

Tecnologías para la Web Semántica

Resumen

Parcial 2

Darién Julián Ramírez

Índice

1. Ingeniería Ontológica I: Methontology	2
1.1. Proceso de Desarrollo de Ontologías	2
1.1.1. Actividades de Gestión de Ontologías	2
1.1.2. Actividades Orientadas al Desarrollo de Ontologías	2
1.1.3. Actividades de Soporte de Ontologías	3
1.2. Methontology	3
1.2.1. Especificación	3
1.2.2. Conceptualización	4
1.2.3. Adquisición de Conocimiento	6
1.2.4. Integración	7
1.2.5. Implementación	7
1.2.6. Evaluación	7
1.2.7. Documentación	7
2. Ingeniería Ontológica II: Lenguajes de consultas	7
2.1. RDQL	7
2.1.1. Ejemplo	8
2.2. Implementación	8
2.2.1. Jena	8
2.2.2. Jena 1	9
2.2.3. Jena 2	9
2.3. SPARQL	9
3. Ingeniería Ontológica III: Evaluación	10

1. Ingeniería Ontológica I: Methontology

Metodología: serie comprensiva, integrada de técnicas o métodos para crear una teoría general de sistemas de cómo debería realizarse una clase de trabajo de pensamiento intensivo.

Método: procedimiento general.

Tarea: asignación de trabajo bien definido por uno o mas miembros de un proyecto.

Técnica: aplicación específica de un método y la manera en que se ejecuta.

1.1. Proceso de Desarrollo de Ontologías

Las siguientes tres actividades se llevan a cabo cuando se construyen ontologías.

1.1.1. Actividades de Gestión de Ontologías

I. Cronograma:

- Identifica tareas a realizar y su orden.
- Especifica horarios y recursos necesarios.

II. Control:

III. Aseguramiento de Calidad:

1.1.2. Actividades Orientadas al Desarrollo de Ontologías

I. Pre-desarrollo:

- Estudio de entorno (plataformas, aplicaciones, etc.)
- Estudio de factibilidad.

II. Desarrollo:

- Especificación de razones de construcción, usos de la ontología, usuarios finales.
- Conceptualización.
- Formalización.
- Implementación.

III. Post-desarrollo:

- Mantenimiento.
- Reutilización.

1.1.3. Actividades de Soporte de Ontologías

- Se realizan en forma simultánea con las actividades de desarrollo.
- Incluyen adquisición de conocimiento, evaluación, integración, mezclado, alineación, documentación y configuración.

1.2. Methontology

1.2.1. Especificación

Desarrollar un documento que contenga la meta de la ontología, nivel de granularidad, alcance y propósito. Identificar los términos a representar, sus características y relaciones.

Preguntas de competencia:

- Dado un conjunto de escenarios informales, se identifican un conjunto de preguntas de competencia en lenguaje natural.
- Serán respondidas por la ontología una vez que este expresada en lenguaje formal.
- Juegan el rol de un tipo de especificación de requerimientos con la que la ontología podrá ser evaluada.
- Ejemplo: Dadas las preferencias de un pasajero (viaje cultural, viaje en la montaña, playa, etc.) y algunas restricciones económicas, ¿qué destino es el más apropiado?

Técnicas de adquisición de conocimiento:

- Reglas generales:
 - Aislar al experto de su trabajo por períodos cortos de tiempo.
 - Enfocarse sobre el conocimiento esencial.
 - Recolectar conocimiento de diferentes expertos.
- Resultados:
 - Personas no expertas podrán entender el conocimiento.
 - El conocimiento podrá ser evaluado.
- Técnicas de generación de protocolo:
 - Diferentes tipos de entrevistas no estructuradas, semi-estructuradas y estructuradas.
 - Diferentes técnicas de reporte.
 - Diferentes tipos de técnicas de observación.
- Técnicas de análisis de protocolo:
 - Usadas con transcripciones de entrevistas u otra información textual.
 - Útiles para identificar varios tipos de conocimiento (objetivos, decisiones, relaciones y atributos).
 - Actúan como vínculo entre el uso de técnicas basadas en protocolo y técnicas de modelado de conocimiento.

- Técnicas de generación de jerarquías:
 - Útiles para construir taxonomías u otras estructuras jerárquicas (árboles de decisión).
- Técnicas basadas en matrices:
 - Se basan en construir y rellenar una matriz de dos dimensiones (tabla). Por ejemplo: tabla de conceptos y propiedades (atributos y valores), problemas y soluciones; tareas y recursos.
- Técnicas basadas en diagramas:
 - Generación y uso de mapas conceptuales, redes de transición de estados, diagramas de evento y mapas de procesos.
 - Útiles para capturar el *qué, cómo, donde, quién y por qué* de tareas y eventos.
 - Se ha comprobado empíricamente que la gente comprende muy bien la notación gráfica, mucho mejor que otros formalismos como la lógica de predicado.
- Técnicas de ordenamiento:
 - Utilizadas para capturar la forma en que las personas comparan y ordenan conceptos.
 - Conducen al descubrimiento de conocimiento acerca de las clases, las propiedades y las prioridades.
- Manuales de instrucciones o libros almacenan conocimiento que puede ser extraído sin necesidad de entrevistas.
- Dificultad por parte de algunos expertos, incluso si tienen voluntad de ayudar, de explicar con palabras cómo resuelven un problema, aunque lo sepan resolver perfectamente. Utilizar técnicas alternativas (sin preguntas).
- Técnicas de observación, en las que el ingeniero observa al experto trabajando e intenta entender y duplicar sus métodos de resolver el problema.
- Técnicas intuitivas, en las que el ingeniero intenta actuar como si fuera él el experto e implementar su propio conocimiento sobre el dominio.

Especificación de requerimientos:

El objetivo de la especificación de requerimientos en el desarrollo de la ontologías es establecer el propósito con que se construye la ontología, cuáles van a ser sus usos y usuarios posibles y qué requisitos debe cumplir esa ontología.

1.2.2. Conceptualización

Organizar el conjunto de términos y sus características en una representación intermedia que el desarrollador de la ontología y los expertos puedan entender. En este caso se construye un glosario de términos, diagrama de relaciones binarias, diccionario de conceptos, tablas de atributos instancias, tablas de atributos clases, tablas de axiomas lógicos, tablas de constantes, tablas de instancias.

Modelado conceptual:

- Determina el resto de la construcción de la ontología.
- Tiene como objetivo organizar y estructurar el conocimiento adquirido durante la actividad de adquisición del conocimiento (fuerte relación entre ambas actividades).
- Convierte una vista informal de un dominio en una especificación semi-formal.
- Utiliza un conjunto de representaciones intermedias, tabulares y gráficas.

- IRs facilitan el proceso de transformación entre la percepción de las personas y lenguajes utilizados para la implementación de ontologías.

Glosario de términos:

- Incluye los términos relevantes del dominio: conceptos, instancias, atributos (propiedades).
- Relaciones entre conceptos.
- Descripciones en lenguaje natural.
- Sinónimos - acrónimos (siglas).

Taxonomía:

- Define la jerarquía de conceptos.
- Top-down, bottom-up, middle-out.
- Relaciones: Subclass-off, Disjoint-decomposition, Exhaustive-decomposition, Partition.

Diagrama de relaciones:

- Establece relaciones entre conceptos de una taxonomía.
- Se debe establecer si los dominios y rangos de cada argumento de cada relación delimita exactamente las clases que son apropiadas para esa relación.
- Los errores aparecen cuando dominios y rangos son imprecisos o sobre-especificados.

Diccionario de conceptos:

- Especifica cuales son las propiedades y relaciones que describen cada concepto de la taxonomía.
- Opcionalmente se incluyen: instancias y atributos de clase e instancias.
- Las relaciones especificadas para cada concepto son aquellas cuyo dominio es el concepto.

Detalle de relaciones:

- Describe en detalle todas las relaciones binarias incluidas en el diccionario de conceptos.
- Para cada relación se debe especificar nombre, nombre del concepto fuente y destino, cardinalidad, su relación inversa y su relación matemática.

Atributos de instancias:

- Describe en detalle los atributos de instancia incluidos en el diccionario de conceptos.
- Los atributos de instancia poseen valores que pueden diferir para cada instancia del concepto.
- Se especifica nombre, tipo, unidad de medida, precision y rango de valores (en el caso de valores numéricos), valores por default si existen, cardinalidad mínima y máxima; atributos de instancia, atributos de clase y constantes utilizadas para inferir valores; atributos que pueden ser inferidos utilizando valores del atributo; fórmulas o reglas que permiten inferir valores del atributo y referencias utilizadas para definir el atributo.

Atributos de clase:

- Describe en detalle los atributos de clase incluidos en el diccionario de conceptos.

- Describen conceptos y toman su valor en la clase donde se definen.
- Se especifica nombre, nombre del concepto donde se define el atributo, tipo, unidad de medida, precisión y rango de valores (en el caso de valores numéricos), cardinalidad; atributos de instancia que pueden ser inferidos utilizando valores del atributo; etc.

Constantes:

- Describe en detalle las constantes definidas en el glosario de términos.
- Se especifica nombre, tipo de valor, unidad de medida para las constantes numéricas, los atributos que pueden ser inferidos usando la constante.

Axiomas formales:

- Componentes de modelado importantes en ontologías heavyweight.
- Expresiones lógicas siempre verdaderas utilizadas para especificar restricciones en la ontología.
- Para cada axioma se especifica nombre, descripción, la expresión lógica que describe formalmente el axioma utilizando FOL; los conceptos, atributos y relaciones a los que se refiere el axioma y las variables utilizadas.

Reglas:

- Componentes de modelado importantes en ontologías heavyweight.
- Utilizadas para inferir conocimiento en la ontología como valores de atributos, relaciones de instancia, etc.
- Methontology propone describirlas en forma paralela con los axiomas formales una vez definidos los conceptos, sus taxonomías, relaciones, atributos y constantes.
- Para cada regla se especifica nombre, descripción, la expresión que describe formalmente la regla; los conceptos, atributos y relaciones a los que se refiere la regla y las variables utilizadas en la expresión. Methontology propone especificar expresiones de reglas utilizando el template `if <conditions>then <consequent>`.
- Lado izquierdo de la regla consiste de conjunciones y átomos.
- Lado derecho de la regla es un átomo simple.

Instancias:

- Define las instancias relevantes que aparecen en el diccionario de conceptos en una tabla de instancias.
- Se especifica nombre, el nombre del concepto al que pertenece y valores de atributos si se conocen.

1.2.3. Adquisición de Conocimiento

Este paso se lleva a cabo de manera independiente en la metodología y su ejecución puede coincidir con otros pasos. Por lo general la adquisición de conocimiento se realiza en tres etapas: reuniones preliminares con los expertos, análisis y revisión de la bibliografía asociada al dominio y, una vez que se tiene un conocimiento base, se refina y detalla hasta completar la ontología.

1.2.4. Integración

Identificar ontologías candidatas que puedan ser reutilizadas en la ontología que se esta construyendo e incorporar aquellas piezas de conocimiento que sean de utilidad.

1.2.5. Implementación

Codificación del modelo conceptual en un modelo codificado en lenguaje ontológico.

1.2.6. Evaluación

Realizar un juicio técnico a la ontología, al ambiente de software asociado y a la documentación con respecto a un esquema de referencia en cada paso de la metodología (requerimientos de especificación, preguntas de competencia y/o el mundo real).

1.2.7. Documentación

Detallar clara y exhaustivamente cada paso completado y los productos generados.

2. Ingeniería Ontológica II: Lenguajes de consultas

2.1. RDQL

- Lenguaje de consulta del estilo SQL para consultar sobre las tripletas RDF.
- Permite especificar patrones que son contrastados con las tripletas del modelo para retornar un resultado.
- RDQL se utiliza para verificar si las preguntas de competencia se pueden responder con la ontología diseñada.



- `SELECT <variables>`
Especifica las variables a ser retornadas.
- `FROM <documentos>`
Indica la fuente RDF a ser consultada (URI). Se puede abreviar mediante el uso de `USING`. Van entre `<...>`. Para más de una fuente, se separan por comas `<... , ... , ...>`
- `WHERE <expresiones>`
Es la parte más importante de la expresión RDQL. Indica las restricciones de las tripletas RDF (*sujeto, predicado, objeto*). Se expresan por una lista de restricciones separadas por comas `<sujeto , predicado , objeto>`, donde sujeto, predicado y objeto pueden ser un valor literal o una variable RDQL.

- **AND** <filtros>
Especifica expresiones booleanas. Indica restricciones que las variables RDQL deben seguir.
- **USING** <espacios_de_nombres> **FOR** <URI>
Declara todos los espacios de nombre. Mecanismo de abreviación para URI's a través de la definición de prefijos.

2.1.1. Ejemplo



¿Dónde habitan los leones?

```

1 SELECT ?x FROM <animal.rdf>
2 WHERE (<animal:leon>, <animal:habitaEn>, ?x)
3 USING animal FOR <http://www.owl-ontologies.com/unnamed.owl>

```

¿Cuáles animales son mamíferos?

```

1 SELECT ?x ?grupo FROM <animal.rdf>
2 WHERE (?x, <animal:esTipo>, ?grupo) AND (?grupo, <rdf:type>, <animal:mamifero>)
3 USING rdf FOR <http://www.w3.org/1999/02/22-rdf-syntax-ns>,
4 animal FOR <http://www.owl-ontologies.com/unnamed.owl>

```

Retornar todos los recursos que tienen la propiedad *First Name* y su valor asociado:

```

1 SELECT ?x, ?fname FROM <vcard.rdf>
2 WHERE (?x, <vcard:FN>, ?fname)
3 USING vcard FOR <http://www.w3.org/2001/vcard-rdf/3.0#N>

```

2.2. Implementación

2.2.1. Jena

Framework desarrollado por HP Labs para manipular metadatos desde una aplicación Java.

Incluye varios componentes:

- Un Parser de RDF.
- API RDF. Permite crear y manipular modelos RDF desde una aplicación Java. Proporciona clases java para representar modelos, recursos, propiedades, literales, declaraciones.
 - *Recurso*: Todo aquello que se puede describir por una expresión RDF.
 - *Propiedad*: Una característica, atributo o relación usada para describir un recurso.
 - *Literal*: Un tipo de dato simple (String, Integer, etc).

- *Declaración*: Un recurso junto con una propiedad y con un valor asociado (*sujeto, predicado, objeto*).
- API de Ontologías con soporte para OWL, DAML y RDF Schema.
- Subsistema de razonamiento.
- Soporte para persistencia.
- RDQL: Lenguaje de consultas de RDF.

2.2.2. Jena 1

- Principalmente soporte para RDF.
- Capacidades de razonamiento limitadas.

2.2.3. Jena 2

- Incluye además una API para el manejo de Ontologías.
- Soporta el lenguaje OWL.

2.3. SPARQL

- Similar a SQL para RDF.
- Lenguaje de consultas Basado en RDQL.
- Modelo = patrones sobre grafos.
- *Patrón triple*: Similar a una tripleta RDF (*sujeto, predicado, objeto*) pero cualquier componente puede ser una variable de consulta. Se permiten sujetos literales.
- *Macheo de un patrón triple a un grafo*: vínculos entre variables y términos RDF.
- *Consulta*:
 - SELECT
Devuelve todo o un subconjunto de las variables enlazadas en una coincidencia de patrón de consulta. Formatos XML o RDF/XML.
 - CONSTRUCT
Devuelve un grafo RDF construido sustituyendo variables en un conjunto de plantillas triples.
 - DESCRIBE
Devuelve un grafo RDF que describe los recursos encontrados.
 - ASK
Devuelve si un patrón de consulta coincide o no.

3. Ingeniería Ontológica III: Evaluación

- Se describe la evaluación como el análisis de una ontología con respecto a una referencia durante cada etapa de su desarrollo.
- Los términos verificación y validación se encuentran incluidos en la evaluación.
- *Verificación y Validación (V&V)*: nombre dado a los procesos que permiten asegurar que la ontología desarrollada satisface su especificación y brinda la funcionalidad esperada por las personas que la solicitaron.
- La verificación se relaciona con la determinación de si la ontología fue correctamente construida. Mostrar que la ontología cumple con su funcionalidad y con los requerimientos no funcionales establecidos. ¿Se está construyendo el producto correctamente?
- La validación permite determinar si la ontología captura correctamente el mundo real que se está modelando. mostrar que la ontología cumpla con las expectativas del cliente. Es más general. ¿Se está construyendo el producto correcto?
- El objetivo de este proceso es determinar si la ontología desarrollada cumple con los principios de diseño y con los requerimientos descriptos en el DERO.
- Controlar que la ontología responde a las preguntas de competencia formuladas.
- Realizar un control de la ontología para la detección de anomalías o malas prácticas en su diseño.
- Existen diversas propuestas que contienen buenas prácticas de diseño para tener en cuenta al momento de desarrollar una ontología.
- Existe un catálogo de errores comunes entre los que podemos encontrar:
 - *Crear sinónimos como clases*: Determinar clases equivalentes en vez de especificar sinonimia. Dos términos sinónimos refieren a la misma clase, no a clases diferentes aunque sean equivalentes. Dos términos sinónimos serían *Auto* y *Automóvil*.
 - Utilizar la relación *es* en vez de utilizar elementos propios o primitivos de lenguajes semánticos como: `rdfs:subClassOf` (expresa subclase), `rdf:type` (que expresa membresía), `owl:sameAs` (expresa igualdad entre instancias).
 - *Creación de elementos no conectados*: Los elementos de la ontología están creados sin conexión con los elementos restantes de la ontología quedando miembros aislados. Un ejemplo de este caso es la definición de la clase `IntegranteEquipo` y no contar en la ontología con la clase `Equipo`.
 - *Definición de relaciones inversas erróneas*: Definir relaciones como inversas cuando no necesariamente son inversas. Por ejemplo, si algo se compra, `esComprado` no sería una relación inversa correcta.
 - *Mezclado de diferentes conceptos en la misma clase*: En este caso se crea una clase cuyo identificador hace referencia a dos o más conceptos. Un ejemplo de este error es el identificador `ProductosyServicios`.
 - *Falta de anotaciones*: En este caso los términos de la ontología carecen de atributos. Esta clase de propiedades facilitan la comprensión de la ontología y su usabilidad desde el punto de vista del usuario.
 - *Falta de disjunción*: En este caso, a la ontología le falta la definición de axiomas de disjunción entre clases o entre propiedades que deberían ser definidas como disjuntas. Por ejemplo, se pueden crear las clases `par` e `impar` sin que sean disjuntas, pero esta representación no es correcta basado en la definición de este tipo de números.
- La herramienta OOPS! (*Ontology Pitfall Scanner*) colabora con la detección de los errores catalogados en ontologías (errores comunes).
 - El motor de inferencia Pellet se utiliza para verificar la consistencia formal de la ontología.

- Las inconsistencias podrían presentarse relacionadas con la disposición de las clases (clases en la misma jerarquía y clases disjuntas), o pueden estar referidas a la relación entre las clases (rango y dominio), en el tipo de atributo o en las reglas de aplicación de la ontología.
- Con respecto a los requerimientos, la evaluación se realiza mediante el uso de las preguntas de competencia.
- Con este fin, la ontología debe instanciarse.
- Se consideran las preguntas de competencia realizadas en la especificación de requerimientos para ver si la ontología puede responderlas.
- Las respuestas obtenidas a las preguntas de competencia se deben someter a la consideración de expertos del dominio quienes deben determinar si son aceptables las repuestas.
- Durante el desarrollo de una ontología se realizan entrevistas con expertos. Es decir, las ontologías se evalúan desde su perspectiva de uso.