\alpha_k = 1$ si el elemento k está bien representado.

* Medida pendiente de automatización.

4. Avances sobre las Advertencias y Medidas

Las advertencias tienen por objetivo exponer posibles errores y corresponden a listar aquellos elementos de la ontología que no cumplen las condiciones requeridas en la actividad de modelado formal, durante la fase de desarrollo de la ontología [Neuhaus et al. 2013]. A diferencia de la propuesta realizada por Poveda-Villalón et al. [2012], se intenta ir más allá de una simple enumeración de elementos ontológicos afectados, siempre que sea posible.

Un razonador puede reconocer qué conceptos corresponden a cuáles definiciones y, así, inferir conocimiento nuevo [Horridge 2009]. Las siguientes advertencias ayudan a detectar aquellos elementos de la ontología sobre los cuales no se podrá razonar.

Advertencia A1: La medida Condiciones Necesarias y Suficientes (Tabla 1) sirve para evaluar la capacidad de razonamiento sobre una jerarquía de clases. Sin embargo, las clases con condiciones necesarias y suficientes (definidas) son muy costosas computacionalmente ya que el razonador tiene que considerar qué cosas deben ser clasificadas bajo ellas, así como también el lugar donde ellas mismas deberían ser clasificadas [Rector 2005]. La determinación de la conveniencia de describir una clase como definida es una tarea que compete al desarrollador, luego para darle soporte a su toma de decisión, se listan todas las clases hoja de la ontología que no poseen al menos un conjunto de condiciones necesarias y suficientes.

Advertencia A2: Uno de los errores más comunes es el uso de restricciones universales sin las correspondientes restricciones existenciales [Rector et al. 2004]. Las restricciones universales establecen que si una relación existe, ésta debe ser con un individuo que es miembro de una clase específica; pero no garantizan la existencia de la relación. Por esta razón, se listan todas las propiedades con restricciones universales sin las correspondientes restricciones existenciales.

Redundancia es un tipo de error que ocurre cuando se redefinen expresiones que ya fueron explícitamente definidas o que pueden ser inferidas usando otras definiciones [Gómez-Pérez 2001]. Esto ocurre cuando hay dos o más clases o instancias en la ontología con la misma definición formal, o cuando hay más de una definición explícita de cualquiera de las relaciones jerárquicas.

La redundancia de relaciones subclase-de ocurre entre clases cuando tienen más de una relación subclase-de entre ellas. Se puede distinguir entre redundancia directa e indirecta (Figura 1). En OntoQualitas, la medida No Redundancia de Relaciones Subclase-de (NRRS, Tabla 2) tiene por objetivo medir la proporción de relaciones subclase-de no redundantes en la ontología y aporta a la evaluación del aspecto concisión [Tibaldo et al. 2015].

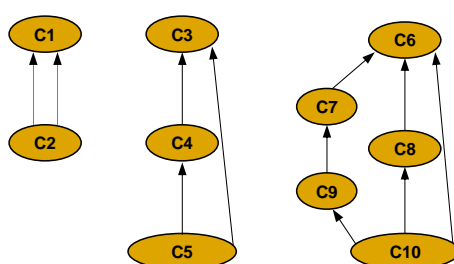


Figura 1. Ejemplos de redundancia directa e indirecta de relaciones subclase-de

La concisión no implica ausencia de redundancias. A veces, un cierto grado de redundancia controlada puede ser útil en las definiciones. En este sentido se puede observar que la detección de redundancias directas permite deducir algún grado de baja calidad en la ontología. Por el contrario, la presencia de redundancias indirectas no posibilita inferir sobre la calidad de la ontología, pero es útil para alertar a un desarrollador sobre su análisis y determinación de su necesidad. En consecuencia, se propone desdoblar la medida NRRS en las siguientes:

- La medida de calidad Relaciones Subclase-de Sin Redundancias Directas (RSSRD), mide la proporción de relaciones subclase-de que no son redundancias directas. Se define como:

$$RSSRD = 1 - RSCDR / RJ$$

donde RSCDR es el número de relaciones subclase-de directas redundantes en la ontología y RJ es el número de relaciones jerárquicas directas en la ontología sin considerar las relaciones jerárquicas directas que tienen a la clase raíz Thing como padre. Los valores de esta medida varían en el rango [0, 1] y su valor óptimo es uno.

Los caminos (sucesión de clases y relaciones subclase-de) que representan redundancias directas tienen longitud uno (1). En este caso, todos los caminos, excepto uno cualquiera de ellos, se consideran redundantes.

- Las relaciones subclase-de con redundancias indirectas no se pueden medir, ante la imposibilidad de determinar en forma automática cuál o cuáles de ellas son redundantes. Por ejemplo, en la Figura 1 se muestran tres caminos alternativos para alcanzar la clase C6 desde la clase hija C10. Se podría suponer que el camino más largo (C10 – C9 – C7 – C6) es no redundante, dado que éste es el

que más detalle ofrece sobre las características de la clase hija. Sin embargo, si se considera alguna cuestión de performance por ejemplo, se podría seleccionar al camino más corto (C10 – C6) como no redundante, por requerir menor esfuerzo computacional.

Con el fin de brindar apoyo en la detección de relaciones subclase-de redundantes, se muestran a modo de advertencia las situaciones conflictivas identificadas, junto con información relevante en dos tablas diferentes. En las relaciones subclase-de se pueden identificar niveles. Visto como un grafo, el nodo raíz (un nodo sin padres) pertenece al nivel uno (1) por convención; las clases que se derivan de él corresponden al nivel dos (2); las subclases de cada una de las clases anteriores, nivel tres (3), y así sucesivamente. En forma tabular se muestra cada clase cuyo nivel es tres (3) o superior, las trayectorias alternativas (sucesión de arcos que pasan por una serie de vértices una sola vez) para llegar a una clase padre de nivel $n - 2$ o inferior.

Además, una segunda tabla resume por cada grupo de tres o más niveles, cuántas clases (n) y relaciones subclase-de (r) pertenecen a dicho grupo y cuál es la cantidad posible de redundancias ($r - n + 1$). Los valores positivos mayores señalan grupos de niveles con más cantidad de relaciones redundantes. El comparar estos valores entre niveles consecutivos permite identificar dónde se están dando las redundancias.

Similarmente a la redundancia de relaciones subclase-de, la redundancia de relaciones instancia-de ocurre entre una instancia y una clase cuando hay más de una relación instancia-de entre ellas. También se distingue entre redundancia directa e indirecta y se propone desdoblar la medida No Redundancia de Relaciones Individuo-de (Tabla 2) en las siguientes:

- La medida de calidad Relaciones Individuo-de Sin Redundancias Directas (RISRD), cuyo objetivo es medir la proporción de relaciones individuo-de que no son redundancias directas. Se define como:

$$RISRD = 1 - RIDR / RIO$$

donde RIDR es el número de relaciones individuo-de directas redundantes en la ontología y RIO es el número de relaciones individuo-de en la ontología. Los valores de esta medida varían en el rango $[0, 1]$ y el valor óptimo es uno.

- Las Relaciones Individuo-de Con Redundancias Indirectas (RICRI) no se pueden medir por las mismas razones que no es posible medir las relaciones subclase-de con redundancias indirectas.

Las medidas antes descriptas permiten evaluar conocimiento taxonómico en las ontologías. Sin embargo existen otros tipos de relaciones, cuya definición también puede ser redundante en el sentido que redefinen expresiones que ya fueron explícitamente definidas. En el framework OntoQualitas se propone la medida Otras Relaciones No Redundantes (ORNR, Tabla 2), cuyo objetivo es medir la proporción de relaciones no jerárquicas no redundantes existentes en la ontología [Tibaldo et al. 2015].

En la determinación de la cantidad de relaciones no-jerárquicas redundantes en la ontología se distinguen aquellas propiedades de objeto que han sido identificadas como equivalentes a otras propiedades de objeto y aquellas que no lo son. El primer caso corresponde a una afirmación explícita de que las propiedades de objeto

involucradas son semánticamente equivalentes [Motik et al. 2012]. Por ende, todas menos una de las propiedades de objeto equivalentes son redundantes. Por ejemplo, si las propiedades de objeto `tieneHermano` y `tieneHermanoVaron` han sido definidas como equivalentes, una cualquiera de ellas es redundante.

En el segundo caso, cuando no se han identificado explícitamente propiedades de objeto equivalentes, es necesario realizar un análisis de todas las propiedades de objeto con exactamente el mismo dominio y rango. Si todas sus características (funcional, inversa funcional, transitiva, simétrica, asimétrica, reflexiva e irreflexiva) son idénticas es probable que lo único que las diferencie sea su nombre. Aún así, no es posible asegurar en forma automática si dichas propiedades son redundantes o no, como se puede observar en los ejemplos de la Figura 2.

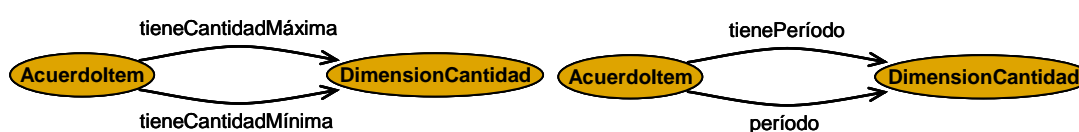


Figura 2. Propiedades de objeto no redundantes y redundantes

En consecuencia se propone desdoblar la medida ORNR en dos:

- La medida de calidad Otras Relaciones Sin Equivalencias (ORSE), cuyo objetivo es medir la proporción de relaciones no jerárquicas sin equivalencias explícitas existentes en la ontología, que se define como:

$$ORSE = 1 - RNJSE / RNJ$$

donde RNJSE es la cantidad de relaciones no-jerárquicas sin equivalencias definidas en la ontología y RNJ es el número de relaciones no-jerárquicas en la ontología. Los valores de esta medida varían en el rango [0, 1] y su valor óptimo es uno.

- Otras Relaciones Con Posibles Redundancias no se pueden medir en forma automática, por las razones expuestas en párrafos precedentes. Para tener una medida útil a la hora de inferir la calidad de una ontología, es necesario realizar un análisis semántico de los nombres de las propiedades de objeto. Hasta tanto se pueda proveer esta funcionalidad, la medida queda pendiente de automatización y en una ventana se muestran posibles propiedades de objeto redundantes para que un desarrollador determine si son redundantes o no.

5. Arquitectura de OntoQualitas

La aplicación OntoQualitas se desarrolla en C# y está dividida en dos proyectos: el framework base, donde se lleva a cabo toda la lógica de la aplicación (Ontoqualitas.Core), y una aplicación de ventana (Ontoqualitas.Form) que presenta toda la información al usuario. Con esta aplicación se sometieron a evaluación distintas ontologías, entre ellas las analizadas manualmente por Tibaldo et al. [2015]. La Figura 3 ilustra parte de la evaluación de la ontología BPMN 2.0 [Rospocher et al. 2014]. Los valores en rojo (segunda columna de la ventana) indican la cantidad de advertencias, que se visualizan haciendo clic sobre el número correspondiente. El guión (“-”) indica que no se puede calcular la medida.

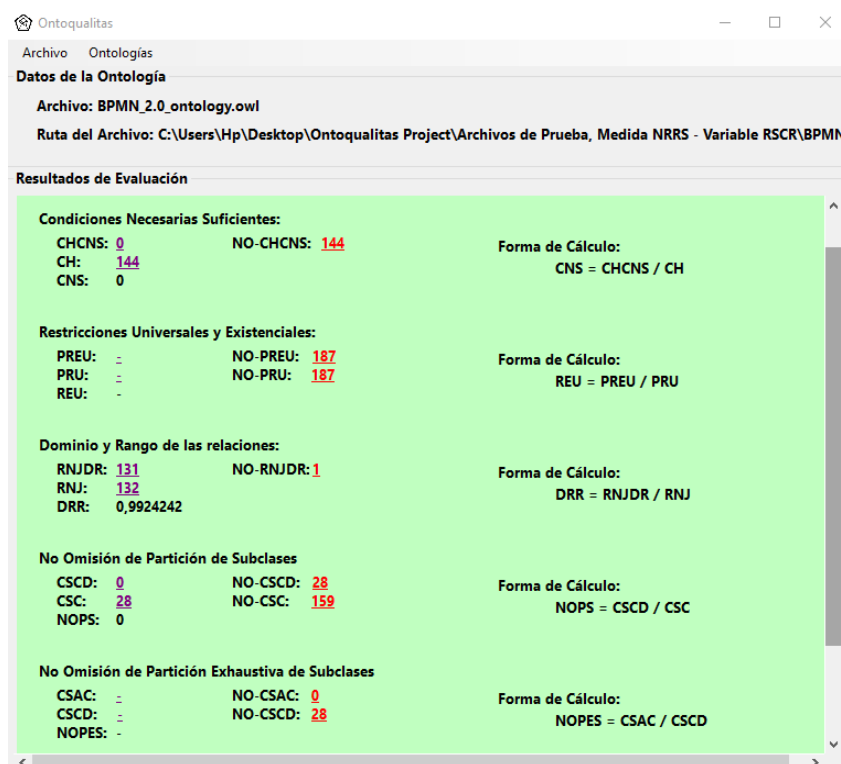


Figura 3. Evaluación de BPMN 2.0

Además, la Tabla 6 muestra, a modo de ejemplo, las medidas de completitud obtenidas con la aplicación OntoQualitas sobre una ontología construida sin el apoyo de las advertencias (O1 [Tibaldo et al. 2015]) en comparación con una ontología desarrollada por el mismo grupo de personas, siguiendo el mismo propósito y con el soporte de la herramienta (O2). Se puede observar que todas las medidas mejoraron sustancialmente, excepto DRR donde la ontología O2 importó otras ontologías ya desarrolladas sin el apoyo de la aplicación OntoQualitas.

Tabla 6. Comparativa medidas de completitud

Medida	O1	O2
Condiciones Necesarias y Suficientes (CNS)	0	1
Restricciones de Existencia y Universalidad (REU)	0	1
Dominio y Rango de las Relaciones (DRR)	1	0,890625
No Omisión de Partición de Subclases (NOPS)	0	0,16666
No Omisión de Partición Exhaustiva de Subclases (NOPEs)	-	0,5

6. Conclusiones y Trabajos Futuros

Partiendo de la recomendación de la Cumbre de Ontología, que la evaluación de las mismas debe realizarse durante todas las fases de su ciclo de vida, se analizaron las medidas propuestas en el framework OntoQualitas y se propuso un conjunto de herramientas de utilidad para la evaluación durante las fases de diseño, desarrollo y posterior mantenimiento de las ontologías. Calcular las medidas de calidad presentadas en OntoQualitas de forma manual resulta un proceso largo, tedioso y sujeto al error humano. La automatización fue un procedimiento necesario y ventajoso, ya que

posibilitó otra visión de algunas medidas y la derivación de otras nuevas. Además permitió dar un soporte mediante listados de advertencias e información relevante acerca de situaciones que pueden ser errores en el desarrollo de las ontologías.

Algunas de las medidas propuestas en OntoQualitas requieren de un procesamiento semántico del lenguaje natural, otras necesitan de un marco de referencia automatizado para poder implementarlas e inferir sobre un aspecto de calidad. El trabajo futuro se enfocará en estos dos aspectos.

Referencias

- Bhatia, M.P.S., Kumar, A. and Beniwall, R. (2016). Ontologies for Software Engineering: Past, Present and Future. In *Indian Journal of Science and Technology*, 9(9), DOI: 10.17485/ijst/2016/v9i9/71384
- Brank, J., Grobelnik, M. and Mladeni, D. (2005), "A survey of ontology evaluation techniques", In: Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005), Ljubljana, Slovenia, p. 166-169.
- Burton-Jones, A., Storey, V.C., Sugumaran, V. and Ahluwalia, P. (2005). A semiotic metrics suite for assessing the quality of ontologies. In *Data & Knowledge Engineering*, 55(1), p. 84-102.
- Colomb, R. (2002), "Quality of Ontologies in Interoperating Information Systems", Technical Report 18/02 ISIB-CNR, National Research Council, Institute of Biomedical Engineering, Padova, Italy, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.2321&rep=rep1&type=pdf>.
- Duque-Ramos, A., Fernández-Breis, J., Stevens, R. and Aussenac-Gilles, N. (2011). OQuaRE: a SQuaRE-based approach for evaluating the quality of ontologies. In *Journal of Research and Practice in Information Technology*, 43(2), p. 159-176.
- Gašević, D., Kaviani, N. and Milanović, M. (2009), "Ontologies and Software Engineering", In: Handbook on Ontologies, Edited by S. Staab and R. Studer, Springer Berlin Heidelberg, p. 593-615.
- Gómez-Pérez, A. (2001). Evaluation of ontologies. In *Int. J. Intell. Syst.*, 16, p. 391-409.
- Horridge, M. (2009), "A practical guide to building OWL ontologies using Protégé 4 and COODE tolos", Edition 1.2, University of Manchester, http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_2.pdf
- ISO/IEC (2014), ISO/IEC 25000:2014. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE), <http://www.iso.org>
- Motik, B., Patel-Schneider, P.F., and Parsia, B. (2012), OWL 2 Web Ontology Language. Structural Specification and Functional-Style Syntax (Second Edition). W3C Recommendation 11 December 2012, The World Wide Web Consortium. <http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/>
- Neuhaus, F., Vizedom, A., Baclawski, K., Bennett, M., Dean, M., Denny, M., Grüniger, M., Hashemi, A., Longstreth, T., Obrst, L., Ray, S., Sriram, R., Schneider, T., Vegetti, M., West, M. and Yim, P. (2013). Towards ontology evaluation across the life cycle. In *Applied Ontology*, 8(3), p. 179-194.

- Object Management Group (2014), Ontology Definition Metamodel Version 1.1, OMG Document Number: formal/2014-09-02, <http://www.omg.org/spec/ODM/1.1/>
- Poveda-Villalón, M., Suárez-Figueroa, M.C. and Gómez-Pérez, A. (2012), “Validating ontologies with OOPS!”, 18th International Conference on Knowledge Engineering and Knowledge Management (EKAW2012), Galway, Ireland.
- Princeton University (2010), Wordnet, <http://wordnet.princeton.edu/>
- Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H. and Wroe, C. (2004), “OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns”, In: Engineering Knowledge in the Age of the Semantic Web, Edited by E. Motta, N. Shadbolt, A. Stutt, and N. Gibbins, Springer Berlin Heidelberg, p. 63-81.
- Rector, A. (2005), “OWL Biomedical Ontology Tutorial”, University of Manchester, <http://www.cs.man.ac.uk/~rektor/modules/cds/OWL%20Biomedical%20Ontology%20Tutorial-v1.pdf>
- Rico, M., Caliusco, M.L., Chiotti, O. and Galli, M.R. (2014). OntoQualitas: A framework for ontology quality assessment in information interchanges between heterogeneous systems. In *Computers in Industry*, 65(9), p. 1291-1300.
- Rospoche, M., Ghidini, C., and Serafini, L. (2014), “An ontology for the Business Process Modelling Notation”, In: Formal Ontology in Information Systems, Proceedings of the Eighth International Conference, FOIS2014, September, 22-25, 2014, Rio de Janeiro, Brazil, IOS Press, vol. 267, p. 133-146.
- Ruiz, F. and Hilera, J. R. (2006), “Using Ontologies in Software Engineering and Technology”, In: Ontologies for Software Engineering and Software Technology, Edited by C. Calero, F. Ruiz and M. Piattini, Springer, New York, p. 49-102.
- Stvilia, B. (2007). A model for ontology quality evaluation. In *First Monday*, 12(12).
- Supekar, K., Patel, C. and Lee, Y. (2004), “Characterizing quality of knowledge on Semantic Web”, In: Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, Edited by V. Barr and Z. Markov, Florida, USA, AAAI Press.
- Tartir, S., Arpinar, I.B., Moore, M., Sheth, A.P. and Aleman-Meza, B. (2005), “OntoQA: metricbased ontology quality analysis”, In: Proceedings of IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources, Edited by D. Caragea, V. Honavar, I. Muslea and R. Ramakrishnan, Houston, TX, IEEE Computer Society, p. 45-53.
- Tibaldo, M., Wilkinson, A., Taverna, M. L., Rico, M. and Galli, M. R. (2015), “Redefinition and Statistical Analysis of Measures for Evaluating the Quality of Ontologies”, In: Argentine Symposium on Ontologies and their Applications 2015, Edited by H. Leone, P. Fillotrani and M. Vegetti, CEUR, vol. 1449, p. 51-60.
- Zhao, Y., Dong, J. and Peng, T. (2009). Ontology Classification for Semantic-Web-Based Software Engineering. In *IEEE Transactions On Services Computing*, 2(4).