

Ontologías en la ingeniería de *software*

[8.1] ¿Cómo estudiar este tema?

[8.2] Introducción

[8.3] Ingeniería de la ontología

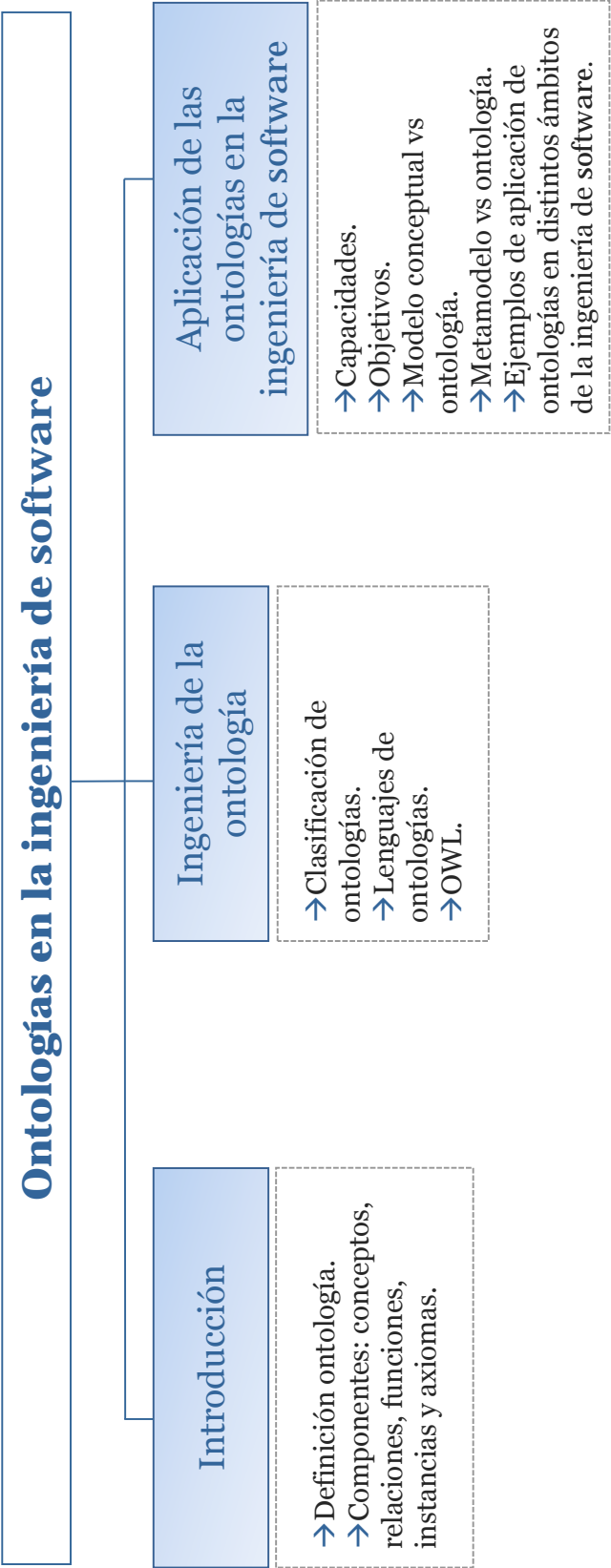
[8.4] Aplicación de las ontologías en la ingeniería de *software*



8

T E M A

Esquema



Ideas clave

8.1. ¿Cómo estudiar este tema?

Para estudiar este tema lee las **Ideas clave** que encontrarás a continuación.

En este tema se estudian los principios de la **ontologías** y la **ingeniería de la ontología** (del inglés *Ontological Engineering*) aplicadas al área de la ingeniería de *software*.

8.2. Introducción

En diversas áreas de las tecnologías de la información (TI) se realizan varios esfuerzos dirigidos a facilitar la reutilización de recursos: información, conocimiento y *software*. Todo ello con el fin de gestionar la complejidad inherente a los sistemas de información actuales. De acuerdo a Rodríguez (2000), son dos los factores que determinan la posibilidad de reutilización:

- » La producción de los componentes teniendo en cuenta criterios que favorezcan la reutilización: construcción de repositorios de elementos reutilizables, potenciación de las características dirigidas a la reutilización (de propósito general, abstracción, documentación, homogeneidad en el diseño, consistencia en el vocabulario...).
- » La incorporación de la reutilización como elemento del proceso de construcción de los sistemas de información.

En este contexto, y en concreto, en el ámbito del desarrollo de sistemas *software*, un tipo de recurso que ha ido adquiriendo cada vez más popularidad es el de las **ontologías**. En los últimos años ha ido produciéndose un interés que va en aumento en lo que se refiere a la construcción, intercambio y utilización de las ontologías con el fin de mejorar la reutilización.

El término «ontología» tiene su origen en el campo de la Filosofía conocida como metafísica, que trata con la naturaleza de las cosas que existen, es decir, el mundo real. Con la premisa de que todo lo que «existe» se puede representar, las ontologías se empezaron a adoptar y a utilizar a finales de los 80 en el área de inteligencia artificial

como medio para el intercambio (del inglés *sharing*) y la reutilización (del inglés *reuse*) de recursos. En la segunda mitad de los 90 se empiezan a aplicar también a la web para la inclusión de descripciones semánticas explícitas de recursos (contenidos y servicios), y actualmente, las ontologías constituyen un eje fundamental en las nuevas tecnologías tanto para la web semántica como para la ingeniería de *software*. La Figura 1 muestra los distintos niveles de representación para la semántica de contenidos.

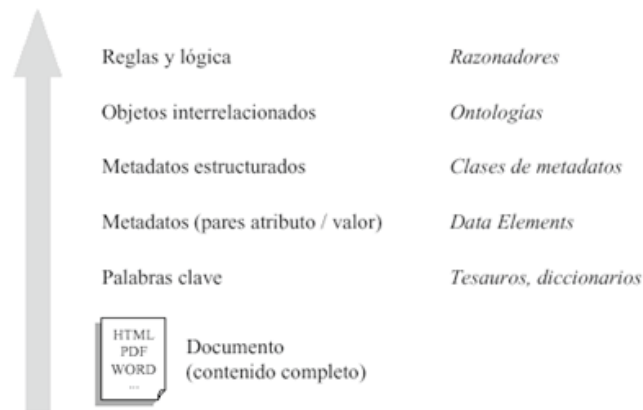


Figura 1. Semántica de contenidos: Niveles de representación.

Las ontologías van a permitir:

- » Compartir conocimiento común sobre la estructura de las cosas.
- » Reutilizar el conocimiento del dominio.
- » Realizar suposiciones explícitas sobre el dominio.

El objetivo tradicional de la investigación en ontologías consiste en dividir el mundo en conceptos (términos) con el fin de determinar las categorías o clasificaciones que definen los objetos en el mundo real. De hecho, este tipo de investigación juega un papel fundamental en el diseño de sistemas *software* aplicados en distintos dominios, como por ejemplo, universidades, bases militares, negocios, fábricas...

Todo ello motivado por la necesidad de entender, diseñar, construir y gestionar los sistemas *software* de una manera más efectiva y reutilizable. Por ejemplo, durante el modelado de análisis del dominio donde va a operar el *software*, se puede identificar, capturar y organizar toda la información relacionada la aplicación a través de ontologías evitando ambigüedades semánticas y contradicciones. El modelo que se obtiene como resultado de este proceso de análisis, ahora en forma de ontología, sigue siendo lo que se

ha denominado anteriormente en esta asignatura como «modelo de dominio» o «modelo de negocio».

Bajo esta perspectiva, las ontologías pueden definirse como representaciones explícitas de una conceptualización compartida (Gruber, 1995). El término «compartida» indica que la ontología captura algún conocimiento consensuado, mientras que el término «conceptualización» hace referencia a la vista simplificada, abstracta de un dominio específico compartido, es decir, el mundo real (Gašević, Kaviani y Devedžić, 2006). Además, tal y como destaca Olivé (2007), se podrían dar varias conceptualizaciones para un mismo dominio, y por tanto, varias ontologías.

Una definición más detallada es proporcionada por Neches et al. (1991) que, definen una ontología como «los términos básicos y las relaciones entre esos términos que reducen el vocabulario de un dominio de interés, así como las reglas que permiten combinar términos y relaciones para definir extensiones del vocabulario». Bajo otra perspectiva, Swartout, Patil, Knight y Russ (1997) definen una ontología como «un conjunto de términos estructurados de manera jerárquica que describen un dominio que se puede utilizar como fundamento en una base de conocimiento».

En la literatura se podrían encontrar muchas más definiciones y cada una centrándose en un punto de vista diferente del concepto de ontología. Por ejemplo, desde el punto de vista del ámbito de aplicación se encuentran las **ontologías generales** y las **ontologías de dominio**. Estas últimas tienen como ámbito de referencia un dominio semántico restringido mientras que las primeras no restringen su dominio.

Unified Medical Language System (UMLS) es una ontología muy conocida en el dominio médico, mientras que [WordNet](#) o [EuroWordNet](#), son dos ejemplos de ontologías generales. Desde el punto de vista de la cobertura conceptual, se encuentran las **ontologías enciclopédicas** y las **ontologías episódicas**. El sistema [Cyc](#) es un ejemplo muy conocido de ontología enciclopédica, que contiene una gran base de conocimiento representada desde un punto de vista técnico (Lenat, 1995). Tiene unos 100.000 términos, con más o menos un millón de datos o aserciones, y un motor de razonamiento para extraer conclusiones sobre esos datos.

Además, la base de conocimiento de Cyc puede soportar múltiples ontologías. Desde el punto de vista de su funcionalidad existen ontologías orientadas o no a una tarea, genérica o específica, o a una aplicación concreta. Desde el punto de vista de la

granularidad de los términos de la ontología se puede encontrar desde ontologías con etiquetas terminológicas hasta ontologías en las que los términos poseen estructura interna (atributos, propiedades) más o menos informativa.

Desde el punto de vista del contenido de los términos se pueden encontrar **ontologías léxicas** y **ontologías conceptuales**. EuroWordNet puede ser considerada una ontología léxica, mientras que WordNet es una ontología léxico-conceptual y Cyc una ontología conceptual. Lo que las distingue es el grado de lexicalización de los términos de la ontología. En Cyc, por ejemplo, las unidades terminológicas de las que está formado son claramente conceptos, independientemente de que existan palabras o términos multipalabra que los definan. En EuroWordNet solo los conceptos lexicalizados se incorporaron a la ontología. En WordNet la situación es intermedia.

Se podrían definir otras muchas clasificaciones, pero independientemente de la clasificación de la ontología, lo normal es que se definan en base a una serie de componentes:

- » **Conceptos.** Los componentes básicos de una ontología van a ser los conceptos, los cuales se encuentran organizados en taxonomías. La información que se asocia a cada concepto puede ser muy variable, desde una simple etiqueta léxica hasta un conjunto potencialmente grande de atributos y valores.
- » **Relaciones.** Los conceptos pueden estar asociados por relaciones del tipo $R: C_1 \times C_2 \times \dots \times C_{n-1} \times C_n$. Es habitual que el formalismo solo permita representar relaciones binarias. Cada ontología suele permitir el uso de un número acotado y cerrado de relaciones. Cada una de las relaciones puede estar dotada de determinadas propiedades que determinan su comportamiento. Ejemplos de relaciones serían:

Subclase-de: Concepto1 x Concepto2
Conectado a: Componente1 x Componente2

- » **Funciones.** Algunas ontologías permiten expresar de alguna forma conocimiento procedimental con el uso de funciones del tipo $F: C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$. Ejemplos de funciones serían:

Madre-de: Persona -> Mujeres
Precio coche usado: Modelo x Año x Kilómetros -> Precio

» **Instancias.** Serían los distintos elementos de la ontología:

- **Categorías.** Incluyen objetos con propiedades comunes dispuestas en taxonomías jerárquicas. Gran parte del razonamiento se da en el ámbito de las categorías. Se puede inferir la categoría de un objeto, sobre la base de sus propiedades, y posteriormente utilizar información de la categoría para hacer predicciones sobre el objeto.

Las categorías van a permitir organizar y simplificar el conocimiento por medio de la herencia. Se pueden tener clases, subclases, clases de clases, etc. Además se pueden tener categorías disjuntas, descomposiciones exhaustivas o particiones.

- **Medidas.** Relaciona objetos a cantidades de tipos particulares (por ejemplo, edad). Las medidas cuantitativas son, en general, fáciles de representar. Otras medidas no tienen una escala de valores única (por ejemplo, la belleza). El aspecto más importante de una medida no es su valor numérico en particular sino el hecho de que pueda ordenarse.
- **Objetos compuestos.** Objetos que pertenecen a categorías por su estructura constitutiva (por ejemplo, las casas están compuestas de paredes, puertas, ventanas...). Se puede tener jerarquías de tipo **partes-de**. Además, se pueden tener objetos compuestos sin estructuras (por ejemplo, una bolsa de caramelos).
- **Tiempo, espacio y cambio.** Para permitir acciones y eventos con diferentes duraciones y que puedan ocurrir simultáneamente.
- **Eventos y procesos.** Eventos individuales que se dan en un tiempo y lugar particular. Los procesos son eventos continuos y homogéneos por naturaleza.
- **Sustancias.** Se pueden distinguir en sustancias temporales y espaciales (por ejemplo, mantequilla). Existen propiedades intrínsecas que son de la sustancia del objeto más que del objeto mismo (color, temperatura a la que se derrite, contenido de grasa, etc.) y propiedades extrínsecas (peso, forma, etc.).

» **Axiomas.** Algunas ontologías permiten expresar conocimiento inferencial a través de los axiomas, es decir, sentencias que siempre son ciertas.

A la hora de su visualización, la forma más común de representarlas es a través de redes semánticas, tal y como se ilustra en la Figura 2, o bien, más habitual, a través de diagramas de clases UML, como se muestra en la Figura 3.

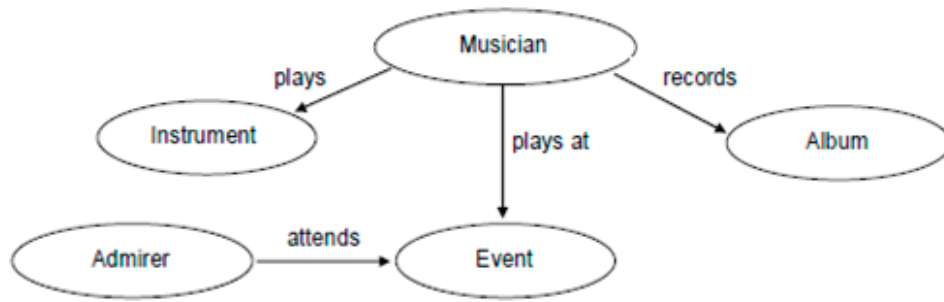


Figura 2. Ejemplo de ontología representada a través de una red semántica (Gašević, Kaviani y Devedžić, 2006).

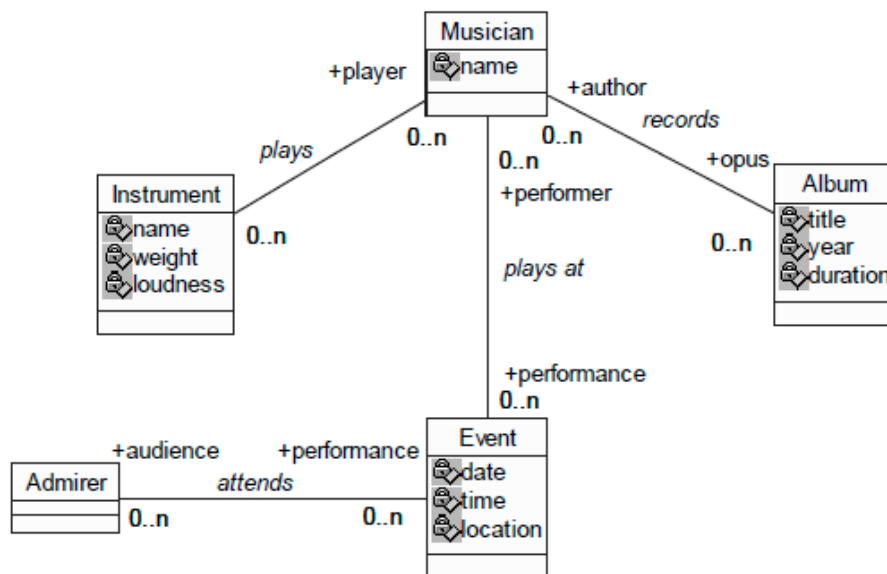


Figura 3. Ejemplo de ontología representada a través de un diagrama de clases UML (Gašević, Kaviani y Devedžić, 2006).

8.3. Ingeniería de la ontología

La **ingeniería de la ontología** (del inglés *Ontological Engineering*, OE) suele ser vista como la panacea a la hora de modelar conocimiento al ser considerada como una disciplina formal que evita las ambigüedades conceptuales, está orientada a la reutilización y a la estandarización, y sirve de base para sistemas más complejos de razonamiento automático (Chandrasekaran, Josephson y Benjamins, 1999) (Gruber, 1991).

La ingeniería de la ontología comprende un conjunto de principios de diseño, procesos y actividades de desarrollo, tecnologías y metodologías que favorecen la creación de

ontologías y su uso a través de todo el ciclo de vida de la ontología: diseño, implementación, evaluación, validación, mantenimiento, despliegue, correspondencia (del inglés *mappings*), integración, intercambio (del inglés *sharing*) y reutilización (Gašević, Kaviani y Devedžić, 2006). La Ingeniería de la Ontología proporciona un fundamento de diseño para el desarrollo base de conocimiento y permite la sistematización de conocimiento basada en el mundo real y en el dominio en estudio (Mizoguchi y Kitamura, 2001).

En este sentido, la ingeniería de la ontología ha demostrado ser útil en distintos ámbitos (KBSI, 1994): (i) llegar a acuerdos o consensos conceptuales; (ii) diseño y desarrollo orientado a objetos; (iii) desarrollo basado en componentes; (iv) diseño de interfaces de usuario; (v) modelado de negocio; (vi) reingeniería de procesos de negocio; y (vii) diseño de esquemas conceptuales. Además, las organizaciones pueden beneficiarse de la ingeniería de la ontología en diversos aspectos (KBSI, 1994):

- » **Mejora en el entendimiento del negocio.** El análisis del negocio en base a ontologías es útil para: (i) identificar el problema que se da en el funcionamiento del negocio (diagnóstico); (ii) identificar de las causas del problema; (iii) identificación de soluciones alternativas para el negocio; (iv) llegar a un consenso entre el equipo implicado en el negocio para tomar una decisión; y (v) compartir y reutilizar el conocimiento en el ámbito del negocio.
- » **Alineamiento entre las tecnologías de la información (TI) y el negocio.** Las ontologías en el ámbito de los sistemas de información para la empresa se pueden utilizar para: (i) desarrollo de los sistemas de información, ya que las ontologías proporcionan un *blueprint* que permite desarrollar sistemas de información más inteligentes y mejor integrados; (ii) desarrollo de sistemas en general, ya que las ontologías se pueden utilizar como modelos de referencia para planificar, coordinar, y monitorizar actividades complejas de desarrollo del producto o proceso del negocio; y (iii) reingeniería de procesos de negocio, ya que las ontologías proporcionan las claves para identificar las áreas que requieren reestructuración organizacional y también sugieren alternativas potenciales de reestructuración que mejorarían la organización.

En el apartado anterior ya se mencionaron distintos tipos de ontologías dependiendo del punto de vista que se tomara, pero de una manera más formal Guarino (1998) clasifica las ontologías en distintas categorías dependiendo de su nivel de especialización:

- » **Ontologías de alto nivel (superior).** Las ontologías de alto nivel describen los conceptos de manera muy general, como sería el tiempo, espacio, asunto, objeto, evento, acción, etc., que son totalmente independientes de un dominio o problema en particular. La Figura 4 muestra un ejemplo de ontología de alto nivel o superior.

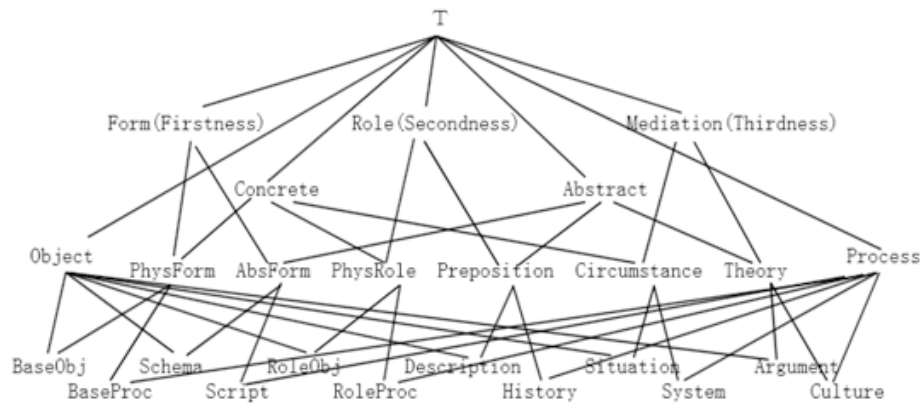


Figura 4. Ontología de alto nivel de SOWA (Sowa, 1995).

- » **Ontologías de domino.** Las ontologías de dominio describen el vocabulario relacionado con un dominio genérico, como por ejemplo, la gestión de servicios de TI, especializando los términos introducidos en las ontologías de alto nivel.
- » **Ontologías de tareas.** Las ontologías de tareas describen una actividad o tarea genérica, como por ejemplo, la monitorización o la medición, especializando los términos introducidos en las ontologías de alto nivel.
- » **Ontologías de aplicación.** Las ontologías de aplicación describen conceptos dependiendo de un dominio o tarea concreta, que suelen ser especializaciones de ambas ontologías. Estos conceptos se suelen corresponder con los roles que toman las entidades del dominio a la hora de llevar a cabo una actividad determinada.

Desde el ámbito de la ingeniería de la ontología, también resulta muy común clasificar las ontologías bajo los tipos «*lightweight*» y «*heavyweight*» (Calero, Ruiz y Piattini, 2006). La distinción entre ambas categorías se debe a una simplificación en el nivel de riqueza de su estructura interna, mientras que las ontologías «*lightweight*» estarán constituidas básicamente por taxonomías, las ontologías «*heavyweight*» serán aquellas donde se modela cierto conocimiento «de una manera más detallada y proporciona mayor número de restricciones en la semántica del dominio en estudio» (Gómez-Pérez, Fernández López y Corcho, 2004). La Figura 5 ilustra la transición continua de una ontología «*lightweight*» a una ontología «*heavyweight*».



Figura 5. Evolución de una ontología desde la categoría «*lightweight*» a la categoría «*heavyweight*».

Fuente: traducida de (Calero, Ruiz y Piattini, 2006).

Siguiendo la Figura 5, se identifican entonces las siguientes categorías para agrupar las ontologías (Calero, Ruiz y Piattini, 2006, 67):

- » **Vocabularios controlados.** Lista finita de términos.
- » **Glosarios.** Listas de términos con sus definiciones proporcionadas en lenguaje natural.
- » **Tesoros.** Se diferencian de las dos categorías anteriores en el sentido que de proporcionan aspectos semánticos a los términos, sinónimos incluidos.
- » **Jerarquías informales.** Jerarquías de términos que no se corresponden a una subclase específica. Por ejemplo, los términos «alquiler de coche» y «hotel» se podrían modelar de manera informal bajo la jerarquía «viaje», ya que se considera que los dos términos forman parte del hecho de viajar.
- » **Jerarquías formales.** En este caso, la relación «es-un/a» se da entre las instancias de una clase y su correspondiente superclase. Su objetivo es hacer uso del concepto de herencia.
- » **Frames.** Ontologías que incluyen clases como propiedades que pueden ser heredadas por otras clases en los niveles inferiores de una taxonomía (jerarquía) formal «es-un/a».

- » **Ontologías con restricciones de valor.** Ontologías que incluyen restricciones que afectan típicamente al tipo de datos de una propiedad, como por ejemplo, la restricción de que un día de mes ha de ser inferior a 32.
- » **Ontologías con restricciones lógicas genéricas.** Las ontologías que pertenecen a este tipo son las ontologías más expresivas, ya que permiten especificar restricciones entre los términos de la ontología utilizando lógica de primer orden.

Desde la aparición de la web semántica, las ontologías se han convertido en el principal recurso de integración y procesamiento de la información online. En consecuencia, han ido apareciendo diversos lenguajes de representación de conocimiento en forma de ontologías, entre los que cabe destacar (Gašević, Kaviani y Devedžić, 2006):

- » KIF (Genesereth y Fikes, 1992). Lenguaje basado en lógica de primer orden.
- » Ontolingua (Gruber, 1992). Lenguaje construido por encima de KIF que incluye representación basada en marcos (del inglés *frame*).
- » RDF (Gandon y Schreiber, 2014). Lenguaje desarrollado por *World Wide Web Consortium* (W3C) como un lenguaje basado en redes semánticas que permite describir recursos Web.
- » RDF Schema (Brickely y Guha, 2014). Lenguaje también desarrollado por W3C que representa una extensión de RDF con primitivas basadas en *frames*. La combinación de RDF y RDF Schema es lo que se conoce como RDF(S).
- » OIL (Fensel et al., 2001). Lenguaje basado en lógica de descripción que incluye primitivas de representación basadas en *frames*.
- » DAML+OIL (Connolly et al., 2001). Lenguaje resultado de la unión de los desarrolladores de *DARPA Agent Markup Language* (DAML) con los desarrolladores de OIL para combinar la expresividad de los dos lenguajes.
- » OWL, también conocido como Web Ontology Language (Smith, Welty, & McGuinness, 2004). Estándar perteneciente a la familia de lenguajes de representación del conocimiento preparado para la web semántica, aunque permite su adopción en otros dominios como podría ser la ingeniería de software. OWL ha alcanzado el estatus de recomendación W3C. Desde un punto de vista técnico, OWL representa una extensión de RDF y RDF Schema que favorece su integración en una gran variedad de aplicaciones haciendo uso de la sintaxis *Extensible Markup Language* (XML) como mecanismo de intercambio.

De todos los lenguajes mencionados anteriormente, OWL constituye en la actualidad el lenguaje de representación de ontologías más utilizado. Debido a su base RDF, las

ontologías OWL se pueden asociar con cualquier tipo de información que se exprese en la web semántica.

Permite, además, la integración de los especificaciones resultantes con una gran variedad de marcos de trabajo, como por ejemplo *electronic business using XML* (ebXML) (OASIS, 2001), y con lenguajes de modelado de negocio como *Business Process Model and Notation* (BPMN) (OMG, 2011), ya que ambos utilizan la sintaxis XML como forma de intercambio de información.

Dentro de OWL, lo más utilizado es un sublenguaje conocido como OWL Description Logics (OWL DL) (Baader et al., 2003). OWL DL favorece la definición de conceptos que describen conocimiento de un dominio. Una de las capacidades más interesantes que proporciona OWL DL es la posibilidad de definir todos los conceptos (clases) de la ontología en base a una serie de condiciones que indican lo que se tiene que dar (condiciones necesarias y suficientes) para que un concepto pertenezca a dicha clase. También se pueden definir nuevas clases a través de la especificación de restricciones basadas en otras clases.

De esta forma, un motor de inferencias puede ejecutar la ontología y calcular la nueva jerarquía de clases inferida a través de todas las condiciones y propiedades definidas en la ontología, así como marcar las clases inconsistentes (por ejemplo, un razonador podría comprobar si una clase definida como subclase de otra, cumple los requisitos para ser realmente subclase o no).

Es importante también indicar que OWL DL está basado en lo que se conoce como *Open World Assumption* (OWA), es decir, la ausencia de una pieza de conocimiento no significa que esa pieza de conocimiento sea falsa, como ocurriría con el enfoque *Closed World Assumption* (CWA), sino que en el enfoque OWA todas las conclusiones (negativas) que se extraen de la base de conocimiento estarán basadas en información que se ha definido de manera explícita en la base de conocimiento (Knorr , Alferes y Hitzler, 2011).

Por ejemplo, con el enfoque OWA, el hecho de que un paciente, por ejemplo, no cumpla las condiciones que se han definido en la ontología para tener un determinado de cáncer, no quiere decir que no tiene cáncer, como ocurriría si se adopta el enfoque CWA, a no ser que en la ontología OWA se indique de manera explícita una conclusión negativa, a través

de la una prueba de laboratorio por ejemplo, que se tiene que dar para que el paciente no tenga dicho tipo de cáncer.

Las ontologías OWL están compuestas de: (i) clases, como representación de conjuntos de individuos; (ii) individuos, que serían las instancias de las clases, es decir, objetos del dominio; y (iii) propiedades en forma de relaciones binarias entre individuos. A través de OWL se pueden definir en la ontología cardinalidades, tipos de datos, relaciones entre clases y razonamiento.

Algunos razonadores, como por ejemplo [Pellet](#), se pueden utilizar para inferir datos adicionales sobre el conocimiento que ha sido establecido de manera explícita en las ontologías OWL. El razonamiento en OWL se puede realizar a nivel de clase, propiedad o instancia, y ejemplos de razonamiento incluirían pertenencia a una clase, equivalencia de clases, consistencia, clasificación de la información, obtención de propiedades adicionales a través de la transitividad, equivalencia, etc.

Una especificación relacionada con OWL es Semantic Web Rule Language (SWRL) (Horrocks et al., 2004), que está basado en [RuleML](#). SWRL proporciona una extensión a OWL que permite definir reglas basadas en lógica, lo que va a permitir dotar a la ontología con mayor expresividad.

Las reglas en conjunción con los datos introducidos en la base de conocimiento se ejecutan como entradas en el motor de reglas infiriendo nuevos datos en la salida de la ejecución. Además, este conocimiento añadido se puede encadenar a otra regla para inferir más conocimiento. Una combinación de reglas y ontologías son las que van a permitir hacer una combinación de los enfoques OWA y CWA explicados anteriormente, según convenga.

Por último, también merece la pena mencionar el lenguaje Semantic Query-Enhanced Web Rule Language ([SQWRL](#)), ya que va a permitir realizar consultas sobre la ontología OWL para poder extraer información (O'Connor y Das, 2009). SQWRL está basado en los fundamentos semánticos de SWRL e incluye un conjunto de operadores que permiten incluir disyunciones, negaciones, contadores... a la hora de definir las consultas que extraerán la información de la base de conocimiento.

8.4. Aplicación de las ontologías en la ingeniería de *software*

Independientemente del ámbito en el que se aplica, ya sea la ingeniería de *software* o cualquier otra disciplina, las ontologías aportan las siguientes capacidades:

- » **Facilitan la comprensión de la estructura del conocimiento.** Durante el análisis de la ontología, los conceptos del dominio y sus relaciones se pueden definir de tal manera que faciliten la especificación de la naturaleza de los conceptos que se están utilizando con respecto al cuerpo de conocimiento que va a ser desarrollado (Chandrasekaran, Josephson y Benjamins, 1998).
- » **Reducen la ambigüedad terminológica y conceptual.** El análisis de la ontología proporciona un marco de trabajo (del inglés *framework*) para conseguir una unificación o consenso entre las diferentes personas o partes implicadas que tendrán diferentes necesidades o puntos de vista, dependiendo de su contexto en particular (Uschold y Gruninger, 1996).
- » **Permiten compartir el conocimiento.** A través de un adecuado análisis de la ontología será posible obtener un conjunto de conceptualizaciones asociadas a un dominio específico, y el conjunto de términos que le da soporte. Con una sintaxis adecuada, todas las conceptualizaciones y sus relaciones que se encuentran definidas y codificadas en una ontología, se podrán compartir con cualquier agente, ya sea persona o sistema, que tenga unas necesidades similares para el mismo dominio (Lassila y McGuinness, 2001).

Con el objetivo de aplicación en la ingeniería de *software* se encuentran muchos estudios en la literatura que categorizan los posibles usos de las ontologías dentro de este campo. Entre algunos usos se encuentran, por ejemplo, las ontologías de dominio, que resultan especialmente útiles para el desarrollo de *software* reutilizable y de calidad, ya que proporcionan una terminología formal que no da lugar a contradicciones ni a ambigüedades.

Además, gracias a las ontologías la fase de elicitación y modelado de los requisitos se ve favorecida en gran medida, ya que permite que la definición de los requisitos sea lo más exacta posible: los expertos del dominio donde va a operar el *software* pueden aportar mucho más conocimiento del dominio gracias al uso de las ontologías con las que estarán más familiarizados (Calero, Ruiz y Piattini, 2006).

Pisanelli, Gangemi y Steve (2002) resumen las características más importantes que presentan las ontologías de cara a su aplicación en la ingeniería de software:

- » Una taxonomía y semántica explícita.
- » Una clara conexión entre los conceptos, sus relaciones y teorías genéricas.
- » Un contexto formal donde no tiene cabida la polisemia.
- » Modularidad del contexto.
- » Axiomatización mínima para identificar las diferencias entre conceptos que son similares.
- » Una adecuada elección de los nombres.
- » Amplia documentación.

Por su parte, Calero, Ruiz y Piattini (2006, 72) listan los objetivos que pueden tener las ontologías cuando se trata de su aplicación en el área de la ingeniería de *software*:

- » **Especificación.** El rol que las ontologías juegan en la especificación, depende del nivel de formalización y automatización que presente la metodología de diseño del sistema.
- » **Confianza.** Las ontologías informales podrían mejorar la confianza en el sistema, ya que sirven como comprobación manual del diseño. A su vez, las ontologías formales permiten la comprobación semiautomática de la consistencia de un sistema *software* con respecto a la especificación declarativa que lleva asociada cada ontología concreta.
- » **Reutilización.** Para que resulten lo más útiles posible, las ontologías deberían ser capaces de dar soporte a la importación y exportación de módulos (partes de la ontología). Si se determinan las clases de dominio y las tareas que se han de dar dentro de los subdominios, las ontologías puede proporcionar un *framework* que ayude a determinar los aspectos de la ontología que se pueden reutilizar entre los distintos dominios y tareas. El objetivo, en este caso, sería la obtención de bibliotecas de ontologías que sean reutilizables y adaptables a diferentes problemas y entornos.
- » **Búsqueda.** Las ontologías se podrían utilizar como metadatos que servirían como índices en repositorios de información.
- » **Fiabilidad.** Si se consigue una representación formal del conocimiento, la comprobación de la consistencia podría ser (semi)automática.
- » **Mantenimiento.** Las ontologías podrían mejorar la documentación asociada al proyecto *software* y reducir los costes de mantenimiento.
- » **Adquisición de conocimiento.** Si se utilizan como punto de partida y como guía para la adquisición de conocimiento, las ontologías podrían mejorar el desarrollo de

sistemas basados en conocimiento, así como la velocidad y la fiabilidad de las aplicaciones.

Por otro lado, a la hora de la aplicación de las ontologías en el ámbito de la ingeniería de *software*, modelos conceptuales de bases de datos y ontologías suelen confundirse, ya que ambas presentan una funcionalidad similar. Sin embargo, tal y como se destaca en (Meersman, 2000) y en (Guarino y Schneider, 2002) es importante conocer las diferencias para no utilizarlos como conceptos equivalentes:

- » Los lenguajes que se utilizan para representar ontologías, como por ejemplo, OWL, tiene mayor riqueza sintáctica y semántica que los lenguajes de definición de bases de datos, como podría ser por ejemplo, SQL.
- » El conocimiento que se proporciona a través de una ontología se representa en forma de información semiestructurada, con textos en lenguaje natural, mientras que las bases de datos proporcionan el conocimiento en base a información estructurada en forma de tablas, clases de objetos, etc.
- » Una ontología debe representar una conceptualización consensuada y compartida, ya que contiene información enfocada a la reutilización y al intercambio. Sin embargo, los identificadores de un esquema de base de datos se utilizarán para un sistema concreto donde no existe la necesidad de alcanzar los acuerdos conceptuales que requiere una ontología para ser considerada válida.
- » Una ontología proporciona la teoría de un dominio, mientras que una base de datos representa la estructura de un contenedor de datos.

En la misma línea, en el ámbito de la ingeniería de *software*, en concreto cuando se adopta el enfoque de desarrollo de *software* dirigido por modelos, suele surgir la duda de si una ontología y un metamodelo serían equivalentes. Sin embargo, una vez más, hay autores que defienden que las características y objetivos en cada caso son distintos.

Para comprender las diferencias, Bertrand y Bézivin (2000) analizaron la relación que se da entre las ontologías de bajo nivel y los metamodelos llegando a la conclusión de que mientras los metamodelos intentan mejorar el rigor de los modelos que son similares pero con distintos puntos de vista, las ontologías hacen lo mismo pero aplicado a modelos de conocimiento.

Devedzic (2002), va más allá e indicó otra diferencia entre ambos conceptos, explicando que sin una ontología, representaciones diferentes de conocimiento para un mismo

dominio serían incompatibles incluso aunque se utilizara el mismo metamodelo para su implementación; es decir, o se da una ontología que establezca la conceptualización compartida del dominio en estudio, o diferentes modelos del sistema serán incompatibles entre sí, por no compartir una base de conocimiento común.

Por su parte, Calero, Ruiz y Piattini (2006) se quedan con la idea de que lo que diferencia a una ontología de un metamodelo es su intención, es decir, mientras que una ontología es prescriptiva y pertenece al dominio del problema, un metamodelo, también es prescriptivo, pero pertenece al dominio de la solución.

En cambio, en algunos casos se puede encontrar que hay autores (incluso, un mismo autor que matiza anteriores conclusiones) que indican que un metamodelo es una forma simple de una ontología, tal y como se muestra en la Figura 6. Además, ¿por qué no se puede utilizar un metamodelo para modelar también el dominio del problema? Con lo cual, dado que hay un acuerdo general, la confusión y el debate están servidos.

Ontologies and meta-models

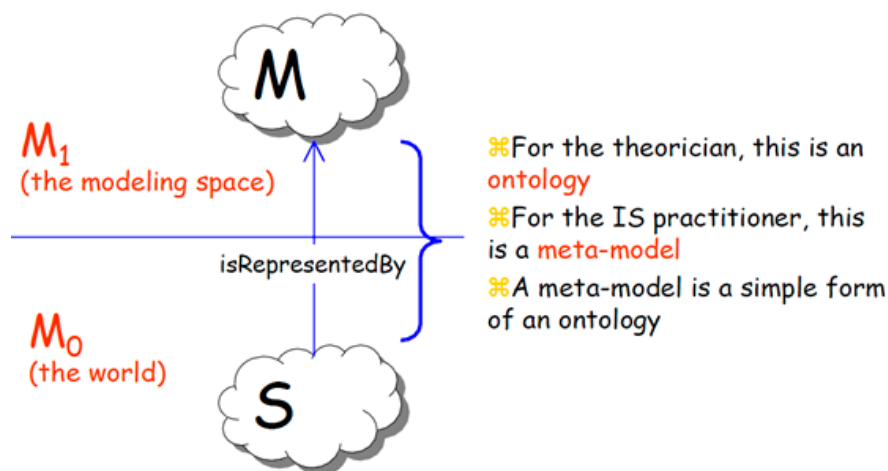


Figura 6. Metamodelo vs Ontología. Fuente: Transparencia de la presentación realizada por Jean Bézivin (Bézivin, 2001)

De hecho también se puede encontrar en la literatura, extensiones ontológicas a metamodelos existentes, como por ejemplo, la propuesta de ampliar *MetaObject Facility* (MOF) (OMG, 2014) con una extensión ontológica que serviría como base en la generación automática de código (Gitzel, Ott y Schader, 2006). La Figura 7 ilustra la extensión propuesta donde los nuevos elementos incluidos en MOF y basados en una ontología aparecen con el fondo blanco.

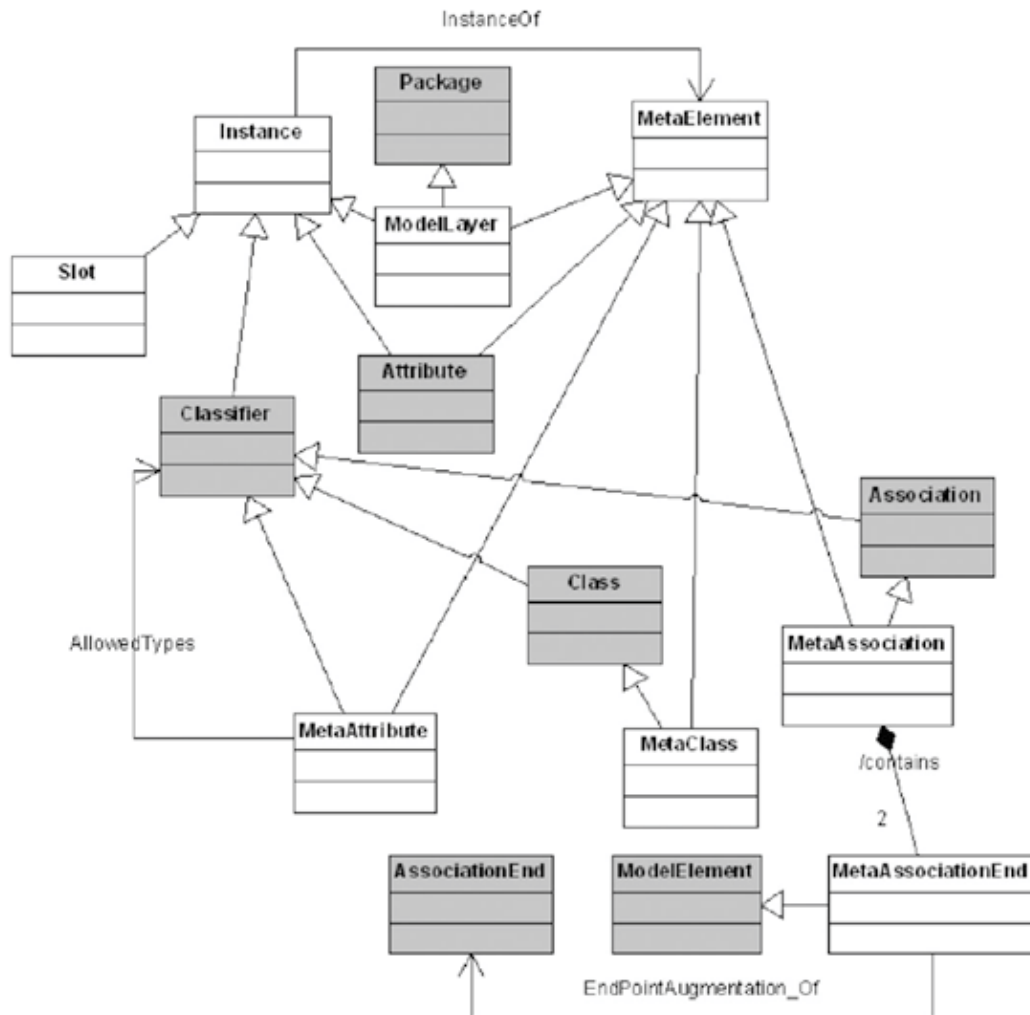


Figura 7. Extensión ontológica a MOF para generación de código (Gitzel, Ott & Schader, 2006).

Existen otras propuestas del uso de ontologías para la implementación de sistemas de información, como la implementada en (Valiente, 2011), donde se intenta formalizar el conocimiento asociado a los sistemas de gestión de servicios de TI en base a ontologías que recogen las buenas prácticas consideradas por *IT Infrastructure Library* (ITIL), de cara a modelar la semántica de las actividades que definen los procesos de gestión de servicios de TI en forma de flujos de trabajo. De esta manera se podrían generar de manera automática modelos de requisitos de alto nivel para implementar sistemas de información que se necesitan para dar soporte a dichos procesos. La Figura 8 ilustra la arquitectura propuesta.

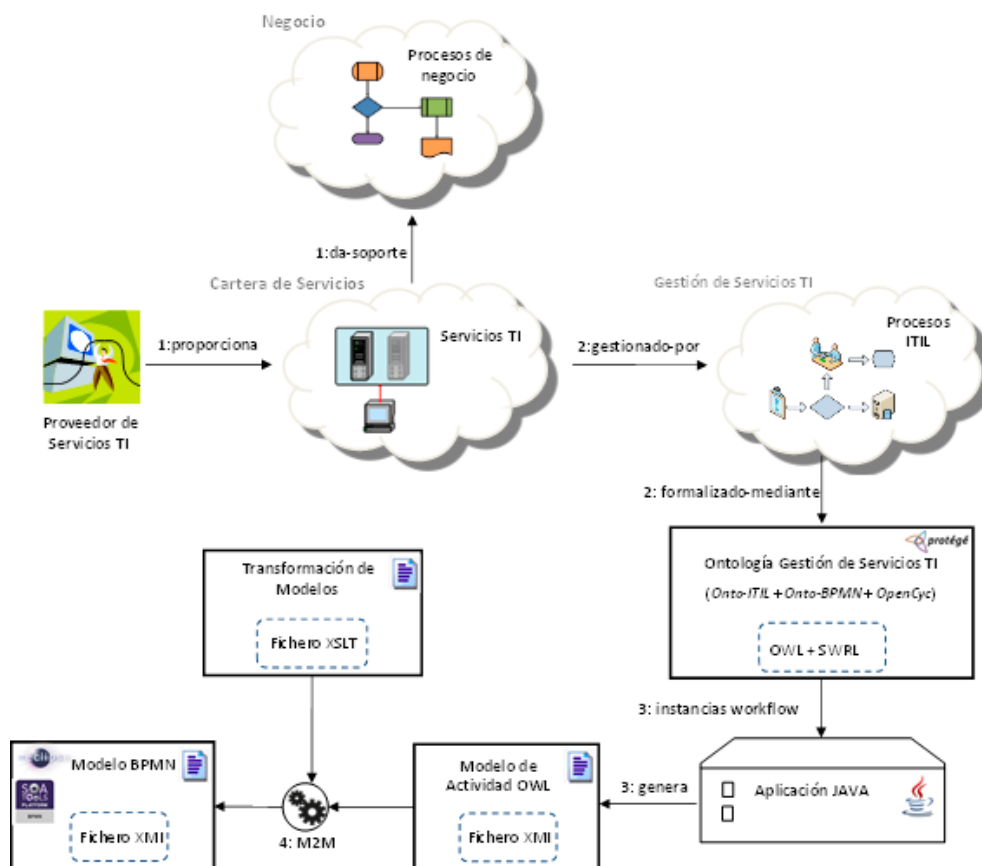


Figura 8. Arquitectura de Onto-ITIL (Valiente, 2011).

Por último, con un enfoque orientado a la gestión de proyectos, en (Ruiz Bertol y Dolado, 2008) se presenta la ontología de dominio *Project Management Ontology* (PMO), que define la estructura común que se da en toda gestión de proyectos y la información que se necesita para poder cumplimentar la ontología. La Figura 10 ilustra el contenido de la ontología PM-Core, la considerada principal de PMO, ya que el resto de ontologías que forman parte de la ontología global PMO se encuentran relacionadas con PM-Core.

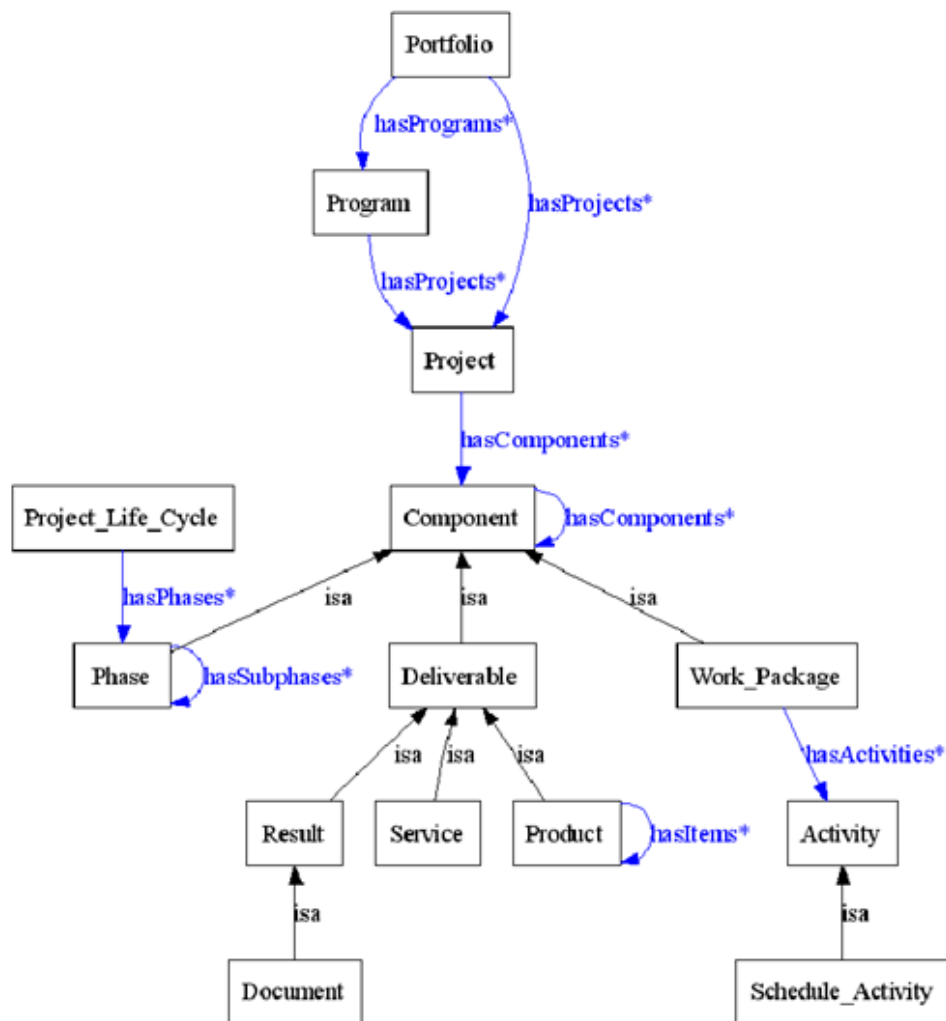


Figura 10. Vista simplificada de la ontología PM-Core (Ruiz Bertol y Dolado, 2008)

8.5. Referencias bibliográficas

Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D. y Patel-Schneider, P.F. (2003). *Description Logic Handbook*. Cambridge University Press.

Bézivin, J. (2001). Models Everywhere. TOOLS USA 2001: Software Technologies for the Age of the Internet, 39th International Conference & Exhibition, Santa Barbara, CA, USA. Recuperado de https://www.researchgate.net/profile/Jean_Bezivin/publications?sorting=newest&page=5

Bertrand, T. y Bézivin, J. (2000). Ontological Support for Business Process Improvement. *Systems Modeling for Business Process Improvement*. Artech House Publishers, London, 313–331.

Brickley, D. y Guha, R.V. (2014). RDF Schema 1.1. W3C *Recommendation*. Recuperado de <http://www.w3.org/TR/rdf-schema/>

Calero, C., Ruiz, F. y Piattini, M. (2006). *Ontologies in Software Engineering and Software Technology*. Springer.

Chandrasekaran, B., Josephson, J.R. y Benjamins, V. (1998). Ontology of Tasks and Methods. *Actas de KAW'98, Banff*, Alberta, Canada.

Chandrasekaran, B., Josephson J. y Benjamins, R. (1999). What are ontologies, and why do we need them. *Journal of IEEE Intelligent Systems*, 14(1), 20-26.

Connolly, D., van Harmelen, F., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F. y Stein, L.A. (2001). DAML +OIL (March 2001) Reference Description. Available: <http://www.w3.org/TR/daml+oil-reference>

Devedzić, V. (2002). Understanding Ontological Engineering. *Communications of the ACM*, 45(4), 136–144.

Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D.L., y Patel-Schneider, P.F. (2001). OIL: An ontology infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 16(2), 38–45.

Gandon, F. y Schreiber, G. (2014). RDF 1.1 XML Syntax. *W3C Recommendation*. Recuperado de <http://www.w3.org/TR/rdf-syntax-grammar/>

García, F., Ruiz, F., Calero, C., Bertoa, M.F., Vallecillo, A., Mora, B. y Piattini, M. (2009). Effective use of ontologies in software measurement. *The Knowledge Engineering Review*, 24(1), 23-40.

Gašević, D., Kaviani, N. & Devedžić, V. (2006). *Model Driven Architecture and Ontology Development*. Springer-Verlag Berlin Heidelberg.

Genesereth, M.R. & Fikes, R.E. (1992). Knowledge Interchange Format, Version 3.0. Reference Manual. *Interlingua Working Group of the DARPA Knowledge Sharing Effort*. Recuperado de: <http://www-ksl.stanford.edu/knowledge-sharing/papers/kif.ps>

Gitzel, R., Ott, I. y Schader, M. (2006). Ontological Extension to the MOF Metamodel as a Basis for Code Generation. *The Computer Journal*, 50(1), 93- 115.

Gruber, T.R. (1991). The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases. *Actas de Second International Conference on Principles of Knowledge Representation and Reasoning*, 601-602. Morgan Kaufmann, Cambridge.

Gruber, T.R. (1992). Ontolingua: A Mechanism to Support Portable Ontologies. *Knowledge Systems Laboratory, Stanford University, CA*.

Gruber, T.R. (1995). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies*, Vol. 43, No. 5-6, pp. 907-928.

Guarino, N. (1998). Formal Ontology in Information Systems. *Actas de FOIS'98, Trento, Italy*. IOS Press, Amsterdam.

Guarino, N. y Schneider, L. (2002). Ontology-Driven Conceptual Modelling: Advanced Concepts. *ER 2002. Pre-Conference Tutorials*.

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B. y Dean, M. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *W3C Member Submission*. Recuperado de <http://www.w3.org/Submission/SWRL/>

KBSI. (1994). *IDEF5 Method Report. Information Integration for Concurrent Engineering (IICE) project, F33615-90-C-0012.* Recuperado de <http://www.idef.com/pdf/Idef5.pdf>

Knorr , M., Alferes, J.J. y Hitzler, P. (2011). Local Closed World Reasoning with Description Logics under the Well-Founded Semantics. *Journal of Artificial Intelligence*. Recuperado de <http://knoesis.wright.edu/faculty/pascal/resources/publications/mknftheo.pdf>

Lassila, O. y McGuinness, D. (2001). The Role of Frame-Based Representation on the Semantic Web. *KSL Technical Report No. KSL-01-02, Jan-2001.*

Lenat, D.B. (1995). Cyc: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 38(11), 33-38.

Meersman, R.A. (2000). The Use of Lexicons and Other Computer-Linguistic Tools in Semantics Design and Cooperation of Database Systems. *Actas de CODAS Conference*, Springer-Verlag, Berlin.

Mizoguchi, R. y Kitamura, Y. (2001). Knowledge systematization through ontology engineering – a key technology for successful intelligent systems. *Invited paper, Pacific-Asian Conference on Intelligent Systems, Seoul.*

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T. y Swartout, W.R. Enabling Technology for Knowledge Sharing. *AI Magazine. Winter 1991*, 36-56.

OASIS. (2001). Business process and business information analysis overview. version 1.0 (ebXML). Recuperado de <http://www.ebxml.org/specs/bpOVER.pdf>

O'Connor, M.J. y Das, A. (2009). SQWRL: a Query Language for OWL. *OWL: Experiences and Directions (OWLED 2009), Fifth International Workshop, Chantilly, VA.*

Olivé, A. (2007). *Conceptual Modeling of Information Systems*. Springer-Verlag Berlin Heidelberg.

OMG (2011). Business Process Model and Notation (BPMN). Version 2.0., *document formal/2011-01-03*. Recuperado de <http://www.omg.org/spec/BPMN/2.0/PDF/>

OMG (2014). Meta Object Facility (MOF) Core Specification, Version 2.4.2. *Document formal/2014-04-03*. Recuperado de <http://www.omg.org/spec/MOF/2.4.2/PDF/>

Pisanelli, D.M., Gangemi, A. & Steve, G. (2002). Ontologies and Information Systems: the Marriage of the Century? *Actas de LYEE Workshop*, Paris.

Rodríguez, H. (2000). Adquisición automática y uso de taxonomías de amplia cobertura. *Workshop on Automatic Acquisition of Linguistic Knowledge*.

Ruiz Bertol, F.J. y Dolado, J. (2008). Una ontología para la gestión del conocimiento de proyectos software. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, 4(1).

Smith, M.K., Welty, C., y McGuinness, D.L. (2004). OWL 2 Web Ontology Language Primer (Second Edition). *W3C Recommendation*. Recuperado de <http://www.w3.org/TR/2012/REC-owl2-primer-20121211/>

Sowa, J. (1995). Distinction, combination, and constraints. *Actas de IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*.

Swartout, B., Patil, R., Knight, K. y Russ, T. (1997). Toward Distributed Use of Large-Scale Ontologies. *Ontological Engineering. AAAI-97 Spring Symposium Series. 1997*, 138-148.

Uschold, M. y Gruninger, M. (1996). Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, 11(2), 93-15.

Valiente, M.-C. (2011). Improving IT Service Management using an Ontology-Based and Model-Driven Approach. *Tesis Doctoral*.

Lo + recomendado

No dejes de leer...

La web semántica: tecnologías y aplicaciones

Castells, P. y Saiz, F. (2006). *La Web Semántica: tecnologías y aplicaciones*.
Apuntes Escuela Politécnica Superior, Universidad Autónoma de Madrid.

Transparencias correspondientes a la asignatura «La web semántica» en la que se sintetiza muy bien el significado de web semántica y el papel de las ontologías.

Accede al artículo desde el aula virtual o a través de la siguiente dirección web:
<http://arantxa.ii.uam.es/~dcamacho/SemanticWeb/material/o-presentacion-o6o7.pdf>

Una ontología para la gestión del conocimiento de proyectos *software*

Ruiz Bertol, F.J. y Dolado, J. (2008). Una ontología para la gestión del conocimiento de proyectos software. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, 4(1).

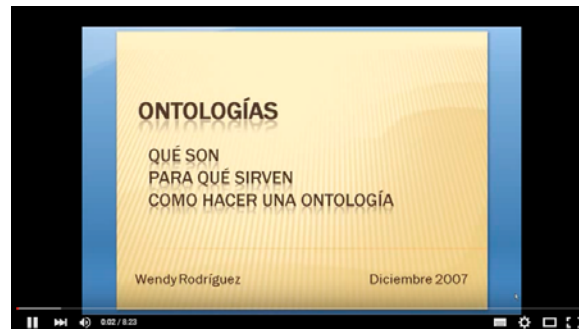
En este artículo se presenta una ontología de dominio denominada *Project Management Ontology (PMO)* que captura toda la estructura asociada a la gestión de proyectos así como la información asociada para cumplimentar la ontología.

Accede al artículo desde el aula virtual o a través de la siguiente dirección web:
<http://www.redalyc.org/articulo.oa?id=92240103>

No dejes de ver...

Ontologías usando *protégé* 1-2

Aunque ya un poco antiguo, este vídeo resulta útil para ver cómo se crean ontologías OWL con la herramienta *Protégé*, un entorno de desarrollo de ontologías muy difundido. Este vídeo constituye la primera parte de una serie de dos vídeos.

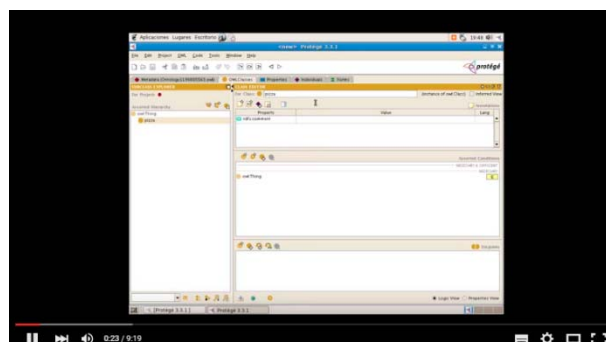


Accede al vídeo desde el aula virtual o a través de la siguiente dirección web:

https://www.youtube.com/watch?v=IWCBNYo_F3M

Ontologías usando *protégé* 2-2

Este vídeo constituye la segunda parte del vídeo anterior acerca de la creación de ontologías OWL con la herramienta *Protégé*.



Accede al vídeo desde el aula virtual o a través de la siguiente dirección web:

<https://www.youtube.com/watch?v=cqGy1bejCCK>

+ Información

A fondo

Ontologies in software engineering and software technology

Calero, C., Ruiz, F. y Piattini, M. (2006). *Ontologies in Software Engineering and Software Technology*. Springer.

En este libro se recoge un conjunto de artículos relacionados con ontologías aplicadas a la ingeniería de *software*.

Ontology design patterns for semantic web content

Gangemi, A. (2005). Ontology Design Patterns for Semantic Web Content. Y. *Gil et al. (Eds.): ISWC 2005, LNCS 3729*, 262 – 276.

En este artículo el autor presenta un marco de trabajo (del inglés *framework*) para utilizar patrones que faciliten o mejoren las técnicas utilizadas durante el ciclo de vida de una ontología.

Disponible en el aula virtual bajo licencia CEDRO

Enlaces relacionados

KEOD

El congreso anual conocido como *International Conference on Knowledge Engineering and Ontology Development* (KEOD) constituye un punto de referencia en la investigación y aplicación de sistemas basados en el conocimiento y en el desarrollo de ontologías en distintos ámbitos de la sociedad. KEOD forma parte del *International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*.



Accede a la página desde el aula virtual o a través de la siguiente dirección web:

<http://www.keod.ic3k.org/>

Ontology Design Patterns . org (ODP)

OntologyDesignPatterns.org es un portal de web semántica dedicado a patrones de diseño de ontologías.

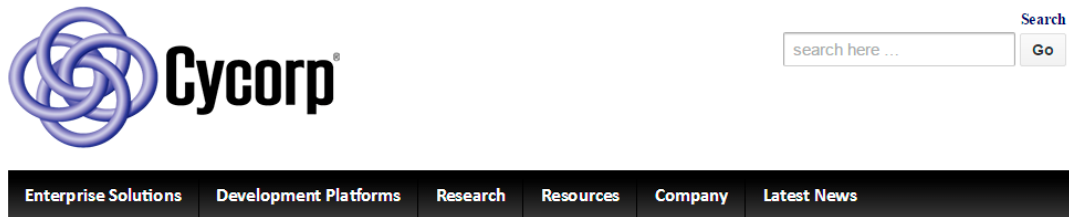


Accede a la página desde el aula virtual o a través de la siguiente dirección web:

<http://ontologydesignpatterns.org/>

OpenCyc

OpenCyc representa la versión pública de la ontología de alto nivel Cyc, la base de conocimiento más completa y extensa del mundo, que se puede utilizar como base para definir nuestras propias ontologías favoreciendo la integración con otras ontologías así como su reutilización.



Accede a la página desde el aula virtual o a través de la siguiente dirección web:

<http://www.cyc.com/platform/opencyc/>

Protégé

Portal de descarga y documentación de la herramienta *Protégé*, un entorno para el desarrollo de ontologías muy difundido.



Accede a la página desde el aula virtual o a través de la siguiente dirección web:

<http://protege.stanford.edu/>

Bibliografía

Gašević, D., Kaviani, N. y Devedžić, V. (2006). *Model Driven Architecture and Ontology Development*. Springer-Verlag Berlin Heidelberg.

Gruber, T.R. (1995). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies*, 43(5-6), 907-928.

Olivé, A. (2007). *Conceptual Modeling of Information Systems*. Springer-Verlag Berlin Heidelberg.

Test

1. Indicar cuáles de las siguientes afirmaciones es correcta:
 - A. Una ontología no permite realizar suposiciones explícitas sobre el dominio que representa.
 - B. Una ontología es formal y representa la estructura de una parcela del mundo real.
 - C. Una ontología no se puede reutilizar, se construye para un dominio concreto.
 - D. Las ontologías representan una conceptualización compartida de un dominio.

2. Las ontologías estarán formadas por los siguientes componentes:
 - A. Conceptos, relaciones, funciones, instancias y axiomas.
 - B. Conceptos, relaciones, funciones, instancias y clases.
 - C. Conceptos, relaciones, funciones, objetos y reglas.
 - D. Clases, relaciones, funciones e instancias.

3. ¿Cuáles de las siguientes opciones representan un lenguaje para la representación de conocimiento?
 - A. Cyc.
 - B. OWL.
 - C. RDF.
 - D. Wordnet.

4. ¿Cuáles de las siguientes opciones serían ejemplos de instancias?
 - A. Axioma.
 - B. Función.
 - C. Sustancia.
 - D. Evento.

5. ¿Cuáles son las representaciones gráficas más comunes para representar ontologías?
 - A. Diagrama de clases UML.
 - B. Diagrama de objetos UML.
 - C. Diagramas de flujo de datos (DFD).
 - D. Las ontologías no tienen representación gráfica.

6. ¿Desde la perspectiva de especialización cuáles de las siguientes opciones representa un tipo de ontología?

- A. Ontología léxica.
- B. Ontología conceptual.
- C. Ontología superior.
- D. Ontología de dominio.

7. ¿Qué es un tesaurus?

- A. Lista finita de términos.
- B. Lista finita de términos con aspectos semánticos, sinónimos incluidos.
- C. Lista de términos de tamaño indeterminado.
- D. Listas de términos con sus definiciones proporcionadas en lenguaje natural.

8. ¿Qué es una jerarquía formal?

- A. Ontología donde la relación «es-un/a» se da entre las instancias de una clase y su correspondiente superclase.
- B. Ontología donde la relación «es-un/a» no se corresponde a una subclase específica.
- C. Ontologías que incluyen restricciones que afectan típicamente al tipo de datos de una propiedad de una clase y su superclase.
- D. Ontologías que permiten especificar restricciones entre los términos en distintos niveles de la jerarquía de clases utilizando lógica de primer orden.

9. De cara a su aplicación en la ingeniería de *software*, las ontologías presentan las siguientes características.

- A. Una clara conexión entre los conceptos, sus relaciones y los componentes *software* que los implementan.
- B. Una clara conexión entre los conceptos, sus relaciones y teorías genéricas.
- C. Escasa documentación, ya que la ontología no se representa en lenguaje natural.
- D. Una taxonomía y semántica implícita.

10. ¿En qué se diferencia un modelo conceptual de base de datos de una ontología?

- A. No hay diferencia, son equivalentes.
- B. La ontología presenta más elementos semánticos pero no sintácticos.
- C. En el modelo conceptual los términos están consensuados, mientras que en la ontología no.
- D. Las ontologías presentan mayor riqueza sintáctica y semántica.