Sistema de Catalogación de Métricas e Indicadores con Potencia de Web Semántica

María de los Ángeles Martín

Grupo de Investigación y Desarrollo en Ingeniería de Software (GIDIS)

Calle 9 y 110, (6360) General Pico, La Pampa, Argentina

martinma@ing.unlpam.edu.ar

Presentada a la Facultad de Informática de la UNLP como parte de los requisitos para la obtención del título de
Magíster en Ingeniería de Software

Director : *Dr.* Luis **OLSINA**

Co-Director : *Dr.* Gustavo **ROSSI**

La Plata, Septiembre de 2004

Facultad de Informática
Universidad Nacional de La Plata - Argentina

Resumen

En las últimas décadas han surgido numerosas propuestas que definen conjuntos de métricas de calidad para la evaluación de productos y procesos de software en distintos dominios de aplicación. También existen en la actualidad algunos trabajos que proponen marcos conceptuales para la documentación de dichas métricas y catálogos que brindan información de utilidad en los proyectos de medición y/o evaluación de software.

Sin embargo, hemos observado que no existe un consenso general sobre cómo ha de documentarse la información de métricas e indicadores en forma efectiva, de manera de facilitar su explotación, reuso y aplicación a los procesos de medición y evaluación, de manera consistente. Además, si bien existen documentos estándares que definen parcialmente la terminología usada para el ámbito de modelos de calidad, métricas, y procesos de medición y evaluación [ISO9126, ISO14598, ISO15939], no existe una terminología universal consensuada entre los investigadores del área, y en consecuencia, la información relacionada proveniente de distintas fuentes no usa un mismo vocabulario, dificultado su aplicación en forma generalizada.

En esta tesis se muestra el trabajo realizado para la construcción del "Sistema de Catalogación de Métricas e Indicadores con potencia de Web Semántica", que intenta dar solución a dichos problemas. Este trabajo proporciona por un lado, una Ontología de Métricas e Indicadores que especifica un vocabulario común, con una semántica bien definida, facilitando la interpretación de la información de métricas e indicadores en forma uniforme, y permitiendo que puedan ser comparados los resultados obtenidos en distintos proyectos en forma consistente. Por otro lado, la implementación de un sistema de catalogación con potencia de web semántica, basado en dicha ontología, que facilita la explotación e intercambio consistente de información de métricas e indicadores tanto por parte de usuarios como por parte de aplicaciones, agentes y herramientas automáticas en la Web. Dicho sistema de catalogación, fue implementado con tecnologías de web semántica, adecuadas para proporcionar búsqueda y navegación semántica (cuya implementación también está detallada en este trabajo), de modo de proveer "entendimiento" automático de la información.

Agradecimientos

Al Dr. Luis Olsina, mi director, por acompañar mi trabajo con aportes enriquecedores y por el apoyo constante a mi tarea.

A la Facultad de Ingeniería, de la Universidad Nacional de La Pampa, por el soporte brindado que facilitó en buena parte este estudio.

A mis hijos Alejandro y Natalí, por soportar varias veces mi ausencia y acompañar mi esfuerzo con cariño y comprensión.

A mis colegas del grupo GIDIS, por haber compartido momentos de trabajo y compañerismo, y por alentarme constantemente. Especialmente, a Hernán Molina y Fernanda Papa por la importante contribución con la implementación del Sistema de Catalogación.

Índice

PARTE 1: INTRODUCCIÓN Y MOTIVACIÓN

PARTE 2: CONCEPTOS INTRODUCTORIOS A WEB SEMÁNTICA Y ONTOLOGÍAS

Capítulo 1- Introducción

1 Introducción

Los últimos avances en la investigación teórica y empírica en el área de Ingeniería de Software han dado como resultado la definición de numerosos conjuntos de métricas para la medición y evaluación de productos y procesos de software en distintos dominios de aplicación.

Observando el volumen cada vez mayor y heterogéneo de dicha información, y teniendo en cuenta su importancia como base fundamental para las actividades de medición o evaluación en el área de ingeniería de software, se puede ver que es necesario disponer de repositorios genéricos y soporte de catalogación que faciliten de un modo extensible, el almacenamiento, consulta, explotación y reuso de toda la información relacionada a métricas e indicadores.

Por otro lado, y no menos importante, para potenciar la interoperabilidad y reuso del repositorio, es esencial contar con un vocabulario común (Ontología).

Una ontología es la especificación explícita de una conceptualización [Gru95]. La ontología permitirá la documentación y explotación de información relacionada a métricas e indicadores desde una perspectiva general y sobre una base conceptual común, consensuada y formalmente especificada. De esta manera, el Sistema de Catalogación de Métricas e Indicadores (SCMI) que se propone en esta tesis, al estar basado en una ontología, posibilitará que la información obtenida de distintas fuentes y relacionada a distintos ámbitos de aplicación pueda ser compartida y reusada.

Además si observamos que las computadoras han dejado de ser dispositivos aislados y se han convertido en puntos de entrada a Internet, la red mundial de intercambio de información, recursos y transacciones de negocio, comprobamos que es muy importante contar con este apoyo tecnológico de intercambio de conocimiento en todos los ámbitos que requieran compartir información, y el dominio de métricas e indicadores no es una excepción. Por tal razón es fundamental que el SCMI sea implementado en la web.

Debido a que la información catalogada se podría encontrar dispersa en servidores heterogéneos distribuidos en todo el mundo, se necesita además, un mecanismo de procesamiento automático para facilitar su búsqueda y explotación; y para lograr este objetivo es necesario la representación semántica de los datos, de manera que pueda ser

procesada por una computadora. En consecuencia, se requieren metadatos que describan de una manera computable, dichos datos.

Afortunadamente, contamos actualmente con promisorios avances en las tecnologías que dan soporte a la automatización del procesamiento de información en la Web, y que seguramente conducirá a una nueva generación de la Web, la denominada "Web Semántica".

En esta tesis se especifica el diseño del Sistema de Catalogación de Métricas e Indicadores basado en la definición de la ontología de Métricas e Indicadores [Mar03b] y su implementación con los lenguajes estándares que se han definido para la Web Semántica (WS), de manera de facilitar su distribución en la web, en repositorios de catalogación heterogéneos y su explotación tanto por humanos como por agentes computarizados.

1.1 Principales Contribuciones

Las principales contribuciones de esta tesis son:

- **Diseño del Sistema de Catalogación de Métricas e Indicadores con potencia de Web Semántica.**

Se presentará el diseño del SCMI basado en una arquitectura en capas adecuado para ser implementado con tecnologías de la Web Semántica. En el diseño se mostrará la composición modular de los subsistemas y las funciones de cada módulo y de cada capa de la arquitectura. Además se especificarán las tecnologías que fueron usadas en la implementación del sub-sistema semántico de consultas para lograr los objetivos propuestos para el catálogo y se definirá un marco conceptual que proporcionará un vocabulario e infraestructura homogénea para su efectiva explotación.

Es importante destacar en este punto que el SCMI propuesto fue diseñado como un repositorio de metadatos de métricas e indicadores y no para guardar instancias de valores resultantes de su aplicación en proyectos específicos, es decir, se guarda información de las métricas y los indicadores, no de las medidas obtenidas o de los valores indicadores calculados.

- **Especificación de una ontología para el ámbito de métricas e indicadores.**

Como resultado de un trabajo de investigación sobre conceptos definidos en el área de métricas e indicadores, y teniendo en cuenta estándares internacionales [ISO9126, ISO14598, ISO15939], trabajos de investigación [Bri02, Gen03, Kitch01, Ols00, Zus98] y la discusión en grupos de especialistas sobre una terminología común para dicho ámbito, se definió una ontología para el dominio de métricas e indicadores,

especificando conceptos, relaciones, propiedades y axiomas. Esta ontología permite compartir información relacionada a métricas e indicadores con una interpretación unificada.

- **Estudio de la potencia que los aspectos de web semántica le confieren al SCMI**.

A través de un prototipo realizado para tal fin y de ejemplos concretos, se muestra la potencia del SCMI con búsqueda y navegación semántica, en la explotación del catálogo tanto por parte de usuarios comunes como de agentes y herramientas automáticas. Además, se muestra cómo este enfoque puede dar soporte a las características de interoperabilidad, generalidad y procesamiento semántico tan importantes a la hora de reusar la información de métricas e indicadores.

1.2 Estructura de la Tesis

La tesis se organiza de la siguiente manera: En el capítulo 2 se presentan los objetivos y motivaciones del trabajo, como así también las principales contribuciones que aporta. Para una mejor comprensión de esta tesis, en los capítulos 3 y 4 se introducen los principales conceptos de la Web Semántica y de Ontologías respectivamente. En el capítulo 5 se define una ontología para métricas e indicadores sobre la que se basará el SCMI. En el capítulo 6 se muestra la arquitectura en capas del Sistema de Catalogación de Métricas e Indicadores (SCMI) propuesto, se muestran detalles de su implementación, y se analiza la potencia que las características de web semántica le confieren al catálogo, a través de la implementación de la búsqueda y navegación semántica. Finalmente, en el capítulo 7 se presentan las conclusiones y se analizan las líneas de trabajos futuros.

Capítulo 2- Motivación

La Ingeniería de Software, al igual que otras ingenierías, necesita de métricas e indicadores para poder especificar, predecir, evaluar y analizar distintos atributos y características de los entes (productos, procesos, etc.) que participan en el desarrollo y mantenimiento del software.

Ingeniería de Software, tal como fue definido en el estándar IEEE 610.12 [IEEE90], es "La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software, es decir la aplicación de ingeniería al software".

La aplicación de un enfoque cuantificable al desarrollo, operación y mantenimiento del software es una tarea compleja que requiere disciplina, estudio y conocimiento de las métricas e indicadores adecuadas para los distintos objetivos de medición y evaluación, con el fin de garantizar la calidad.

Por todo ello, y por la importancia que le damos al hecho de medir y evaluar, para hacer del desarrollo de software una verdadera ingeniería, es que vemos la necesidad de contar con métodos y sistemas de medición y evaluación robustos, y con la información de métricas e indicadores, necesaria para que las mediciones resulten más objetivas, precisas, repetibles, reproducibles y significativas.

En los últimos años varios trabajos de investigación han realizado importantes aportes en la definición de métricas y en la proposición de metodologías y modelos a ser aplicados en la Ingeniería de Software [Alb83, Cap91, Charis, Con86, DavCG, McC76, Men01, Ols01, SMLab, TotMe, ZDMIS].

A pesar del indiscutible valor de los avances logrados por dichas investigaciones en la definición de métricas, a la hora de aplicar dicha información en el ámbito de Ingeniería de Software, nos encontramos con algunos de los siguientes problemas:

- ✓ No se cuenta con un catálogo suficientemente completo y estructurado con información clasificada relacionada a métricas e indicadores que sirva como base para la selección de las mismas y como apoyo al diseño del proceso de medición o evaluación.

- ✓ La información disponible sobre métricas, indicadores y modelos de evaluación no está basada en criterios comunes, y en muchos casos está definida para

dominios específicos, dificultando su aplicación a procesos de evaluación genéricos y su reuso en distintos proyectos de evaluación y/o medición.

- ✓ No existe una terminología común consensuada para el ámbito de métricas e indicadores que permita una fluida comunicación de información entre distintos proyectos de evaluación y facilite la aplicación de resultados obtenidos en distintos trabajos de investigación y casos de estudio. Si bien existen algunos estándares internacionales ISO/IEC donde se definen términos relacionados a modelos de calidad de software, medición y proceso de evaluación [ISO9126, ISO14598, ISO15939], se puede observar en dichos documentos, la ausencia de definición de algunos conceptos, y a veces, los mismos términos están definidos de distinta manera en distintos estándares.

El Sistema de Catalogación de Métricas e Indicadores (SCMI) con potencia de Web Semántica que se propone en esta tesis, tiene como objetivo dar solución a los problemas planteados anteriormente y cubrir una importante carencia que es la definición de una ontología para el dominio de Métricas, Indicadores y Conceptos Calculables de software.

2.1 Estado del Arte en Sistemas de Catalogación de Métricas

Actualmente existen varios trabajos reconocidos de investigación relacionados a la definición de métricas de productos y procesos de software, y algunos trabajos incipientes de catalogación de información de las mismas.

Respecto a trabajos de definición de métricas de software podemos citar algunos ejemplos como ser: las métricas de complejidad de McCabe [McC76], métricas de longitud de código fuente [Con86], métricas de punto de función [Alb83, Cap91], métricas para productos web [Men01, Ols01], entre otros.

Sin embargo, la comunidad de ingenieros de software no ha acordado un conjunto de métricas universalmente aceptada por el área como ha sucedido en disciplinas más tradicionales como la física o la química. Además, al momento de reusar los resultados de todos estos trabajos de investigación nos encontramos con el problema de que la información no está catalogada en forma homogénea y estructurada, y no usa una terminología común dificultando la aplicación de los resultados obtenidos en investigaciones previas, al tener que recurrir a distintas fuentes de información, que en muchos casos se contradicen.

Algunos organismos y empresas privadas [Charis, DavCG, SMLab, TotMe, ZDMIS], han realizado trabajos de selección y/o catalogación de métricas para software, para dominios de aplicación específicos. El objetivo principal de estos catálogos es brindar documentación que ayude a ingenieros y usuarios en el soporte de

medición y evaluación del software. En algunos casos, inclusive se provee de las herramientas necesarias que aplican dichas métricas y/o dan soporte al proceso de análisis.

En las siguientes secciones mostramos, por un lado, algunos ejemplos de sistemas y catálogos de información de métricas para software existentes en la actualidad, y por otro, algunos trabajos de investigación que definen marcos de trabajo y modelos para almacenar información de métricas de software.

2.1.1 Ejemplos de Sistemas de Información y/o Catalogación de Métricas de Software

En esta sección mostramos ejemplos de algunas empresas u organismos (ver Tabla 2.1) que han realizado trabajos donde se documentan métricas de software que pueden ser usados durante distintas fases del ciclo de vida del software, si bien no toda la información de métricas se ha organizado de la misma manera y no todas las empresas brindan los mismos servicios de información.

Empresa/ Organismo	Comentario
SMLab	Laboratorio de Medición de Software de la Universidad de Magdeburg, Alemania [SMLab].
ZD-MIS	Sistema de información de Medidas de Zuse y Drake. Empresa privada [ZDMIS].
Charismatek Software Metrics	Métricas de Sofware Charismatek. Empresa privada [Charis].
Total Metrics	Servicio de Consultoría [TotMe].
David Consulting Group	Servicio de Consultoría [DavCG].
ISO 9000	Organización Internacional de Estandarización [ISO9126, ISO14598, ISO15939].

Tabla 2.1 Trabajos relacionados a Documentación de Métricas de Software

A continuación realizamos una breve descripción de cada uno de ellos:

El **Laboratorio de Medición de Software** (SMLab) [SMLab] de la Universidad de Magdeburg, Alemania, brinda extensa información de métricas y herramientas. Los miembros del equipo de trabajo de SMLab guiados por el Prof. Reiner R. Dumke, han concentrado las actividades de distintas comunidades relacionadas a métricas de software, que van desde foros, conferencias y workshops, a artículos y aplicaciones de herramientas de métricas. Presentan un amplio rango de herramientas y temas

relacionados a métricas y han creado una gran comunidad de participantes que incluye miembros de todo el mundo.

Aún cuando los servicios de información brindada por este laboratorio pueden ser de mucha utilidad en los proyectos de medición de software, la información proporcionada no está catalogada en forma estructurada y uniforme, dificultando su reuso. El ingeniero de software debe realizar una difícil labor de búsqueda y análisis de la información contenida en las publicaciones ofrecidas antes de poder aplicar sus resultados.

El paquete **Zuse/Drake Measure-Information-System (ZD-MIS)** [ZDMIS], es un sistema que cataloga un gran conjunto de métricas. Este proyecto fue iniciado por Horst Zuse y Karin Drave, además de ofrecer un importante compendio de métricas, presenta un marco de trabajo para prueba de software. El sistema ZD-MIS contiene una base de datos con más de 1500 métricas de software de distintas clases, descriptas de manera uniforme. De cada métrica se describen distintos aspectos como ser: definición de la métrica, atributos que mide, tipo de medida y tipo de escala, clase de entidad de la métrica (producto, recurso o proceso), entre otros. La base de datos contiene también más de 180 métricas para software Orientado a Objetos. La información de métricas se puede seleccionar por más de una docena de criterios, además permite confeccionar una guía de métricas de acuerdo a criterios de usuario, ayudando a científicos, ingenieros y otros participantes a seleccionar métricas de acuerdo a sus necesidades particulares.

El sistema también provee más de 1600 referencias bibliográficas y un glosario de más de 600 términos, que ayudan al usuario a tener un mejor entendimiento del área de métricas de software.

Si bien este sistema provee un catálogo estructurado que describe un conjunto de métricas clasificadas de acuerdo a varios criterios, el mismo no es interoperable, en el sentido de que no funciona sobre cualquier plataforma (sólo disponible para plataformas Windows sobre PC´s), y la base de datos propietaria se ofrece cerrada, impidiendo el procesamiento de la información de métricas en forma automática por computadoras.

Además este sistema no ofrece información de apoyo al análisis de los resultados de las mediciones, ya que sólo posee información de métricas pero no de indicadores, tan importante como ayuda al proceso de evaluación del software.

La empresa **Charismatek Software Metrics** [Charis], brinda herramientas de medición de software y servicios de consultas de métricas y métodos de medición, ellos desarrollaron la herramienta de Punto de Función WORKBENCH, que aplica métricas de software basadas en las mediciones de puntos de función.

La información sobre métricas de software que ofrece esta empresa no está catalogada ni clasificada y se limita casi exclusivamente a temas relacionados a mediciones de puntos de función, dejando sin cubrir el resto de los tópicos referentes a medición y evaluación.

Otras compañías que proveen documentos y servicios de consultas para la aplicación de métricas de software son: **Total Metrics** [TotMe], que provee información para la aplicación práctica de métricas de software y **David Consulting Group** [DavCG], que han escrito varios artículos sobre sus experiencias en la implementación de métricas de software.

Estas compañías tampoco brindan información catalogada en forma estructurada y se limitan a proporcionar documentación en forma de artículos.

Es muy importante destacar también el trabajo realizado por el Organismo Internacional de Estandarización (ISO) [ISO9126, ISO14598, ISO15939], que ha definido estándares para la administración de calidad del software y procesos de medición. Pero este organismo no ha definido ni recomendado aún, qué conjunto de métricas e indicadores usar en los distintos casos de evaluación de software.

Observando el estado actual de los sistemas de catalogación de métricas e indicadores podemos concluir que:

A pesar de la importancia del esfuerzo realizado por científicos, empresas y organismos en la catalogación de métricas de software e información relacionada, y de la construcción de herramientas de apoyo al proceso de medición, no existen en la actualidad criterios comunes en la documentación y administración de dicha información y no hay un acuerdo en la estructuración homogénea de los datos, de manera que puedan ser reusados fácil y consistentemente.

Además, la información se encuentra dispersa en distintos sistemas y recursos, y en algunos casos se desconocen las fuentes, lo que dificulta su búsqueda y acceso.

Otro problema no menor que tienen estos sistemas, es que al no contar con una estructura de información común, su aplicación y explotación en forma conjunta es una tarea de difícil resolución.

2.1.2 Ejemplos de Marcos de Trabajo para Almacenar Información de Métricas de Software

En esta sección analizaremos dos trabajos de investigación donde se proponen marcos de trabajo para almacenar información de métricas de software.

El primero de ellos es la propuesta de Kitchenham et al [Kitch01], en su artículo "Modeling Software Measurement Data". Este trabajo presenta un método para especificar modelos de conjuntos de datos con el objetivo de capturar las definiciones y relaciones entre las medidas de software.

Los autores afirman que es necesario una definición apropiada de los conjuntos de datos de medidas, como así también los metadatos que definen dichos conjuntos de valores, para asegurar que los valores de las medidas sean repetibles, comparables y puedan ser analizados correctamente.

En este trabajo se plantea la necesidad de un método estándar para especificar conjuntos de datos de medidas de software, que esté debidamente documentado y que pueda ser intercambiado con confianza; es más, los autores sugieren que el método propuesto en este trabajo, puede ser usado como dicho estándar.

El método presentado está definido en términos de un modelo de datos de Entidad-Relación a nivel conceptual, permitiendo tanto que puedan ser modelados los conjuntos de datos de las métricas, como las instancias de los valores obtenidos.

Es decir, este modelo serviría tanto para almacenar información que define las métricas, lo que los autores presentan como el modelo Entidad-Relación de los metadatos, como los valores de las instancias de dichas métricas, a través del modelo Entidad-Relación de los conjuntos de datos de las medidas.

Pero este trabajo falla en especificar cómo la información de métricas (los metadatos) pueden ser compartidos, reusados y explotados. Es decir, no muestra detalles de implementación ni presenta soluciones al problema de interoperabilidad y reuso. Además el método, por estar definido en un modelo de Entidad-Relación, carece de detalles conceptuales de composición, herencia y otro tipo de relaciones entre las entidades asociadas. Por ejemplo, no se modelizan las métricas indirectas (es decir, métricas que se componen de otras métricas y una función de cálculo); ni se considera la relación entre distintas entidades, tal como la relación entre proyecto y productos o procesos. Modelos para la relación entre entidades y métricas son elementos importantes para explotar la información relacionada al proceso de medición.

Un trabajo posterior de Lawler y Kitchenham [Law03], presenta una tecnología para modelar mediciones que se basa en una rigurosa definición de las mediciones a ser recolectadas en forma automática, las métricas (los autores se refieren a ellas como las funciones sobre las mediciones atómicas que deberán ser calculadas) y las asociaciones entre las medidas y las herramientas de soporte al proceso de desarrollo (herramientas de diseño, de administración de proyectos, de desarrollo, etc). El

problema que tiene esta tecnología es que solamente tiene sentido si está embebida en una herramienta de medición automática y puede ser aplicada a proyectos de software totalmente automatizados por herramientas de diseño y desarrollo. Este trabajo además no presenta ningún modelo de cómo almacenar y explotar la definición de las métricas que usa.

Otro trabajo en esta dirección es la propuesta de Olsina et al [Ols02b], en la que se presenta un marco conceptual capaz de proporcionar un vocabulario, modelo e infraestructura homogénea (llamado por los autores, framework), que permitiría catalogar métricas de software en un repositorio común de manera de facilitar la especificación, documentación y explotación de tales métricas en forma genérica.

El repositorio propuesto en este informe está basado en estándares de amplia aceptación como pueden ser UML, XML, RDF y tecnologías de Web Semántica lo que permite el desarrollo de herramientas industriales para tratar de automatizar los procesos de especificación, consulta y análisis de métricas, y facilita el reuso y la explotación de la información relacionada a métricas de una manera distribuida y homogénea, tanto por parte de agentes humanos como automatizados.

Ese artículo presenta las bases para tal framework, analiza los requisitos generales que debería cumplir, propone la estructura del repositorio general de métricas (basado en un modelo conceptual de métricas), y muestra una arquitectura concreta para su implementación. En particular, se hace hincapié en una serie de tecnologías (por ejemplo, esquemas XML), que se usarían en la implementación del repositorio.

Si bien este trabajo presenta una base conceptual importante hacia la construcción de un catálogo genérico de información relacionada a métricas de software, que sirve como punto de partida para el repositorio especificado en esta tesis, el trabajo puede ser enriquecido desde varios puntos de vista:

El modelo conceptual presentado, sólo considera la catalogación de métricas sin la especificación y documentación de indicadores ni criterios de evaluación de acuerdo a distintas necesidades de información, tan importantes en el proceso de medición y evaluación de software.

La arquitectura propuesta, no está basada en la definición de una terminología común, consensuada (es decir, no está basada en una ontología). Y no se especifica cómo la información del repositorio podría en este caso, ser explotada en forma generalizada y uniforme.

2.1.3 Conclusión sobre el Estado del Arte en Sistemas De Catalogación de Métricas e Indicadores

Como se puede observar en las secciones anteriores, varios trabajos se han realizado en pos de la definición, documentación, catalogación y explotación de información de métricas de software, como así también de la construcción de herramientas y servicios de apoyo al proceso de evaluación. Sin embargo, surgida la necesidad de seleccionar métricas o indicadores, de establecer criterios de evaluación, de seleccionar herramientas de apoyo, y de buscar información relacionada al área de medición o evaluación, nos encontramos con que los trabajos antes mencionados fallan en alguno o varios de los siguientes aspectos:

La información no está catalogada en forma conjunta, esto dificulta el reuso de resultados obtenidos en investigaciones previas, al tener que recurrir a distintas fuentes de información, que en muchos casos se desconocen o, peor aún, se contradicen.

La información no está estructurada consistentemente, la gran cantidad de información sobre métricas existente actualmente y en continuo crecimiento como resultado de distintos trabajos de investigación, es totalmente heterogénea en cuanto a su esquema y la naturaleza de sus datos. Esta característica impide su procesamiento en forma conjunta y la posibilidad de poder relacionar y/o comparar los resultados obtenidos en distintos proyectos de evaluación .

Es importante contar con un catálogo de información de métricas estructurado uniformemente y suficientemente especificado aún cuando dicho catálogo no contemple el almacenamiento de valores (instancias) de las medidas, como es el caso de mi propuesta, pues partiendo de información de métricas debidamente documentada, los valores obtenidos como resultado de su aplicación pueden ser comparados y/o transformados con confianza.

No se usa una terminología común, si bien existen documentos estándares que definen parcialmente la terminología usada para el ámbito de modelos de calidad, métricas, indicadores y procesos de evaluación [ISO9126, ISO14598, ISO15939], no existe una terminología universal consensuada entre los investigadores del área, y la información relacionada proveniente de distintas fuentes, no usa una mismo vocabulario, dificultado su aplicación en forma generalizada.

No se especifica cómo estas métricas podrían ser coleccionadas y/o almacenadas, si bien existen algunos trabajos de investigación [Kitch01, Ols02b] que especifican modelos para la documentación de datos de métricas y mediciones, estos fallan en que no están suficientemente especificados y no se basan en tecnologías estándares que faciliten la recuperación, reuso e interoperatividad.

2.2 Necesidad de un Catálogo de Métricas e Indicadores basado en una Ontología con Potencia de Web Semántica.

Teniendo en cuenta el análisis del estado del arte en sistemas de catalogación de métricas de la sección anterior, y evaluando la importancia de contar con un catálogo para soporte a los procesos de medición y evaluación en el área de Ingeniería de Software observamos la necesidad de contar con un sistema debidamente estructurado para explotar la información sobre métricas e indicadores.

Por otra parte, es conveniente que dicho catálogo pueda ser fácilmente extensible y escalable para que los resultados obtenidos por distintos trabajos de investigación, puedan ser reusados en otros proyectos y además la información esté siempre disponible y sea de fácil acceso, aún cuando se haya generado en distintos lugares, distribuidos geográficamente. Para cumplir con estos requisitos vemos necesario que el catálogo sea implementado en forma distribuida en la Web. Esto resolvería una de las debilidades observadas en la sección 2.1.1 en el sistema ZD-MIS, desarrollado para PC´s con una base de datos centralizada.

Además considero importante contar con una terminología común para que la interpretación de la información sea uniforme, y puedan ser comparados los resultados obtenidos en distintos proyectos. Es decir, es necesario que el catálogo esté basado en una ontología.

Para permitir el desarrollo de herramientas que automaticen los procesos de especificación, consulta y análisis de métricas e indicadores, de una manera interoperable, es necesario que la semántica de la información del catálogo también esté explícitamente publicada en la web, es decir, es necesario que el catálogo se desarrolle con las nuevas tecnologías de Web Semántica. Esto permitirá que la información del catálogo pueda ser procesada no sólo por seres humanos sino también por agentes, herramientas automáticas u otro software de computadoras para potenciar su explotación.

Con el fin de solucionar los problemas que se plantean en el estado del arte, esta tesis contribuye en la especificación y diseño del Sistema de Catalogación de Métricas e Indicadores basado en una Ontología con Potencia de Web Semántica.

2.3 Características Deseables del Catálogo

En esta sección enumeramos algunas de las características que debiera cumplir el Sistema de Catalogación de Métricas e Indicadores para que sea de utilidad y sirva de apoyo a los procesos de evaluación en el área de Ingeniería de software e Ingeniería Web.

- ✓ **Interoperabilidad:** que facilite el intercambio de información entre distintas aplicaciones de usuario o herramientas, aún cuando estén implementadas en plataformas distintas. Para garantizar la interoperabilidad en este sistema, se propone su implementación bajo estándares de marcado (XML, SXML), estándares de Web semántica (como RDF, RDFS, OWL) y de arquitecturas de ambientes distribuidos de amplia difusión.

- ✓ **Generalidad:** que proporcione un marco común de trabajo en la documentación de métricas e indicadores que pueda ser aprovechado en distintos ámbitos y desde distintas perspectivas. Para este fin, se propone un modelo que no está comprometido con un dominio de software específico.

- ✓ **Simplicidad:** que el uso del repositorio, ya sea por parte de los administradores, usuarios o herramientas finales deba hacerse a través de interfaces y operaciones estándares simples, como por ejemplo: Registro, Búsqueda, Exploración, etc. con el fin de proveer un entorno fácil de usar, aprender y operar.

- ✓ **Extensibilidad:** que facilite la publicación de nueva información y la actualización de la existente. Para lograr este objetivo, tanto la estructura como la semántica de los datos (los metadatos) formarán parte del sistema de catalogación.

- ✓ **No-ambigüedad (en la interpretación de la información):** Para que la información del catálogo pueda ser claramente interpretada y, como consecuencia de ello, los resultados de su aplicación a distintos proyectos de evaluación puedan ser confiablemente comparados. Esto se podrá lograr si se establece una terminología común, consensuada y formalmente especificada; es decir, el catálogo estará basado en una Ontología.

Capítulo 3- La Web Semántica

"*La Web semántica no constituye una Web independiente, sino una ampliación de la actual; en la cual la información está dotada de significados bien definidos, con el fin de permitir un mejor trabajo en cooperación entre humanos y computadoras*", Tim Berners-Lee, James Hendler, Ora Lassila [Ber01].

3.1 Introducción

A través de distintos trabajos de investigación y desarrollo ya se están dando los primeros pasos tendientes a transformar la web existente en su nueva generación: "La Web Semántica". Tim Berners-Lee et al, en el artículo "The Semantic Web" [Ber01] presentan las principales características que tendrá la futura web y afirman que en un futuro no lejano, se producirán desarrollos importantes que introducirán prestaciones nuevas, al lograr que las máquinas multipliquen su capacidad de procesar y "comprender" los datos dispersos en la web y que hoy solamente se exhiben en la pantalla.

Hasta ahora, la Web se ha desarrollado con suma rapidez, pero concebida mucho más como medio de proporcionar documentos a los humanos que para la manipulación de datos e informaciones procesables de forma automática. La Web semántica aspira a cubrir esta deficiencia.

3.1.1 La Representación del Conocimiento

Para que la Web semántica funcione, las computadoras han de tener acceso a repositorios de información debidamente estructurados y a conjuntos de reglas de inferencia que puedan utilizar para efectuar razonamiento automático. Tales sistemas se han venido estudiando en inteligencia artificial desde mucho antes de la creación de la Web. La representación del conocimiento, nombre que suele recibir este área, se encuentra en la actualidad en un estado comparable con la del hipertexto antes del advenimiento de la Web: por ahora es una buena idea, de la que existen algunas demostraciones interesantes [Bla98, Fer97, Fra99], pero para que manifieste con plenitud sus posibilidades ha de quedar vinculada a un sistema global.

Los sistemas tradicionales de representación del conocimiento han sido, de ordinario, sistemas centralizados, que exigen a todos sus usuarios que compartan exactamente la misma definición de conceptos comunes, como "padre" o "vehículo".

Pero el control central resulta agobiante; y, por otra parte, si se aumenta el tamaño y alcance de tales sistemas pronto resultan imposibles de manejar.

Los investigadores de la Web semántica [W3Csw], por el contrario, proponen un sistema descentralizado para representar el conocimiento, aunque aceptan que la necesaria versatilidad exige un precio: la aparición de contradicciones, proposiciones falsas y de preguntas sin respuesta. Este planteamiento se asemeja al de la Web común: cuando comenzó el desarrollo de Internet, sus opositores señalaron que nunca llegaría a ser una biblioteca bien organizada, a falta de una base de datos central y careciendo de la estructura necesaria, nunca se podría tener la seguridad de encontrar todo cuanto contiene y que la información encontrada sea de calidad. Pero la potencia expresiva del sistema hizo disponibles vastas cantidades de información, y los motores de búsqueda (que hace un decenio hubieran parecido muy poco prácticos) producen ahora índices suficientemente completos de gran parte del material diseminado por ella.

El problema que la Web semántica ha de resolver, consiste en proporcionar un lenguaje capaz de dar expresión tanto a datos como a las reglas para razonar sobre datos, y que permita, además, la exportación a la Web de las reglas de inferencia de cualesquiera sistemas de representación de conocimientos que ya existan.

La tarea que la comunidad de la Web semántica tiene ante sí en este momento consiste en dotar de lógica a la Web, es decir, de medios para usar reglas de inferencia, elegir vías de acción y responder a preguntas.

Ya están disponibles tres importantes tecnologías para el desarrollo de la Web semántica: el lenguaje *XML (eXtensible Markup Language,* lenguaje de marcado extensible) [Bra98], el lenguaje RDF *(Resource Description Framework,* un marco de descripción de recursos) [Bri00, Las99] y el lenguaje OWL (*Ontology Web Language,* lenguaje de ontologías en la web) [Bech04]. El lenguaje XML permite crear etiquetas propias, es decir, marcas o anotaciones no visibles ("tags"), como <telefono> o <universidad>, que marcan a datos insertados en páginas de la Web o a secciones de texto. El lenguaje XML permite a los usuarios añadir a sus documentos estructura arbitraria, pero nada dice sobre lo que significan tales estructuras.

Para expresar significado, se usa el lenguaje RDF, que lo codifica en conjuntos de ternas. Los elementos de cada terna vienen a ser como el sujeto, el verbo y el objeto de una oración elemental. Estas ternas pueden escribirse usando marcas XML. En el descriptor RDF, los documentos contienen sentencias que declaran qué objetos particulares (personas, documentos o lugares) tienen propiedades ("es esposa de", “es creado por" o “se encuentra ubicado en”) con ciertos valores (otra persona, otra página de la Web u otro lugar). Esta estructura proporciona un medio natural para describir la vasta mayoría de los datos procesados mediante máquinas. El sujeto y el objeto quedan identificados mediante un URI (identificador universal de recursos), como los utilizados

en los vínculos de las páginas de la Web. (Los URL -sigla de *Universal Resource Locators*- son los URI de tipo más utilizado.). También los verbos quedan identificados mediante URI, lo cual permite definir conceptos o verbos nuevos sin más que definir el URI correspondiente en algún lugar de la Web.

El lenguaje humano se enriquece al utilizar un mismo término para aludir a entes distintos. En automatización ocurre lo contrario. Por ejemplo, en el diccionario de la Real Academia Española, el término "medida" tiene al menos dos significados: 1.-Acción y efecto de medir y 2.-Expresión del resultado de una medición. Imaginemos que queremos compartir los resultados de mediciones realizadas con otro grupo de evaluación, entonces transfiero las medidas desde mi base de datos a la suya, sin saber que el concepto de "medidas" en mi base de datos se refiere a la acción de medir y contiene la fecha en que se realizó; si el otro grupo interpreta estos datos como valores resultantes de las mediciones, las conclusiones serán incorrectas. Este problema se resuelve utilizando un URI distinto para cada concepto concreto. De esta manera la medida como acción de medir puede distinguirse de los valores obtenidos.

Las ternas del descriptor de recursos RDF forman mallas de información sobre entes relacionados. Dado que RDF utiliza XML para codificar la información en los documentos, los URI garantizan que los conceptos no sean meras palabras de un documento, sino que están vinculados a una definición unívoca que todo el mundo puede encontrar en la Web. Supongamos, por ejemplo, que tenemos acceso a una variedad de bases de datos con información de métricas, incluida su descripción. Si queremos encontrar métricas relacionadas a un determinado atributo, necesitamos saber en cada base de datos cuáles son los campos que representan nombres de métricas, y cuáles atributos mide. RDF puede especificar que "(el campo 5 de la base de datos A) (es un campo de tipo) (atributo)," utilizando para cada componente URI´s en lugar de frases.

3.1.2 Ontologías

Otro problema con que nos podemos encontrar al compartir datos en la web, es que dos bases de datos pueden utilizar identificadores diferentes para un mismo concepto, por ejemplo, atributo y attribute. Un programa que se proponga comparar o combinar información tomada de ambas bases de datos ha de estar informado de que estos dos términos se están empleando para designar una misma cosa. Otro componente importante de la Web semántica son las ontologías, ellas proporcionan una solución a este problema.

Las ontologías están formadas por colecciones de definiciones. En filosofía, una ontología es una teoría que trata de la naturaleza de la existencia, o del tipo de cosas que existen. Los investigadores de inteligencia artificial y de la Web se han apropiado del

término y lo han incorporado a su jerga; para ellos, una ontología es un archivo o un documento que define formalmente términos y las relaciones entre términos [Bor97] . Reglas lógicas integradas a las ontologías, harán posible el razonamiento automático sobre el conocimiento que esté disperso en la web. Con conceptos (y sus relaciones) formalmente definidos y reglas de inferencia debidamente especificadas, un programa o agente, podrá deducir nuevas sentencias (declaraciones) a partir de las sentencias existentes.

3.1.3 Agentes

La auténtica potencia de la Web semántica no se hará efectiva hasta que se creen muchos programas que recopilen contenidos de la Web tomados de diversas fuentes, procesen la información e intercambien sus resultados con otros programas. La eficacia de estos agentes informáticos crecerá exponencialmente conforme vayan estando disponibles en la Web más contenidos entendibles por computadoras y más servicios automáticos (entre ellos, otros agentes) [Ber01]. La Web semántica promueve esta sinergia: incluso agentes no expresamente diseñados para trabajar en colaboración pueden transferirse datos uno al otro si éstos se hallan provistos de semántica.

Las firmas digitales constituirán otra característica esencial. Las firmas digitales son bloques encriptados de datos que las computadoras y los agentes pueden utilizar para comprobar que la información agregada procede de una fuente identificada y digna de confianza. Parece obvio que los agentes deberían desconfiar de las aserciones que leyeran en la Web semántica si no han verificado las fuentes de información.

Aunque existen ya en la Web muchos servicios automáticos, por carecer de semántica los demás programas y los agentes en particular no tienen forma de localizar uno que realice una tarea específica. Este proceso, la búsqueda de servicios, sólo podrá producirse cuando exista un lenguaje común (ontología) para describirlos de modo tal que otros agentes puedan "comprender" tanto la función ofrecida como la manera de aprovecharla. Servicios y agentes pueden dar a conocer su función depositando, por ejemplo, sus descripciones en anuarios semejantes a las Páginas Amarillas.

En la actualidad se dispone de unos pocos sistemas de búsqueda de servicios, pero son de bajo nivel. Estas iniciativas, se basan en la estandarización de un conjunto predeterminado de descripciones de funciones. Pero la estandarización no puede ir demasiado lejos, pues no hay forma de prever todas las necesidades futuras.

La Web semántica es, por el contrario, más flexible. Los agentes productor y consumidor pueden llegar a entenderse mediante el intercambio de ontologías, que proporcionan el vocabulario requerido para "dialogar". Los agentes podrían incluso crear por sí mismos nuevas capacidades de razonamiento cuando descubrieran nuevas ontologías.

Un proceso típico consistirá en la creación de una "cadena de valores" donde se van transfiriendo subcadenas de información de un agente a otro, cada uno de los cuales "añade un valor", tratando de construir el resultado definitivo solicitado por el usuario final. Por ejemplo, si hubiera agentes que brindaran servicios automáticos de búsqueda de métricas adecuadamente definidas para una necesidad específica, éstos podrían comunicar los resultados a herramientas automáticas de medición para obtener los valores de las mediciones, que a su vez podrían ser pasados a otro agente para su análisis y así siguiendo.

Cabe destacar que para crear automáticamente bajo demanda cadenas de valor complejas, algunos agentes se valdrán no sólo de la Web semántica, sino también de técnicas de inteligencia artificial. Pero la Web semántica proporcionará los cimientos y la estructura necesarias [W3Csw] para hacer más realizables tales técnicas.

En la siguiente etapa, la Web semántica desbordará el mundo virtual y se extenderá hasta nuestro mundo físico. Las URI pueden apuntar a cualquier cosa, sin exceptuar a las entidades materiales. Significa esto que podemos servirnos de ontologías para describir objetos, como teléfonos móviles o aparatos de televisión. Tales equipos pueden anunciar, al estilo de los agentes informáticos, las funciones que pueden realizar y el modo en que pueden ser controlados.

Ya se han dado los primeros pasos en esta dirección. Se ha trabajado en el desarrollo de una norma para la descripción de las capacidades funcionales de los equipos (por ejemplo, los tamaños de pantalla) y las preferencias del usuario. Este estándar, construido sobre el RDF, ha sido llamado CC/PP *(Composite CapabilitylPreference Profile)* [Kly04] . En un principio, permitirá que los teléfonos móviles y otros clientes no estándar de la Web describan sus características de modo que el contenido de la Web pueda ajustarse para ellos sobre la marcha.

3.2 Arquitectura de la Web Semántica

La Web Semántica está surgiendo como una evolución de la web actual a la que se agrega una estructura para captar el significado de los contenidos de las páginas y proporcionar un ambiente donde las aplicaciones puedan procesar y relacionar contenidos provenientes de distintas fuentes.

En la propuesta de desarrollo de la Web Semántica del consorcio W3C [W3Csw] se sugiere una arquitectura básica en capas (ver Fig. 3.1), comenzando por la capa de nivel inferior XML (eXtended Markup Language) que permite estructurar sintácticamente los datos, siguiendo por una capa RDF que define la semántica de dichos datos, luego por la capa ontológica que define consensuadamente conceptos y relaciones para distintos dominios y por último, la capa lógica, que define las reglas

lógicas y mecanismos para hacer inferencias. A continuación explicamos con más de detalle la función de cada una de las capas.

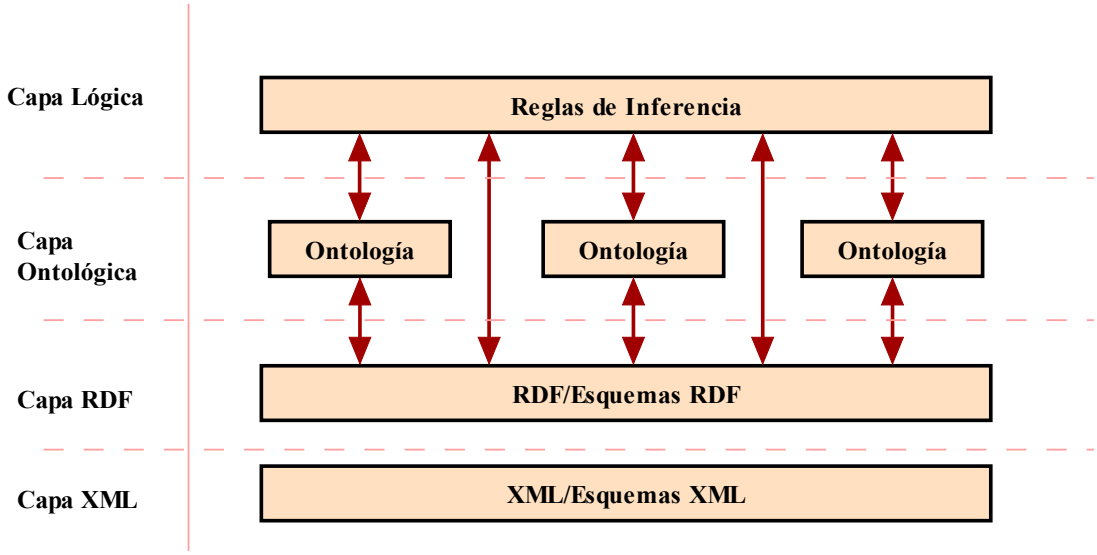

Figura 3.1 Arquitectura básica para el enfoque semántico.

3.3 La Capa XML y Esquemas XML (XMLS)

Las páginas web tradicionales contienen datos no estructurados que si bien son fáciles de interpretar por los seres humanos no pueden ser procesados por máquinas debido a la falta de estructura sintáctica. Cada vez son más frecuentes las páginas web que contienen tablas, estadísticas, formularios repletos de datos que proceden de alguna fuente de información estructurada, sin embargo, estos datos al ser convertidos en texto **HTML** sin estructura lógica alguna, pierden toda su capacidad operativa. El lenguaje XML (eXtensible Markup Language) fue creado en 1998 por el W3C y soluciona este problema permitiendo describir la estructura de los contenidos de datos que se encuentran en las páginas Web. Utilizando este lenguaje es posible definir arbitrariamente una estructura para un documento, aún cuando no exista ninguna información sobre el significado de dicha estructura. El lenguaje **XML** permite incluir *metadatos* (datos que describen datos) dentro de los documentos web de modo que las computadoras del usuario, puedan realizar tareas de manipulación de la información.

Por ejemplo un documento web que muestre información de una métrica podría ser codificado en HTML como se muestra en la figura 3.2. El texto mostrado que es fácilmente entendible por un ser humano no puede ser procesado automáticamente, por no tener los datos estructurados de alguna manera consistente. Es decir, ante el cambio de posición de un campo, el procesamiento seria inválido.

```
<html>
<body>
<p><u>MÉTRICA</u> : <b>DE</b></p>
<p>Errores por módulo</p>
<p>Autor: M. Martín </p>
<p>Fecha: 2003-05-01 </p>
<p>Fórmula: #Errores /#Módulos </p>
<p>Escala: Numérica </p>
<p>Unidad: E/MOD </p>
</body>
</html>
```

Figura 3.2 Ejemplo de código HTML para mostrar información de métrica.

El lenguaje XML, en cambio, permite incluir *metadatos* (datos que describen datos) dentro de los documentos web de modo que las computadoras pueden realizar tareas de manipulación de la información. En la figura 3.3 se muestra el código XML de un documento web que muestra la misma información de métrica del ejemplo anterior, en él se puede observar que los datos se disponen de una manera estructurada, a través de etiquetas que describen en forma precisa la sintaxis de cada campo de datos.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
    href="http://gidis.edu.ar/MetricasIndicadores.xsl"?>

<Métrica>
  xmlns=http://gidis.edu.ar/MetricasIndicadores.xsd>
  xmlns:dc="http://purl.org/dc#"

   <Identificador>DE</Identificador>
   <Nombre> Errores por módulo </Nombre>
   <dc:Creator> M. Martín < /dc:Creator >
   <dc:Date>2003-05-01< /dc:Date >

   <formula> #Errores /#Módulos </formula>
   <escala>
       <tipoDeEscala> Numérica </tipoDeEscala>
       <unidad> E/MOD </unidad>
   </escala>

</Métrica>
```

Figura 3.3 Ejemplo de código XML para mostrar información de métrica.

En el ejemplo, la etiqueta <?xml-stylesheet> permite referenciar un archivo externo que contiene la especificación de la forma de presentación de los datos en pantalla a través de un lenguaje estándar relacionado con XML, el Extensible Stylesheet

Language (XSL). Este lenguaje de transformación sirve para traducir documentos **XML**, en documentos HTML bien formados, teniendo en cuenta su presentación (tipo de letra, color, formato, etc. para cada campo de datos), permitiendo de esta manera separar la especificación de la presentación de los datos propiamente dichos.

El lenguaje **Esquema XML (XMLS)** permite definir qué elementos puede contener un documento **XML**, cómo están estructurados, qué atributos y de qué tipo pueden tener esos elementos. Es decir XMLS permite definir las etiquetas de marcado para estructurar los datos XML, por lo tanto es un *metalenguaje* ya que nos permite definir nuevos lenguajes XML adecuados para su uso en distintos dominios de una manera flexible y extensible. En el ejemplo anterior, las etiquetas <Métrica>, <escala>, etc. se han definido de esta manera, en un esquema XML que conforma un documento externo.

Este esquema XML es un documento que puede ser accedido a través de una dirección Web, identificada por un URI (Uniform Resource Identifier), y define un espacio de nombres para el dominio. De esta forma se puede disponer de un sitio común de validación para todo aquel que desee utilizar este lenguaje para su propio trabajo. En un documento Web (con XML, XMLS, RDF, etc) es posible usar etiquetas definidas en distintos espacios de nombres correspondientes a diferentes dominios. En el ejemplo de la figura 3.3, se puede observar que con el atributo **xmlns** se especifican los espacios de nombres para distintos dominios, en este caso el de métricas (las etiquetas Métrica, descripción, fórmula, etc) y el de documentos bibliográficos (las etiquetas Creator y date), que están definidos en archivos XML separados, "http://gidis.edu.ar/ MetricasIndicadores.xsd" y "http://purl.org/dc#" respectivamente, constituyendo vocabularios distintos para cada aplicación.

Los estándares XML y XMLS aportan mucha potencia y flexibilidad a las aplicaciones basadas en la Web, proporcionando numerosas ventajas a los programadores y usuarios, a saber:

- *Búsquedas con más significado.* Los datos se pueden etiquetar de forma exclusiva con distintos espacios de nombres en XML, lo que permite que un cliente especifique, por ejemplo, que desea buscar descripciones de métricas desarrolladas en el año 2003. Las búsquedas que utilizan los métodos actuales, por palabras calves, probablemente mezclarían todos los documentos (relacionados a métricas o no) del año 2003, con todos los documentos que se refieran a métricas. Además, con **XML** los métricas se podrían clasificar fácilmente en categorías, por ejemplo por autor, atributos, escala u otros criterios.

- *Integración de datos procedentes de distintas plataformas.* La capacidad de integrar varias bases de datos no compatibles entre sí era muy difícil. **XML** permite combinar fácilmente los datos estructurados procedentes de fuentes distintas, si usan el

mismo esquema XML. Se pueden utilizar agentes de software para integrar los datos en un servidor de nivel medio desde bases de datos de fondo (back-end) y otras aplicaciones. A continuación, dichos datos se pueden entregar a clientes u otros servidores para su agregación, procesamiento y distribución.

La característica de extensibilidad y flexibilidad de **XML** permite describir los datos contenidos en una gran variedad de aplicaciones muy diversas, desde las recopilaciones descriptivas de páginas Web hasta los registros de datos de una base de datos. Además, dado que los datos especificados en **XML** son auto-descriptivos, se pueden intercambiar y procesar sin necesidad de descripciones adicionales.

En resumen, XML provee una estructura sintáctica a la información contenida en documentos web permitiendo que dicha información pueda ser procesada tanto por agentes humanos como computarizados, permite separar el contenido de un documento de su presentación y, además, establece un estándar para compartir datos provenientes de distintas fuentes, facilitando la interoperabilidad entre aplicaciones. Se puede ver de este modo que **XML** constituye la capa más baja dentro de la arquitectura de las aplicaciones (ver figura 3.4), sobre la que se puede montar cualquier estructura de tratamiento de datos, hasta llegar a la presentación.

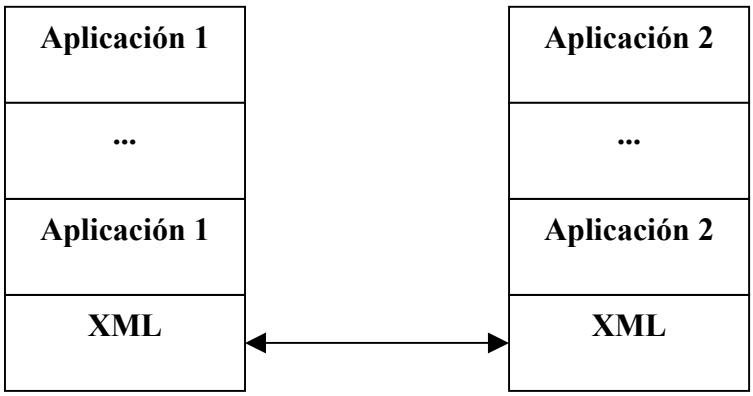

Figura 3.4 Arquitectura de aplicaciones que usan XML como intercambio de datos.

Si bien XML permite definir datos de una forma estructurada, de manera que puedan ser compartidos y procesados automáticamente, no permite especificar la semántica de dicha estructura, de una manera formal que pueda ser interpretada por una computadora. Por ejemplo, si analizamos el código XML presentado en la figura 3.3, sólo el usuario humano entiende el significado de expresiones tales como tipoDeEscala, Métrica, etc. XML se limita a describir la estructura de la información, la sintaxis, pero no su semántica. De acuerdo al diccionario de filosofía [Ang81], "semántica" es una disciplina que estudia cómo los símbolos se refieren a otros objetos. Es decir, el significado de los símbolos (intención) se define en términos de los objetos a los cuáles

él hace referencia (extensión). Si bien se suele decir que XML es un lenguaje para representar significado, no representa semántica en el sentido que acabamos de describir, tal como la etiqueta <TipoDeEscala> del ejemplo 3.2; no hace referencia a ningún otro objeto como tal, sin embargo, la relación con otros conceptos es concebida en la mente del lector humano.

3.4 La Capa RDF y Esquemas RDF (RDFS)

Actualmente, el lenguaje recomendado por el W3C para representar semántica en la nueva generación de la web es en este momento el RDF (Resource Description Framework) [Hay04, Las99, Man02]. Mientras que XML permite definir datos de una forma estructurada, RDF permite definir la semántica de esos datos.

En las siguientes secciones se describirá con más detalle los aspectos básicos de RDF y RDFS mostrando sus principales características a través de ejemplos de aplicación al dominio de estudio de esta tesis: métricas y mediciones.

3.4.1 El Modelo Básico RDF

El modelo RDF se compone de tres conceptos principales:

Recursos: Todas las cosas descriptas por expresiones RDF se denominan recursos. Los recursos, representan cualquier entidad (lugares, personas, objetos) del mundo real y están identificados por un URI (Universal Resource Identificator). Un recurso puede ser una página Web, una parte de una página Web; p. ej. un elemento HTML o XML específico dentro del documento fuente, o un objeto que no sea directamente accesible vía Web, p. ej. un libro impreso.

Propiedades: Una propiedad es un aspecto específico, característica, atributo, o relación utilizado para describir un recurso. Cada propiedad tiene un significado específico, define sus valores permitidos, los tipos de recursos que puede describir, y sus relaciones con otras propiedades. Las características de las propiedades se definen con RDFS (RDF Schema) que explicaré más adelante.

Sentencias (declaraciones, enunciados): Un recurso específico junto con una propiedad denominada, más el valor de dicha propiedad para ese recurso es una sentencia RDF. Estas tres partes individuales de una sentencia se denominan, respectivamente, sujeto, predicado y objeto. Una propiedad es a su vez un recurso. El objeto es el valor asignado a dicha propiedad, y puede ser una cadena simple de caracteres (string), u otro recurso, es decir, un recurso especificado por un URI.

Las sentencias constituyen la construcción básica que establece el modelo de datos RDF. Es decir, el significado de los datos se expresa mediante un conjunto de sentencias RDF que formalmente son representadas por tri-uplas (sujeto, predicado, objeto). Otra forma de representar las declaraciones (sentencias) RDF es a través de un grafo rotulado como se muestra en la Fig. 3.5.

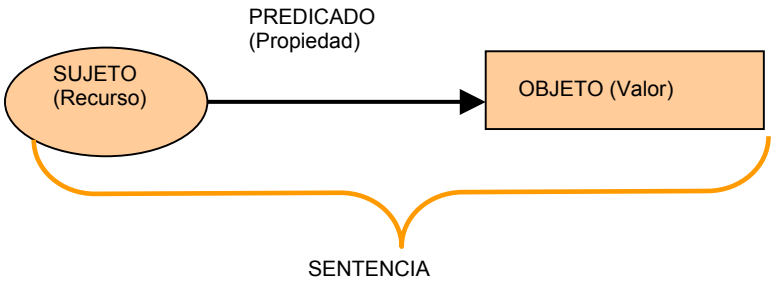

Figura 3.5 Representación gráfica de las sentencias RDF.

Por ejemplo, la figura 3.6 ilustra la construcción de dos sentencias que describen un **recurso**, en este caso el recurso es una métrica que está definida en el URI http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo. Las dos sentencias del ejemplo son:

"El 20-04-2003 fue catalogada 'http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo'"
"El atributo *Densidad De Errores* de URI http://gidis.edu.ar/DatosMetricas# DensidadDeErrores es cuantificado por la métrica http://gidis.edu.ar/DatosMetricas# ErroresPorMódulo".

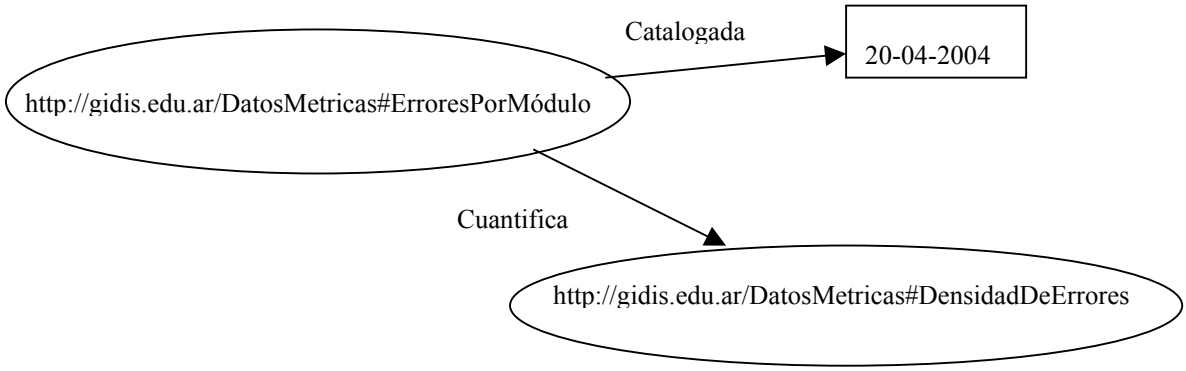

Figura 3.6 Representación gráfica las sentencias RDF.

En estos gráficos (también denominados "diagramas de nodos y arcos"), los nodos (dibujados como óvalos) representan recursos y los arcos representan propiedades

etiquetadas. Los nodos que representan cadenas de literales pueden dibujarse como rectángulos.

La dirección de la flecha es importante. El arco siempre empieza en el sujeto y apunta hacia el objeto de la sentencia. En este diagrama simple la sentencia:

"El atributo *Densidad De Errores* de URI http://gidis.edu.ar/DatosMetricas# DensidadDeErrores es cuantificado por la métrica http://gidis.edu.ar/DatosMetricas# ErroresPorMódulo". También puede interpretarse o leerse "la métrica http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo cuantifica al atributo http:// gidis.edu.ar/DatosMetricas#DensidadDeErrores", o en general "<sujeto> <predicado> <objeto>".

La tabla 3.1 muestra las partes correspondientes a las sentencias mostradas en la Fig. 3.6. Las sentencias representadas formalmente por triuplas (predicado, sujeto, objeto), le confieren al modelo de datos de RDF la cualidad de ser entendido tanto por seres humanos, como por máquinas, ya que tienen acceso a la representación formal de conjunto de triuplas de este modelo.

Predicado (Propiedad)	Sujeto (Recurso)	Objeto (Valor)
Catalogada	http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo	20-04-2004
Cuantifica	http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo	http://gidis.edu.ar/DatosMetricas#DensidadDeErrores

Tabla 3.1 Representación mediante triuplas de las sentencias RDF.

3.4.2 XML como Lenguaje de Especificación de la Sintaxis RDF

Uno de los principales aspectos que han contribuido al éxito de la tecnología RDF en el contexto de la web, es la posibilidad de poder representar e intercambiar datos (sentencias) RDF usando XML [Bra98]. Además toda la sintaxis RDF está definida en XMLS en el documento http://www.w3.org/1999/02/22-rdf-syntax-ns.

Sin embargo, es importante remarcar que el modelo de datos RDF no es serializado en un árbol sintáctico XML, donde se debe respetar la jeraquía especificada por el esquema XML relacionado. El modelo de datos RDF (a diferencia del modelo de árbol XML) es un grafo dirigido, lo que nos permite una mayor potencia para representar sentencias (podemos aseverar cualquier cosa sobre cualquier objeto), y esto es totalmente independiente de su representación serializada XML.

La figura 3.7 muestra cómo las sentencias de la tabla 3.1 pueden ser serializadas usando XML. La segunda línea del código indica el trozo que usa sintaxis RDF, e identifica con los prefijos "rdf" y "m" la ubicación de los documentos que contienen las definiciones de los elementos usados, es decir los espacios de nombres. Las demás líneas representan una descripción RDF del recurso identificado por el URI http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo, que en este caso representa una métrica, la marca <m:catalogada> indica que la métrica que se está describiendo tiene una propiedad llamada "catalogada" cuyo valor es "20/04/2004" y cuya semántica está definida en el vocabulario asociado al prefijo m:. Además, la marca <m:cuantifica> indica que la métrica también tiene una propiedad llamada "cuantifica" cuyo valor es el objeto (que en este caso es un recurso, no un literal) identificado por el URI "http://gidis.edu.ar/DatosMetricas#DensidadDeErrores".

```
<?xml version="1.0" >
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:m="http://gidis.edu.ar/Esquema#">
  <rdf:Descritpion about="http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo">
    <m:catalogada>20/04/2004</m:catalogada>
    <m:cuantifica
rdf:resource="http://gidis.edu.ar/DatosMetricas#DensidadDeErrores"/>
  </rdf:Descritpion>
</rdf:RDF>
```

Figura 3.7 Serialización en XML de sentencias RDF.

Cuando escribimos una frase en lenguaje natural utilizamos palabras que, en lo posible, encierran la intención de transmitir un significado inequívoco. En el caso de descripciones RDF se debe establecer el significado preciso de cada propiedad para que el procesamiento automático de las sentencias dé el resultado esperado. Es muy importante que tanto el escritor como el lector de una sentencia (declaración) entiendan el mismo significado para los términos utilizados, tales como **catalogada, cuantifica,** etc.

En RDF el significado se expresa a través de un esquema. Podemos pensar en un esquema como en una especie de diccionario. Un esquema define los términos que se utilizarán en una declaración RDF y le otorgará significados específicos. Para evitar confusiones entre definiciones independientes (y posiblemente conflictivas) del mismo término, RDF utiliza la facilidad de los *namespace* de XML. Los namespaces ("espacios de nombre") son simplemente una forma de asociar el uso específico de una palabra en el contexto del diccionario (esquema) en que se puede encontrar la definición. En RDF, cada predicado utilizado en una sentencia debe ser identificado con un sólo namespace, o esquema. Sin embargo una descripción de recurso puede contener sentencias con

predicados de varios esquemas. En la sección 3.4.6 haremos una descripción más detallada de espacios de nombres y esquemas RDF.

3.4.3 Definición de Tipos

El modelo RDF prevé algunas primitivas importantes para una mejor descripción de los recursos, de ellas podemos destacar la primitiva rdf:type, para definir tipos (instancias) de recursos, y las primitivas para definir contenedores, que explicaremos en la siguiente sección.

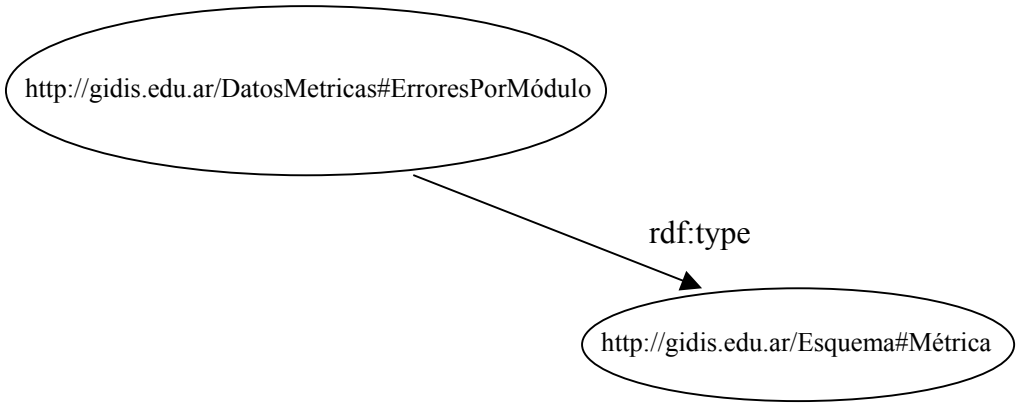

Figura 3.8 Definición de tipo en RDF.

Con la primitiva **rdf:type**, se puede indicar que un determinado recurso es de cierto tipo. La figura 3.8 ejemplifica dicha primitiva. En ella se especifica que el recurso http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo es de tipo métrica.

```
<?xml version="1.0" >
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:m="http://gidis.edu.ar/Esquema#">
  <rdf:Descritpion about="http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo">
    <rdf:type rdf:resource="http://gidis.edu.ar/Esquema#Métrica"/>
    <m:catalogada>20/04/2004</m:catalogada>
    <m:cuantifica
rdf:resource="http://gidis.edu.ar/DatosMetricas#DensidadDeErrores”/>
  </rdf:Descritpion>
</rdf:RDF>
```

Figura 3.9 Definición de tipo en RDF/XML.

De hecho rdf:type indica una relación binaria entre dos elementos, estableciendo un mecanismo de instanciación, o sea especifica que un elemento es una instancia de

otro. De esta manera se incluyen en una misma descripción datos y metadatos. En la figura 3.9 se muestra la correspondiente descripción en XML del ejemplo de la figura 3.8.

3.4.4 Contenedores en RDF

Muchas veces es necesario expresar que un recurso está relacionado con una colección de objetos, por ejemplo, para expresar que una métrica fue desarrollada por más de una persona, o para enumerar las distintas métricas que cuantifican un atributo de una entidad. Los contenedores RDF se usan para mantener tales listas de recursos o literales. RDF define tres tipos de objetos contenedores:

Bag: Una lista desordenada de recursos o literales. Los Bags (bolsa en español), se utilizan para indicar que una propiedad tiene múltiples valores y que no es significativo el orden en que se den tales valores.

Sequence: Una lista ordenada de recursos o literales. Sequence se usa para manifestar que una propiedad tiene múltiples valores y que el orden de los valores es significativo.

Alternative: Una lista de recursos o literales que representan alternativas para un valor (individual) de una propiedad. Una aplicación que utiliza una propiedad cuyo valor es una colección alternativa, sabe que se puede elegir como correcto uno cualquiera de los ítems en la lista.

Las definiciones de Bag y Sequence permiten explícitamente duplicar valores. RDF no define un concepto puntual de Set (conjunto), que podría ser un Bag sin duplicados, porque la esencia de RDF no impone un mecanismo de validación en el caso de que se violen estas restricciones.

Para representar colecciones de recursos, RDF utiliza un recurso adicional que identifica la colección específica (una instancia de uno de los tipos objetos contenedores definidos arriba). La propiedad *type,* se utiliza para hacer esta declaración. La relación de pertenencia entre estos recursos contenedores y los recursos que pertenecen a la colección se establece a través de un conjunto de propiedades de filiación que se denominan simplemente "_1", "_2", "_3", etc. Los recursos contenedores pueden tener otras propiedades añadidas a las propiedades de pertenencia, y la propiedad *type*. Cualquiera de tales sentencias adicionales describe al contenedor.

Por ejemplo en el modelo RDF de la figura 3.10, se representa la sentencia: “Los valores de mediciones tomadas para la métrica ErroresPorMódulo son 0.74, 1.05, y 0.8”.

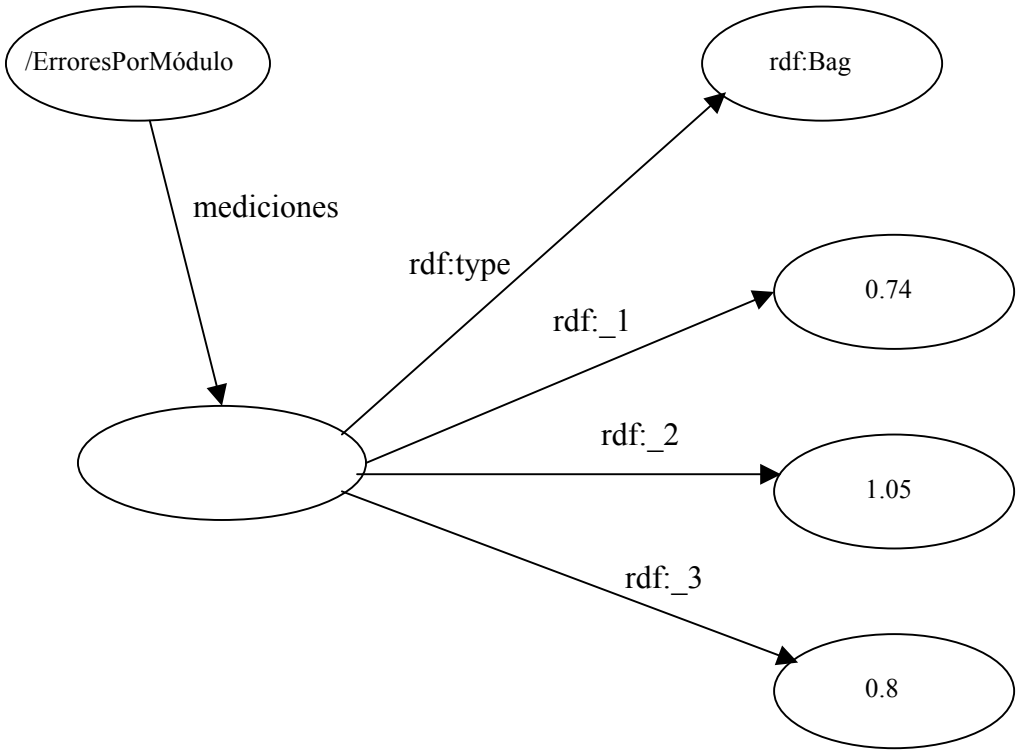

Figura 3.10 Ejemplo de uso del contenedor Bag

Los contenedores no son equivalentes a propiedades repetidas del mismo tipo, en la figura 3.11 se muestra la serialización RDF/XML del ejemplo anterior. Nótese como RDF/XML utiliza **li** como elemento de conveniencia para evitar que tenga cada miembro un número explícito. El elemento li asigna las propiedades _1, _2, y así siguiendo si fuera necesario.

```
<?xml version="1.0" >
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:m="http://gidis.edu.ar/DatosMetricas#">
  <rdf:Descritpion about="http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo">
    <m:mediciones>
      <rdf:Bag>
            <rdf:li> 0.74 </rdf:li>
            <rdf:li> 1.05 </rdf:li>
            <rdf:li> 0.8 </rdf:li>
      </rdf:Bag>
    </m:mediciones>
  </rdf:Descritpion>
</rdf:RDF>
```

Figura 3.11 Ejemplo de uso de contenedores Bag en RDF/XML.

3.4.5 Reificación en RDF (sentencias sobre sentencias)

Además de permitir sentencias sobre los recursos Web, RDF puede usarse para crear sentencias sobre otras declaraciones RDF; nos referiremos a éstas como sentencias de alto nivel. Para realizar declaraciones sobre otras declaraciones, tenemos que construir un modelo de la sentencia original; este modelo es un nuevo recurso al que podemos anexar propiedades adicionales. Esta característica de RDF de poder considerar una sentencia como un recurso permite anidar declaraciones, y a este proceso se lo denomina formalmente "*reification" (transformación de algo abstracto en concreto)* en el contexto de la representación del conocimiento.

Formalmente, una reificación en RDF significa expresar una sentencia como un recurso con cuatro propiedades:

subject: La propiedad subject identifica el recurso que describe la sentencia modelada; es decir, el valor de la propiedad subject es el recurso sobre el cual se hace la sentencia original.

predicate: La propiedad predicate identifica la propiedad original en la declaración modelada. El valor de la propiedad predicate representa la propiedad específica en la sentencia original.

object: La propiedad object identifica el valor de la propiedad en la sentencia modelada. El valor de la propiedad object es el objeto en la sentencia original.

type: El valor de la propiedad type describe el tipo del nuevo recurso. Todas las sentencias transformadas reificadas son instancias de rdf:statement; es decir tienen una propiedad type cuyo objeto es rdf:statement.

Un nuevo recurso con las cuatro propiedades anteriores representa la sentencia original y puede usarse como objeto de otras declaraciones.

Para mostrar un ejemplo, supongamos que queremos expresar la siguiente sentencia: en el URL http://gidis.edu.ar/Documenos/Metricas está documentada la métrica http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo que fue catalogada el 20-04-2004. Podemos reificar la sentencia S, representada con la tri-upla (Catalogada, [http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo], “20-04-2004”). Esta reificación dará origen a un nuevo recurso descripto por el siguiente conjunto de sentencias:

(rdf:type, [R], [rdf:statement])
(rdf:subject, [R], [http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo])
(rdf:predicate, [R], [m:Catalogada])
(rdf:object, [R], “20/04/2004”)

La figura 3.12 muestra la representación de esta reificación en un grafo

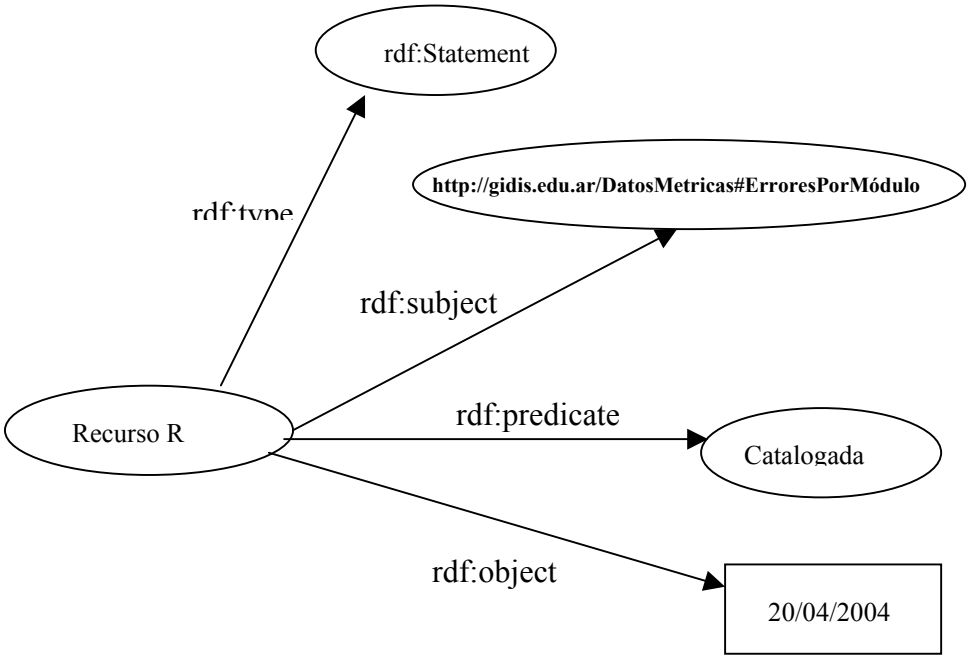

Figura 3.12 Reificación de una sentencia RDF.

Después de una reificación, es posible hacer una aserción sobre la sentencia, ya que ésta se transforma en un nuevo recurso, en nuestro ejemplo, si queremos afirmar sobre la sentencia S, que esta documentada en el URL http://gidis.edu.ar/ Documenos/Metricas, agregamos la propiedad m:documentada al recurso generado con la reificación, como se muestra en la figura 3.13.

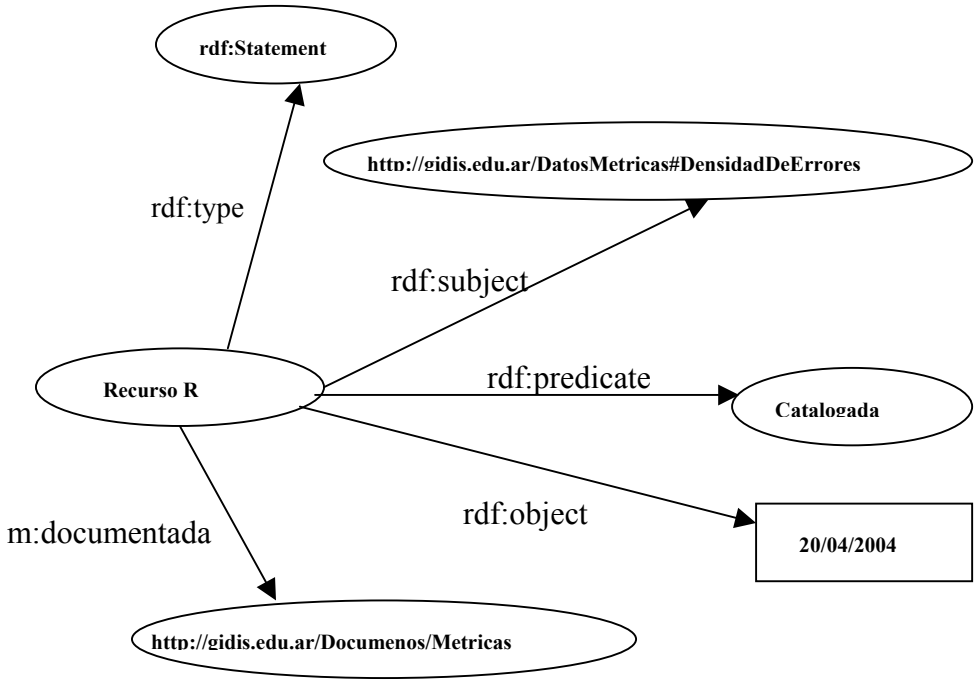

Figura 3.13 Sentencia sobre una sentencia RDF.

En la figura 3.14 se muestra la serializacion en XML de la sentencia S correspondiente al ejemplo anterior.

```
<?xml version="1.0" >
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:m="http://gidis.edu.ar/Esquema#">
  <rdf:Descritpion >
   <rdf:subject
rdf:resource="http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo"/>
   <rdf:predicate rdf:resource= http://gidis.edu.ar/DatosMetricas#Catalogada/>
   <rdf:object>20/04/2004</rdf:object>
   <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Statement"/>
   <m:documentada rdf:resource="http://gidis.edu.ar/Documenos/Metricas"/>
  </rdf:Description>
</rdf:RDF>
```

Figura 3.14 Ejemplo de Reificación codificada en RDF/XML.

3.4.6 RDF Shema

RDF no ofrece ningún mecanismo para especificar el vocabulario usado en las descripciones (sentencias) tal como el término "documentada" usado en el ejemplo anterior. RDF Schema (RDFS) [Bri00], es el lenguaje que completa el modelo RDF básico en la función de alcanzar interoperabilidad semántica en el contexto de la web. Esta propuesta del W3C provee un sistema de tipos básicos conjuntamente con un mecanismo para definir nuevos tipos, conformando distintos espacios de nombres, permitiendo de esta manera, que comunidades de usuarios de distintos dominios puedan compartir sus propios vocabularios. RDFS es una extensión de RDF que permite:

- Describir los conceptos usados en aplicaciones RDF
- Especificar restricciones de tipos para los sujetos y objetos de las triuplas.

RDFS se puede ver como una forma de modelado orientado a objetos (OO) en la web. Con los atributos predefinidos rdfs:Class y rdfs:subClass es posible definir una jerarquía de clases, y las instancias pueden referirse a ellas a través del atributo rdf:type de RDF.

El mecanismo para la definición de tipos en un esquema RDF es ligeramente diferente a la definición de tipos de las metodologías tradicionales de modelado OO. En ésta última, la principal preocupación es la identificación de las entidades que serán representadas como clases, subclases y sus correspondientes atributos; en RDFS, en cambio, las propiedades (atributos) son definidas en términos de las clases de recursos a las cuales ellas se aplican. Este enfoque centrado en las propiedades facilita la

descripción semántica de los recursos existentes en la web (enumerando sus propiedades y valores asociados), principal objetivo de la arquitectura RDF.

A continuación se describen las principales primitivas de modelado RDFS que están agrupadas en clases, propiedades y restricciones para facilitar su explicación. La figura 3.15 (extraída del reporte técnico del W3C [Bri00]), muestra la relación existente entre estas primitivas y las primitivas del modelo RDF básico, sobre el cual está construido RDFS.

En dicha figura, los rectángulos con puntas redondeadas indican clases y las flechas señalan cuál es la clase que define al recurso. La relación subclase se indica con un rectángulo (la superclase) que contiene a otro rectángulo (la subclase). Por último, se muestra que todo objeto en RDF es un recurso (*Resource*).

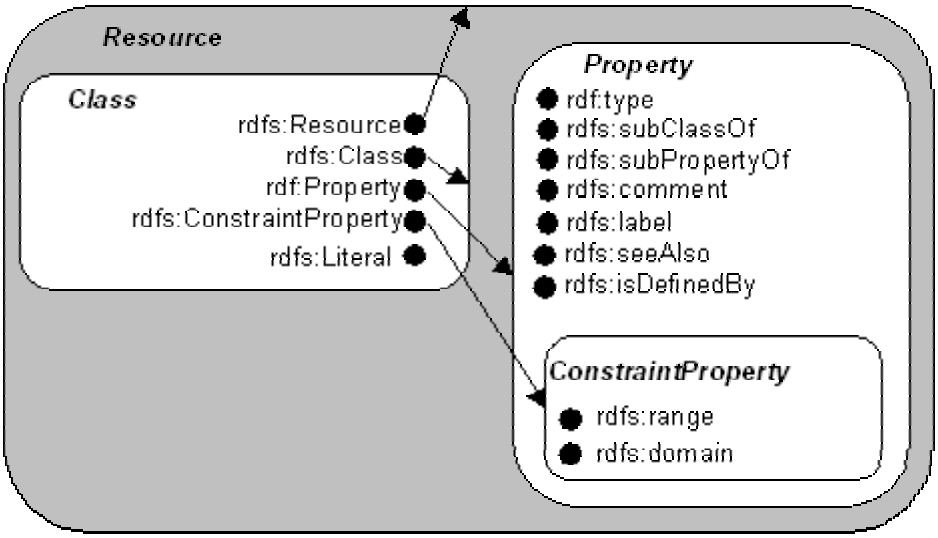

Figura 3.15 Jerarquía de clases del modelo RDFS [Bri00].

3.4.6.1 Clases

RDFS permite la definición de subclases que heredan las definiciones de una o más clases ascendentes, permitiendo la implementación de herencia múltiple. Esta característica incorpora gran flexibilidad al modelo RDF, porque las clases que se están definiendo pueden heredar de clases ya definidas en esquemas previos, especializando los metadatos, promoviendo de esta manera compartir y reusar estos esquemas. Para este propósito, las principales primitivas que ofrece RDFS son:

rdfs:Resource - representa la clase genérica en el modelo RDF Schema. Todo objeto descrito por expresiones RDF es un recurso.

rdfs:Class – es subclase de *rdfs:Resource* y representa el concepto genérico de tipo o categoría, similar a la noción de clase en orientación a objetos.

rdf:Property – es subclase de *rdfs:Resource* y representa un aspecto del recurso que se está describiendo, similar a la noción de atributo en orientación a objetos.

3.4.6.2 Propiedades

Las propiedades posibilitan expresar relaciones entre instancias y sus clases o clases y sus superclases. Relaciones entre propiedades también son permitidas, obteniéndose así una jerarquía de propiedades. Las principales propiedades disponibles en RDFS son:

rdf:type – es subclase de *rdf:Prorperty* y denota que un recurso es instancia de una clase, poseyendo todas sus características. Un recurso puede ser instancia de más de una clase.

rdfs:**subClassOf** - es subclase de *rdf:Property* y denota la relación de subclase/superclase entre dos clases. Esta propiedad es la base para la especificación de la herencia múltiple, y tiene característica de transitividad.

rdfs:subPropertyOf - es subclase de *rdf:Property* y denota la relación de especialización entre dos propiedades, posibilitando la definición de una jerarquía de propiedades.

3.4.6.3 Restricciones

Este mecanismo permite asociar restricciones a las propiedades de un recurso. Las principales primitivas de restricciones son:

rdfs:domain – es una instancia de la clase *rdfs:ConstraintProperty* y especifica a qué clase se aplica una propiedad.

rdfs:range – es una instancia de la clase *rdfs:ConstraintProperty* y restringe los valores (u objetos) que una propiedad puede asumir.

Para ilustrar mejor todos estos conceptos, en la siguiente figura (Fig. 3.16) se muestra un ejemplo de aplicación de esquemas RDF al dominio de métricas.

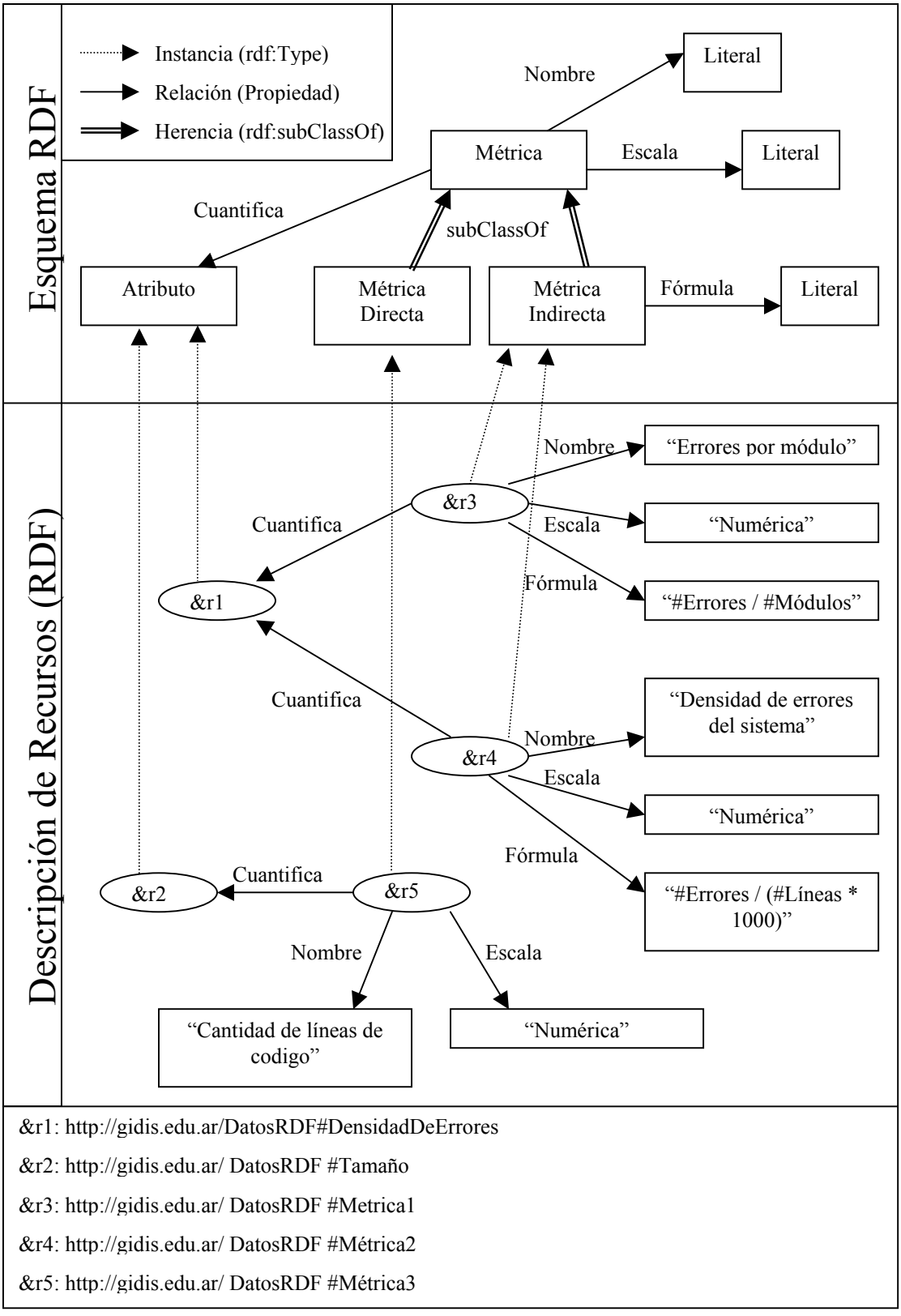

Figura 3.16 Un ejemplo de modelo RDFS para el dominio de Métricas y algunas instancias RDF.

La figura 3.16 muestra la definición del esquema RDF para el dominio de métricas (parte superior), y la descripción de algunas instancias de métricas en

sentencias RDF (parte inferior). El tipo de gráfico que se usó está basado en la propuesta de Staab et al [Sta00] que sugiere un diseño preliminar para modelar ontologías en RDF(S).

En la parte superior de la figura se ilustra cómo los conceptos y relaciones de un dominio se pueden modelar con una jerarquía de clases RDFS, y propiedades respectivamente. Los rectángulos representan clases en el dominio de métricas que se traducen en clases en RDFS, usando la primitiva **rdfs:Class** (ver código RDFS de la figura 3.17), excepto la clase **Literal ,** que es una clase predefinida del lenguaje RDFS.

Las flechas con línea simple representan relaciones en nuestro modelo conceptual, que se traducen en propiedades RDFS, usando la primitiva **rdf:Property** (ver código RDFS de la figura 3.17). Por último, las flechas de línea doble representan la relación de herencia, proporcionada por la propiedad predefinida **rdfs**:**subClassOf.**

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

<rdfs:Class rdf:ID="Métrica">
  <rdfs:label xml:lang="sp">Métrica</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="MétricaDirecta">
  <rdfs:label xml:lang="sp">MétricaDirecta</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Métrica"/>
</rdfs:Class>

<rdfs:Class rdf:ID="MétricaIndirecta">
  <rdfs:label xml:lang="sp">MétricaIndirecta</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Métrica"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Atributo">
  <rdfs:label xml:lang="sp">Atributo</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdf:Property rdf:ID="Cuantifica">
  <rdfs:label xml:lang="sp">Cuantifica</rdfs:label>
  <rdfs:domain rdf:resource="#Métrica"/>
  <rdfs:range rdf:resource="#Atributo"/>
</rdf:Property>

<rdf:Property rdf:ID="Nombre">
  <rdfs:label xml:lang="sp">Nombre</rdfs:label>
  <rdfs:domain rdf:resource="#Métrica"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>

<rdf:Property rdf:ID="Escala">
  <rdfs:label xml:lang="sp"> Escala </rdfs:label>
  <rdfs:domain rdf:resource="#Métrica"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>

<rdf:Property rdf:ID="Fórmula">
  <rdfs:label xml:lang="sp"> Fórmula </rdfs:label>
  <rdfs:domain rdf:resource="#Métrica"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>

</rdf:RDF>
```

Figura 3.17 Código del esquema RDF para el ejemplo de Métricas

En la figura 3.17 se muestra el código RDFS que implementa el esquema de nuestro ejemplo de métricas. En él se definen las clases y propiedades que darán origen a un nuevo vocabulario (espacio de nombres) para nuestro dominio; se muestra además la definición de restricciones sobre las relaciones, usando las propiedades **rdfs:domain** y **rdfs:range.** Esta especificación restringe los tipos de recursos que se pueden usar en las posiciones de objeto y valor en las triuplas RDF; la idea es similar a la de integridad de datos en los sistemas de bases de datos. Esta característica permite validar las sentencias RDF(S) no sólo desde el punto de vista de la sintaxis.

En la parte inferior de la figura 3.16 se muestran algunos datos de métricas instanciados del esquema RDFS, los óvalos representan recursos cuyos URI´s se describen al pie de la figura, y sus descripciones RDF se muestran en el código RDF/XML de la figura 3.18. Los rectángulos representan valores (en este caso del tipo string), las flechas de línea llena representan propiedades instanciadas del esquema de la parte superior y, por último, las flechas de líneas punteadas representan la relación de instanciación usando la primitiva **rdf:type** (ver fig. 3.18).

```
<?xml version="1.0" >
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:m="http://gidis.edu.ar/Esquema#">

  <rdf:Descritpion rdf:ID="DensidadDeErrores">
    <rdf:type rdf:resource="http://gidis.edu.ar/Esquema#Atributo"/>
  </rdf:Descritpion>

  <rdf:Descritpion rdf:ID="Tamaño">
    <rdf:type rdf:resource="http://gidis.edu.ar/Esquema#Atributo"/>
  </rdf:Descritpion>

  <rdf:Descritpion rdf:ID="Métrica1">
    <rdf:type rdf:resource="http://gidis.edu.ar/Esquema#MétricaIndirecta"/>
    <m:Nombre>Errores por módulo </m:Nombre>
    <m:Escala>Numérica</m:Escala>
    <m:Fórmula> #Errores / #Modulos </m:Fórmula>
  </rdf:Descritpion>

  <rdf:Descritpion rdf:ID="Métrica2">
   <rdf:type rdf:resource="http://gidis.edu.ar/Esquema#MétricaIndirecta"/>
    <m:Nombre>Densidad de errores del sistema </m:Nombre>
    <m:Escala>Numérica</m:Escala>
    <m:Fórmula> #Errores / (#Linas * 1000) </m:Fórmula>
  </rdf:Descritpion>

  <rdf:Descritpion rdf:ID="Métrica3">
    <rdf:type rdf:resource="http://gidis.edu.ar/Esquema#MétricaDirecta"/>
    <m:Nombre>Cantidad de líneas de código </m:Nombre>
    <m:Escala>Numérica</m:Escala>
  </rdf:Descritpion>

</rdf:RDF>
```

Figura 3.18 Serialización en RDF/XML de algunas instancias de Métricas.

En las definiciones RDFS las primitivas **Class**, **subClassOf**, etc., le dan a las sentencias RDF una semántica cuya interpretación es común al modelado orientado a objetos. Esta semántica nos dice cómo los conceptos se relacionan entre sí jerárquicamente y qué restricciones presentan sus atributos.

Con la ayuda del mecanismo de espacio de nombres XML, las descripciones RDF (ver figura 3.18) pueden usar vocabularios definidos en distintos esquemas, en nuestro ejemplo el esquema de la figura 3.17 está referenciado en el espacio de nombres de URI http://gidis.edu.ar/Esquema#. Un importante aspecto del modelo de datos RDF/RDFS es que permite combinar meta descripciones de distintas partes de la web en un único grafo.

Es importante observar que RDFS solamente permite modelar relaciones binarias entre recursos, además, no permite especificar restricciones de cardinalidad sobre las relaciones ni sus propiedades (simétrica, transitiva), tampoco se pueden especificar condiciones necesarias y suficientes para definir pertenencia a una clase. En resumen, RDFS carece de poder expresivo para representar restricciones lógicas; de ahí que sea necesario en la web semántica, una capa lógica por encima de RDFS (tal como lo ilustramos en la figura 3.1).

3.5 La Capa Ontológica

La utilización de URIs para describir las propiedades de los recursos en RDF garantiza la definición única de los conceptos que representan los datos de documentos en la web. Sin embargo, un mismo concepto se puede expresar con términos distintos en distintos sitios o aplicaciones Web.

Para que la información existente en distintos sitios y repositorios dispersos en la web se puedan procesar y relacionar conjuntamente, es necesario que exista una definición compartida y consensuada de los "términos" utilizados para representar los conceptos y sus relaciones contenidos en distintos documentos. Por ejemplo, si la fórmula asociada al cálculo de una métrica indirecta, es descripta en RDF con la propiedad <m:Fórmula> en algunos repositorios, y con la propiedad <m:Ecuación> en otros, las computadoras no podrán interpretar si se está tratando del mismo concepto.

La solución propuesta para la web semántica es el uso de ontologías. Para los investigadores en web semántica, una Ontología es un documento o archivo que define de una manera formal y consensuada los términos utilizados en un dominio y la relación entre estos términos [W3Csw].

Las ontologías más típicas para la Web constan de una taxonomía y de un conjunto de reglas de inferencia (axiomas). La taxonomía define clases de objetos y

relaciones entre ellos. Por ejemplo, una métrica indirecta puede definirse como un tipo de métrica, se puede establecer que las métricas directas solamente sean aplicables a atributos, y así sucesivamente.

Las reglas de inferencia integradas en las ontologías proporcionan una potencia mayor todavía. Una ontología puede expresar la regla "Si una entidad E, (por ejemplo producto de software) tiene un atributo A, y A puede ser medido por la métrica M, entonces M sirve para evaluar E". Un programa podría entonces deducir fácilmente qué métricas aplicar para evaluar una determinada entidad y realizar el proceso en forma automática. En realidad, la computadora no "comprende" nada de esta información, pero ahora sí es capaz de manipular los términos mucho más eficazmente de modo mucho más útil y significativo para usuarios humanos.

Provista la Web de ontologías, se avistan soluciones a los problemas terminológicos. El significado de los términos o marcas XML utilizados en una página de la Web puede ser definido mediante enlaces que vayan desde la página hasta un documento que especifique la ontología.

Además las ontologías pueden mejorar el funcionamiento de la Web de muchas formas. Pueden aplicarse de un modo sencillo para mejorar la precisión de las búsquedas: el programa de búsqueda puede limitar su consulta a las páginas que aludan a un concepto exacto, en lugar de presentar todas las obtenidas por medio de palabras clave ambiguas.

Por ejemplo, se podría querer recuperar de la web todas las fórmulas de cálculos de métricas de software. Con un buscador semántico, se podría buscar por el término **Fórmula** de la ontología definida para el ámbito de métricas de software, que tiene una semántica bien definida, y sólo recuperaría los URI de las fórmulas de dichas métricas. Los buscadores actuales en cambio, al buscar por palabra clave “fórmula”, pueden recuperar todos los documentos que tengan información de cualquier fórmula de cálculo, fórmula química, receta de cocina, etc.

Aplicaciones más avanzadas podrían utilizar las ontologías para relacionar la información de una página con las estructuras de conocimiento y las reglas de inferencia asociadas, potenciando el procesamiento automático de las bases de conocimientos dispersos en la web.

En el próximo capítulo, se hará una descripción detallada de los conceptos relacionados a ontologías, su clasificación, metodologías de diseño y tecnologías para su implementación en la web semántica.

Si bien las ontologías sencillas se pueden implementar en la web usando los lenguajes RDF/RDFS vistos (ya que permiten especificar conceptos, relaciones y

restricciones sobre dichas relaciones), para tener una real eficacia en el procesamiento automático de la semántica de los datos, se necesita poder especificar las restricciones lógicas (axiomas) y reglas de inferencia, de manera que agentes automáticos puedan deducir nuevas sentencias a partir de ontologías y reglas consistentemente especificadas. De ahí la importancia de adicionar una capa lógica por encima de la ontológica, para proveer a la web semántica de capacidad de razonamiento sobre sus datos y metadatos.

3.6 La Capa Lógica

La capa lógica está compuesta por un conjunto de axiomas y reglas de inferencia que los agentes (computacionales o humanos) podrán utilizar para relacionar y procesar la información. Estas reglas ofrecen el poder de deducir nuevas sentencias a partir de los datos y estructuras que están descritos en las capas XML y RDF, usando además las relaciones entre esos datos y estructuras definidas en la capa Ontológica.

La expresividad de RDF y RDFS para modelar ontologías completas es muy limitada: en el caso de RDF se limita (en líneas generales) a representar predicados binarios, y RDFS (también en líneas generales) está limitado a jerarquía de clases, jerarquía de propiedades con definición de restricciones de dominio y rango de dichas propiedades. RDF/RDFS carece de soporte para tipos de datos primitivos, carece de poder expresivo para representar axiomas (no hay negación, implicación, cardinalidad, cuantificación), no es posible definir propiedades de propiedades (transitividad, simetría, etc.), no permite especificar condiciones necesarias y suficientes para establecer la pertenencia a una clase.

Sin embargo, el grupo de trabajo de ontologías en la web del W3C [WebOnt], ha especificado recientemente el leguaje OWL [Bech04] (Ontology Web Language), que adiciona mayor expresividad a RDF y RDFS con una semántica formal basada en la lógica descriptiva. Este es el lenguaje recomendado por el W3C para la definición de ontologías en la web. "*RDF y OWL sientan las bases para aplicaciones de Web Semántica*" dijo Tim Berners-Lee, director del W3C e inventor de la World Wide Web [W3C04].

Anteriormente, dos grupos de investigación uno de América y otro de Europa, ya habían identificado la necesidad de un lenguaje de modeldo de ontologías más potente que RDF/RDFS. Éstos dos grupos dieron lugar a una iniciativa común que dio origen al lenguaje llamado DAML+OIL [DamlOil] (el nombre se origina como la unión de la propuesta americana DAML-ONT [Ste00] y el lenguaje europeo OIL [Oil]). DAML+OIL fue considerado como el punto de partida por el grupo de trabajo de ontologías del W3C para la definición del lenguaje OWL.

OWL está especificado sobre RDF y RDF Esquema y añade importantes primitivas para la descripción de clases y propiedades: entre otras, relaciones entre clases (p.e. complemento de, disjunta a), cardinalidad de propiedades (p.e. mínimo dos, exactamente uno), igualdad de clases, mayor riqueza de tipos en las propiedades, características de propiedades (p.e. simetría, transitividad), y clases enumeradas.

El lenguaje OWL es el estándar propuesto para implementar ontologías en la web y especificar axiomas de la capa lógica, ya que nos permite describir la semántica del conocimiento de una forma procesable por la máquina. Conjuntamente con la especificación del lenguaje se provee una especificación formal de su semántica, de manera que se puede dar soporte de razonamiento sobre la capa ontológica a través de una traducción de OWL en lógica de predicado o lógica descriptiva (en el capítulo 5 mostraremos ejemplos de axiomas para el ámbito de métricas e indicadores).

Capítulo 4- Ontologías

"*Lo que nosotros observamos, no es la naturaleza en sí misma, sino la naturaleza expuesta a nuestro método de cuestionamiento*", Werner Heisenberg *, Físico y Filósofo*

4.1 Introducción

La vertiginosa evolución de la web como un medio de publicación de documentos, y las tecnologías de la información y comunicación relacionadas, han creado las condiciones propicias para difundir y compartir documentación científica e información catalogada.

Sin embargo, para que toda esta información pueda ser compartida, reusada, interpretada y aplicada uniformemente a distintos ámbitos, es necesario contar con una terminología común, y una estructura de metadatos (datos acerca de cómo están estructurados los datos), que permita tanto el procesamiento automatizado de la información, como su correcta interpretación.

Las ontologías son una solución a esta necesidad, ya que proveen los metadatos necesarios para una representación declarativa del conocimiento de un domino, que puede ser comunicado entre personas y computadoras; además, proporciona una definición formal de conceptos consensuados que asegura la interpretación correcta del conocimiento compartido y, por último, brinda un vocabulario común, bien definido para el intercambio de información en el dominio en cuestión.

Desde el comienzo de los noventa, las ontologías se volvieron un tema común de investigación para algunas comunidades del área de Inteligencia Artificial, incluyendo ingeniería del conocimiento, procesamiento de lenguaje natural y representación del conocimiento.

Recientemente la noción de ontología se ha extendido a otras áreas, tales como integración de información desde orígenes heterogéneos, búsqueda semántica de información en Internet y gestión del conocimiento organizacional. La razón de su difusión y potencial adopción está sustentada básicamente en todo lo que se promete: una comprensión común y compartida de algún dominio que puede ser comunicado entre personas y computadoras.

4.2 Definición

La definición más citada del término **Ontología** en el área de Inteligencia Artificial es la de Gruber [Gru95] "Una ontología es la especificación explícita de una conceptualización."

Para la Inteligencia Artificial lo que existe es exactamente aquello que puede ser representado computacionalmente. Cuando el conocimiento de un dominio es implementado en un formalismo declarativo, el conjunto de objetos que pueden ser representados es llamado el universo de discurso. Este conjunto de objetos, y las relaciones descriptibles entre ellos, se refleja en el vocabulario con el cual se representa el conocimiento.

En la especificación de una ontología se asocian mediante definiciones, los nombres de las entidades del ámbito en cuestión, (por ejemplo clases, relaciones, funciones u otros objetos), con cierto texto legible por el ser humano que las describe y con axiomas que restringen su interpretación.

Borst [Bor97] modificó ligeramente la definición de Gruber: "Las ontologías se definen como la especificación formal de una conceptualización compartida".

Studer, Benjamins, y Fensel [Stu98], agregaron expresividad a las definiciones de Gruber y Borst explicitando:

Conceptualización se refiere a un modelo abstracto de algún fenómeno en el mundo, proveniente de haber identificado los conceptos relevantes de dicho fenómeno.

Explícita se refiere a que los conceptos usados y las restricciones para su uso se definen explícitamente.

Formal se refiere al hecho de que la ontología debería ser legible o interpretable por una computadora.

Compartida refleja la noción de que una ontología captura conocimiento consensuado, es decir, no es conocimiento privado de un individuo, sino aceptado por un grupo o comunidad.

4.3 Componentes de una Ontología

Una ontología consta de un conjunto no vacío de conceptos identificados como entidades relevantes en el dominio a modelar, un conjunto de atributos que describen los conceptos y que pueden ser propios o heredados en una especialización, un conjunto de

relaciones entre dichos conceptos, un conjunto de funciones (que son un caso especial de relaciones), un conjunto de axiomas que vinculan elementos de la ontología en condiciones que deben cumplirse siempre y un conjunto de instancias. En las siguientes secciones pasaremos a describir cada uno de estos elementos.

4.3.1 Conceptos

Un concepto puede ser cualquier cosa acerca de la cual algo se pueda aseverar, y por tanto puede ser un objeto físico (tangible), u objetos intangibles como por ejemplo la descripción de una tarea, función, acción, estrategia, entre otros. Cada concepto tiene un término asociado como nombre y un conjunto de atributos que lo identifican.

En la figura 4.1 se muestra mediante un diagrama UML un pequeño ejemplo simple pero ilustrativo de ontología, donde los conceptos están representados con rectángulos que tienen en su parte superior el término asociado y en la parte inferior, la lista de sus atributos. Por ejemplo, en dicha figura se puede distinguir el concepto Métrica, cuyos atributos en la ontología presentada son nombre, interpretaciónValor, objetivo y referencias (en la sección 5.3.2.4 se describe con precisión los términos, atributos y relaciones del dominio de métricas e indicadores).

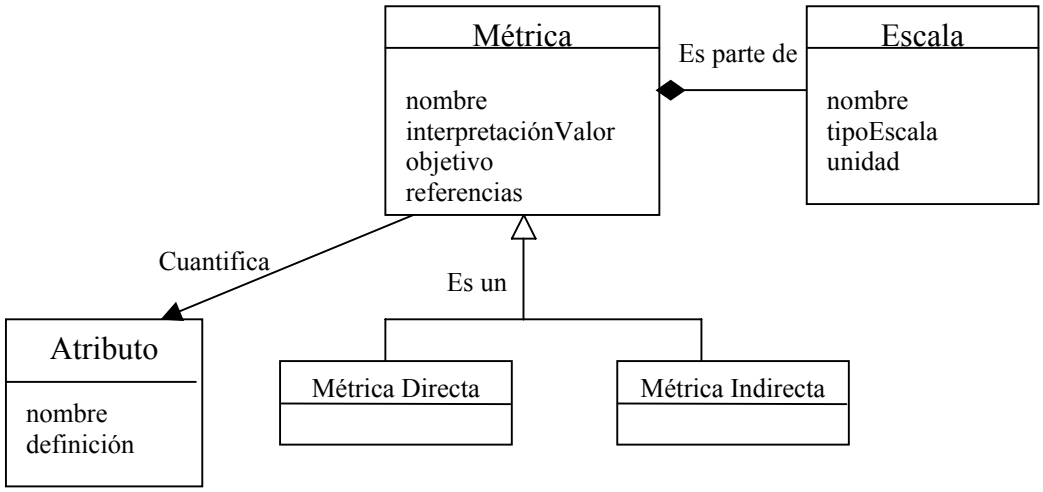

Figura 4.1 Conceptos, atributos y relaciones en una ontología.

4.3.2 Relaciones

Las relaciones representan el tipo de interacción entre los conceptos de un dominio, son formalmente definidas como subconjuntos del producto cartesiano de n conjuntos, esto es R: C1 x C2 x … x Cn. En la figura 4.1 las relaciones están

representadas con flechas, como por ejemplo la relación *cuantifica* que asocia cada métrica con el atributo al cual se aplica.

Algunas relaciones tienen un significado especial, a saber: las relaciones binarias de especialización (Es un) y de composición (Es parte de), que en el dibujo de la figura 4.1 están representadas con las convenciones establecidas por el modelo UML. En nuestro ejemplo, las métricas directas y las métricas indirectas son tipos de métricas, por lo tanto heredan todos los atributos del concepto métrica a través de la relación taxonómica (*Es un*). Por otro lado, una métrica tiene como uno de sus componentes una escala, en este caso la relación es del tipo parte-todo (*Es parte de*) .

En general los modelos ontológicos definen la relación taxonómica *Es un* como irreflexiva, transitiva y asimétrica. En tanto que la propiedad transitiva permite inferir una jerarquía (taxonomía) en la estructura, las propiedades restantes son útiles para chequear consistencia.

Para modelar la relación de que un conjunto de conceptos son partes que constituyen otro concepto, se usa la relación *Es parte de* (part-whole), que se suele definir con las propiedades irreflexiva y asimétrica.

4.3.3 Funciones

Son un caso especial de relaciones donde el enésimo elemento de la relación es único para los n-1 anteriores. Formalmente las funciones se definen como F: C1 x C2 x….x Cn-1 x Cn. Ejemplos de funciones son las relaciones Madre-de, o en el ejemplo de la figura 4.1 la relación *cuantifica* es una función ya que para cada métrica el atributo que evalúa es único.

4.3.4 Axiomas

Los axiomas se usan para modelar verdades que se cumplen siempre en la realidad modelada. Los axiomas definidos en una ontología pueden ser estructurales o no estructurales.

Un axioma estructural establece condiciones relacionadas a las jerarquías de la ontología, conceptos y atributos definidos. Un ejemplo de axioma estructural puede ser “Una métrica directa no puede ser a la vez métrica indirecta”. Es decir:

$MD \cap MI = \varnothing$

Siendo MD = conjunto de métricas directas y MI conjunto de métricas indirectas

Los axiomas no estructurales establecen relaciones entre atributos de un concepto, y son específicos de cada dominio. Un ejemplo de axioma no estructural puede ser "si el atributo tipoEscala es 'nominal' u 'ordinal', la unidad de esa escala debe ser null.

4.3.5 Instancias

Se usan para representar elementos específicos del dominio de la ontología. En nuestro ejemplo de la figura 4.1, podrían definirse escalas o métricas específicas detallando los valores de sus atributos.

4.4 Clasificación de las Ontologías.

Algunos autores [Gom99, Hei97, Miz95, Sow00], han clasificado las ontologías existentes según diferentes criterios. A continuación se exponen algunos de ellos y un resumen final que los relaciona por el grado de usabilidad y reusabilidad que supone cada clase.

4.4.1 Clasificación por Grado de Axiomatización

Según Sowa [Sow00], las ontologías se pueden clasificar en Terminológicas o Formales de acuerdo al grado de axiomatización que tenga la definición de sus categorías.

Ontologías Terminológicas: Una ontología terminológica define términos y sus relaciones en taxonomías que involucran tanto relaciones de subtipo y supertipo, como las que relacionan partes con un todo (part-whole), pero no incluyen axiomas y definiciones expresadas en lógica o algún tipo de lenguaje formal procesable por una computadora. Esto hace que se disponga de menos información del universo modelado, pero permite a su vez, por la relativa simplicidad de su especificación, construir ontologías de gran tamaño.

La mayoría de los campos de la ciencia, ingeniería, negocios y jurídico, han desarrollado sistemas de terminología o nomenclatura para designar, clasificar y estandarizar sus conceptos en enormes ontologías terminológicas. Ejemplos de estas ontologías son:

EDR. (The Electronic Dictionary Research project) [Edr02], es un proyecto japonés que ha desarrollado un diccionario con mas de 400.000 conceptos, con su traducción a palabras en inglés y japonés con muy poco detalle acerca de cada uno de sus términos.

WordNet [WordN], es una jerarquía de 166.000 palabras, muy usada para procesamiento de lenguaje natural por su temprana disponibilidad en Internet en http://www.cogsci.princeton.edu/~wn/.

***Ontologías Formales*:** Una ontología formal tiene sus categorías restringidas por axiomas y definiciones expresadas en lógica formal o en algún tipo de lenguaje procesable por la computadora. Las ontologías formales suelen tener menos cantidad de conceptos que las terminológicas, pero sus axiomas y definiciones pueden soportar computaciones e inferencias más complejas.

La diferencia entre una ontología terminológica y una formal, según Sowa [Sow00], es en el grado de especificación. Teóricamente, en la medida que se adicionen axiomas a una ontología terminológica, podrá evolucionar a una formal, pero la definición de axiomas no es una tarea trivial, y es por eso que en la práctica es difícil tal evolución.

4.4.2 Clasificación por Dependencia del Contexto

Según Mizoguchi, Vanwelkenhuysen e Ikeda [Miz95] las ontologías pueden clasificarse conforme al grado de dependencia del contexto que presenten, en el sentido de que aquellas menos dependientes del contexto serán las candidatas a ser más reusadas. La clasificación que surge es:

***Ontologías de dominio*:** Expresan conceptualizaciones que son específicas a un dominio particular, colocando restricciones en la estructura y contenido de un dominio de conocimiento mediante axiomas que se cumplen siempre entre los elementos de dicho dominio. Su principal objetivo es permitir el reuso de la ontología para diferentes aplicaciones que involucren al mismo dominio. Ejemplos de estas ontologías son:

The EngMath ontology [Gru94a], que es una ontología para el ámbito de la ingeniería en matemáticas, disponible en la biblioteca de ontologías de Ontolingua.

The Enterprise Ontology [Usc98] que se ubica en el dominio del modelado de procesos empresariales. Es por lo tanto, una colección de términos y definiciones relevantes al entorno empresarial.

Ontologías generales o de sentido común: Definen un vocabulario relacionado a cosas, eventos, tiempo, espacio, causalidad, comportamiento, función, etc.

La ontología CYC [Cyc02] disponible en http://www.Cyc.com es una ontología de este tipo, ya que provee una gran cantidad de conocimiento humano general. Además está dividida en micro teorías que permiten su mejor administración.

Meta ontologías, Ontologías genéricas (Core Ontologies): Son similares a las de dominio, pero los conceptos que definen se consideran genéricos a través de diferentes áreas de conocimiento y por ello reusables en diferentes dominios. Típicamente, las ontologías genéricas definen conceptos como estado, evento, proceso, acción, etc. Los conceptos de las ontologías de dominio son a menudo definidos como especializaciones de conceptos existentes en ontologías genéricas. Cabe destacar que el límite para considerar una ontología genérica no está bien definido, pero la distinción es intuitivamente significativa y útil cuando es necesario organizar ontologías en bibliotecas.

Un ejemplo de este tipo de ontologías es The Mereology Ontology (ontología de mereología) propuesta por Borst [Bor97] y que define los conceptos relacionados a la relación part-of y sus propiedades.

4.4.3 Clasificación por el Sujeto de Conceptualización

Algunos autores como Gomez-Perez, entre otros [Gom99, Hei97], también clasifican las ontologías por el tipo de dominio que modelan, o por el sujeto de conceptualización, en esta sección mostramos algunos tipos de ontologías comúnmente definidos en la literatura, las principales categorías de ontologías identificadas son:

Ontologías de tareas: Proveen un sistemático vocabulario de los términos usados para resolver problemas asociados con tareas particulares, ya sean dependientes o no del dominio. Por ejemplo, relacionados a la tarea de evaluación, tendremos términos que involucren a los conceptos de "medición", "cálculo" y "objetivo", así como el término que refiera a la acción "generación de informe".

Ontologías de aplicación: Contienen todas las definiciones que son necesarias para modelar el conocimiento requerido para una aplicación particular en un dominio dado. Típicamente son una mezcla de conceptos provenientes de ontologías de dominio y de ontologías genéricas. Las ontologías de aplicación no se construyen con el propósito de lograr reusabilidad en diferentes dominios.

Ontologías de dominio: Coincide con la clasificación propuesta por Mizoguchi, Vanwelkenhuysen e Ikeda [Miz95] que fue explicada anteriormente en la clasificación por dependencia del contexto (sección 4.4.2) .

Ontologías genéricas: Coincide con la clasificación propuesta por Mizoguchi, Vanwelkenhuysen e Ikeda [Miz95] que fue explicada anteriormente en la clasificación por dependencia del contexto (sección 4.4.2) .

Ontologías de representación de conocimiento: Describen las primitivas de representación usadas para formalizar conocimiento en paradigmas de desarrollo de sistemas basados en el conocimiento, es decir, explican la conceptualización que subyace en un formalismo de representación de conocimiento. Se pretende que sean neutrales con respecto a las entidades del mundo, es decir, que provean un marco representacional sin hacer aseveraciones acerca del mundo. Las ontologías genéricas y de dominio son descriptas usando ontologías de representación.

El ejemplo más representativo de esta clase de ontologías es la Frame-Ontology [Gru94b] disponible en el servidor Ontolingua: http://www-ksl-svc.stanford.edu. Esta ontología expone las primitivas de representación usadas en lenguajes basados en marcos (frames), definiendo términos como clases, subclases, atributos, valores, relaciones , axiomas, entre otros, y permitiendo que otras ontologías puedan ser especificadas usando convenciones basadas en frames.

Ontologías de más Alto Nivel (Top-Level) : Pretenden establecer una estructura básica, bajo la cual todos los términos de cualquier ontología existente, deberían poder relacionarse. Hasta ahora el principal problema es que no existe una ontología única de este tipo. Ejemplos de estas ontologías son:

Ontología Top-Level de Sowa [Sow00]: Esta ontología incluye las categorías básicas derivadas de una variedad de fuentes en lógica, lingüística, filosofía e inteligencia artificial.

La ontología es presentada como una estructura reticulada donde el concepto de más alto nivel es el *tipo universal*. Como subtipos del tipo universal se definen conceptos primitivos: *independiente, relativo*, etc; combinando estos conceptos primitivos se obtienen nuevos conceptos, hasta obtener el reticulado completo. Para cerrar el reticulado, el tipo *absurdo (absurd)* se considera un subtipo de todos los conceptos que no tienen descendientes.

Ontología CYC [Cyc02]: (También clasificada como ontología general o de sentido común en la sección 4.2.2), contiene alrededor de 3000 términos captados de los conceptos más generales de la realidad humana tradicionalmente consensuada.

Wordnet [WordN]: Esta ontología es una gran base de datos léxica para el idioma inglés (por eso se categoriza también como ontología terminológica). Está organizada en 70,000 conjuntos de sinónimos (“synsets”), cada uno de éstos conjuntos representa un concepto léxico subyacente. Los *synsets* están enlazados entre sí vía relaciones. Wordnet divide el léxico en cinco categorías: nombres, verbos, adjetivos, adverbios y funciones.

Modelo Superior Generalizado [Bat95]: La ontología *Generalized Upper Model* (GUM), es una ontología general independiente del dominio. GUM está dividida en dos jerarquías, la primera contiene todos los conceptos y la segunda contiene todos los roles.

En la figura 4.2 se puede ver cómo difieren algunas de las propuestas de ontologías de más alto nivel entre sí, en cuanto a la clasificación de los términos más generales.

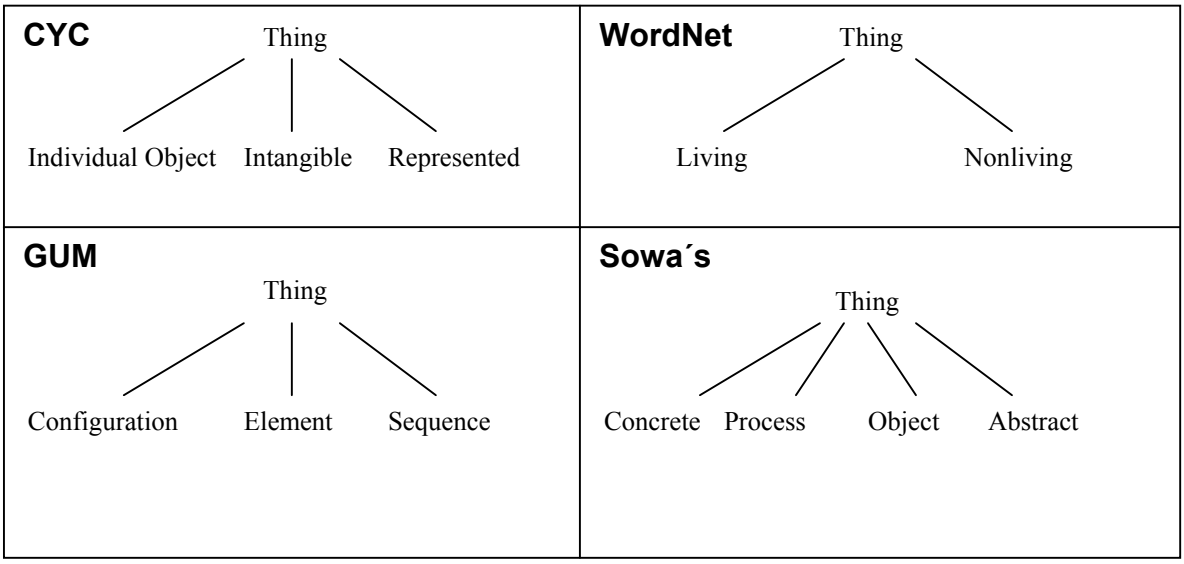

Figura 4.2 Diferencias entre ontologías top-level [Cha99].

4.4.4 Conclusiones sobre la Clasificación de Ontologías

A modo de resumen considerando los distintos tipos de ontologías, se muestra un esquema extraído del trabajo de Benjamins y Gómez Pérez [Ben96], en la figura 4.3, que relaciona los tipos de ontologías más comunmente usadas por su grado de usabilidad y reusabilidad. Se puede concluir que en la medida en que una ontología es más usable en un contexto dado, es menos reusable en otro contexto [Ben96].

En la figura 4.3 se ubican de abajo hacia arriba las ontologías por su grado de generalidad y reusabilidad, las más generales abajo. El mismo esquema las clasifica también según sean ontologías orientadas a un domino específico (a la izquierda) o a la tarea (a la derecha).

Las Meta Ontologías, Ontologías de dominio y Ontologías de aplicación capturan el conocimiento estático del dominio en una forma independiente del método de resolución de problemas que se pueda utilizar. Por otro lado las Ontologías de tareas y

Ontologías de dominio-tarea están destinadas a conceptualizar y exponer conocimiento de resolución de problemas.

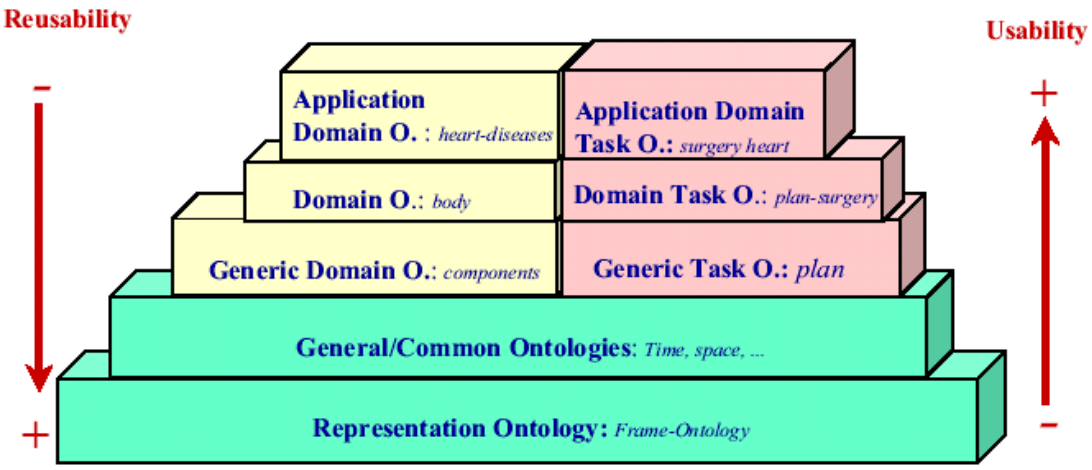

Figura 4.3 El equilibrio entre reusabilidad y usabilidad en ontologías (extraído de [Ben96]).

Teniendo en cuenta las distintas clasificaciones de ontologías existentes en la literatura (para las que no existe un único criterio consensuado), la ontología de métricas e indicadores que se propone en esta tesis (desarrollada en el capítulo 5), se clasifica como **ontología de dominio.** Es una ontología de dominio porque se define y es aplicable al dominio de métricas e indicadores para el propósito específico de medición y evaluación en el ámbito de Ingeniería de Software; y se ubica en un nivel intermedio de generalidad, (y por lo tanto, de grado de usabilidad y reusabilidad según [Ben96]).

4.5 Recomendaciones de Diseño.

Algunos autores [Ben96, Gru93], establecen características deseables que deberían cumplir las ontologías, y resaltan la importancia de tenerlas en cuenta al momento de diseñar una ontología. A continuación se presentan algunas de estas recomendaciones.

Claridad. Según Gruber [Gru95] una ontología debería efectivamente comunicar el significado propuesto para los términos definidos en ella. Las definiciones deberían entonces ser tan objetivas como sea posible.

Completitud. Según Gruber [Gru95] siempre que sea posible, debería proporcionarse una definición completa, es decir, definida por condiciones necesarias y suficientes, y no solamente por condiciones necesarias. Además, todas las definiciones deberían estar documentadas en lenguaje natural.

Coherencia. Según Gruber [Gru95] una ontología debería ser coherente, es decir, debería sólo concluir inferencias que sean consistentes con las definiciones que contiene la propia ontología. En definitiva, esto significa que sus axiomas deberán ser lógicamente consistentes.

La coherencia debería alcanzar también a los conceptos que son definidos informalmente, como por ejemplo lo que es descripto en lenguaje natural y ejemplos que se proporcionan para clarificar ideas. Si una sentencia que puede ser inferida de los axiomas de la ontología contradice la definición de un ejemplo presentado informalmente, entonces se deberá considerar que la ontología es inconsistente.

Extensibilidad. Según Gruber [Gru95] una ontología debería soportar la definición de nuevos términos basándose en el vocabulario existente, de manera tal que no requiera la revisión de las definiciones existentes.

Esto se lograría manteniendo cierto equilibro entre ser lo suficientemente específicos en la definición como para permitir el uso en el ámbito para el cuál la ontología se define, pero no demasiado específicos, de manera que la ontología pueda ser de utilidad para otros usos futuros.

Mínimo compromiso ontológico. Según Gruber [Gru95] una ontología debería hacer la menor cantidad de aseveraciones posibles acerca del mundo que está siendo modelado, permitiendo de esta forma que las partes que están comprometidas con la ontología (los diferentes agentes que la usarán) tengan libertad de especializar e instanciar la ontología cuando sea necesario.

Diversificación de jerarquías. Para que la ontología se vea favorecida con el poder que otorgan los mecanismos de herencia múltiple [Ben96], es conveniente usar tantos criterios de clasificación como sea posible, de manera de representar más conocimiento en la ontología. De esta manera es muy sencillo agregar un término porque se puede definir fácilmente especificado desde los conceptos preexistentes y criterios de clasificación.

Minimización de la distancia semántica entre conceptos hermanos. Conceptos similares se deben agrupar y representar como subclases de una clase y deberían ser definidos usando las mismas primitivas. Por otro lado, conceptos que son menos similares se deberían representar aparte en la jerarquía [Ben96].

Estandarización de nombres. La estandarización de nombres definiendo y respetando reglas para su formación siempre que sea posible, es una característica deseable para ayudar en el mantenimiento de una ontología [Ben96].

Una posible estandarización es especificar el nombre de una relación como la concatenación del nombre de la ontología (o el del concepto que es primer elemento de la relación) con el nombre simple de la relación y con el nombre del concepto destino. Por ejemplo, para la ontología de la figura 4.1, el nombre de la relación *Cuantifica,* podría ser: *Métrica_Cuantifica_Atributo,* lo que ayudaría al entendimiento del significado de la relación.

4.6 Metodologías de Diseño y Construcción.

Ante la falta de un consenso entre las comunidades involucradas, que permita la formulación de una metodología estándar, cada grupo de desarrollo ha usado su propio conjuntos de principios, criterios de diseño y etapas en el proceso de diseño y construcción de una ontología.

Afortunadamente varios autores [Cor01, Fer99a] han realizado trabajos de recopilación mostrando la perspectiva general y el estado del arte en desarrollo de metodologías de diseño y construcción para ontologías.

A continuación se presentan las metodologías más relevantes, conjuntamente con ejemplos de la aplicación de las mismas en diferentes experiencias.

4.6.1 Metodología de Grüninger y Fox

La metodología propuesta por Grüninger y Fox [Grü95] recomienda un método formalizado para construir ontologías. La figura 4.4 ilustra los pasos de la metodología de Grüninger y Fox., que detallamos a continuación:

4.6.1.1 Pasos Recomendados para la Construcción de la Ontología

Paso 1: Definición de los escenarios motivadores. El desarrollo de ontologías es motivado por escenarios, que surgen en las aplicaciones. En particular, tales escenarios pueden ser presentados como la descripción de problemas que no han podido ser resueltos hasta el momento, posibles aplicaciones en las cuales la ontología podrá ser usada, entre otros.

Un escenario motivador suele proveer un conjunto de soluciones intuitivamente posibles a los problemas enumerados. Esas soluciones dan una semántica informal para los objetos y relaciones que serán luego incluidos en la ontología.

Cualquier propuesta de creación de una ontología o extensión de una existente, debería describir uno o más escenarios motivadores, y el conjunto de soluciones proyectadas a los problemas de dichos escenarios.

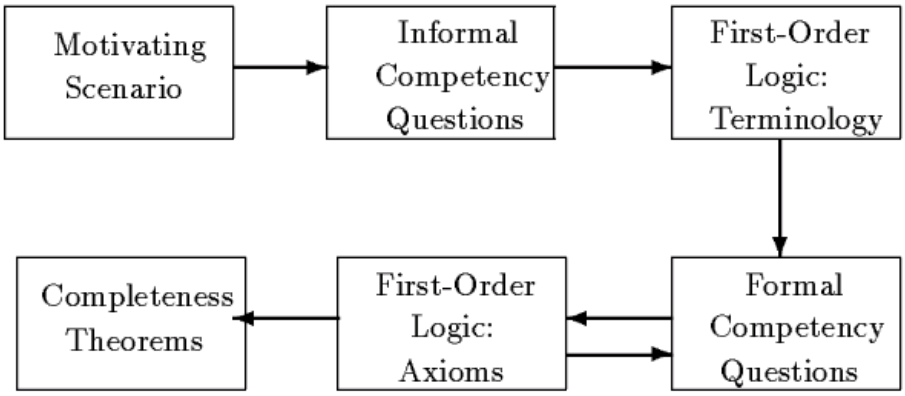

Figura 4.4 Etapas en la metodología de Grüninger y Fox (extraído de Grü95]).

Paso 2: Especificación informal de las preguntas pertinentes (competency questions). Dado el escenario motivador, un conjunto de preguntas surgirán como demandas que la ontología a construir debería resolver. Se pueden considerar esas preguntas como requerimientos en forma de interrogantes (competency questions). Una ontología debe ser capaz de representar esos interrogantes usando su terminología, y ser capaz de caracterizar las respuestas a ellas usando axiomas y definiciones. Las preguntas pertinentes son usadas para determinar el alcance de la ontología.

Se consideran informales, pues no están aún expresadas en el lenguaje formal de la ontología. Las preguntas pertinentes o interrogantes, deberían ser definidas de una manera estratificada, de forma que la respuesta a un interrogante pueda ser usada para responder interrogantes más generales de la misma u otra ontología por medio de operaciones de composición y descomposición. Esta estratificación, en realidad es una forma de identificar conocimiento para reuso.

Las preguntas no generan compromisos ontológicos de diseño, sino que son usadas luego, para evaluar los alcances de la ontología en cuanto a la capacidad para satisfacer los requerimientos que plantean.

Paso 3: Especificación de la terminología. A partir de las preguntas formuladas en el paso anterior, se puede extraer el conjunto de términos usados para expresarlas y considerarlo como base para la terminología a usar en un lenguaje formal.

Si se está diseñando una nueva ontología, para cada interrogante informal planteado, debe haber objetos, atributos, y relaciones en la ontología en construcción que son intuitivamente requeridos para resolver la pregunta.

El primer paso en la especificación de la terminología de la ontología es identificar los objetos en el dominio de aplicación, que serán representados por constantes y variables del lenguaje.

Luego se definen los atributos de objetos y relaciones, usando predicados en lógica de primer orden, o en un formalismo equivalente.

Paso 4: Formalización de las preguntas pertinentes (competency questions). Una vez definida la terminología de la ontología, se definen formalmente las preguntas pertinentes en función de dicha terminología.

Paso 5 Especificación de axiomas formales. Los axiomas en una ontología especifican las definiciones de términos y restricciones en su interpretación. Se definen como sentencias en lógica de primer orden.

Para los autores de la metodología [Grü95] un conjunto simple de objetos, o un conjunto de cláusulas básicas en lógica de primer orden, no constituye una ontología. Deben proveerse axiomas para definir la semántica del conjunto.

Si el conjunto de axiomas es insuficiente para representar las preguntas pertinentes y caracterizar sus soluciones, es un indicativo de que se deben agregar nuevos objetos o axiomas a la ontología hasta lograrlo. Se puede ver entonces, que la construcción de axiomas y la evaluación de las preguntas pertinentes es un proceso iterativo.

Paso 6: Verificar la completitud de la ontología. Una vez que las preguntas pertinentes han sido formalmente enunciadas se deben definir las condiciones bajo las cuales las soluciones son consideradas completas.

4.6.1.2 Ontologías Desarrolladas usando la Metodología

Esta metodología fue usada para construir las ontologías del proyecto TOVE (TOronto Virtual Enterprise) en el Enterprise Integration Laboratoy de la Universidad de Toronto [TOVE] . Dichas ontologías constituyen un modelo integrado y formalizado

usando lógica de primer orden que incluye Enterprise Design Ontology, Project Ontology, Scheduling Ontology y Service Ontology.

4.6.2 Metodología de Unschold y King

Esta metodología se basa en la experiencia de los autores en el desarrollo de la ontología: "*Enterprise Ontology*" [Usc98], y provee guías para el desarrollo de ontologías.

4.6.2.1 Pasos Recomendados para la Construcción de la Ontología

Paso 1: Identificar el propósito. El objetivo de esta etapa es dejar claro porqué se construye la ontología y qué uso se le pretende dar. También es útil en este momento identificar y caracterizar el rango de posibles usuarios y aplicaciones de la ontología.

Paso 2: Construir la ontología. En esta etapa se distinguen tres sub-tareas:

Paso 2.1: Capturar la ontología. Por la tarea de capturar la ontología los autores entienden lo siguiente:

- Identificar los conceptos claves y relaciones en el dominio de interés.
- Producir definiciones textuales precisas y carentes de ambigüedad que describan tales conceptos y relaciones.
- Identificar los términos para referirse a tales conceptos y relaciones.

Los autores usan un enfoque middle-out[1] para ejecutar esta etapa, y recomiendan que antes de buscar los conceptos más generales o particulares como conceptos clave, se identifiquen los conceptos más importantes, que luego serán usados para obtener el resto en la jerarquía por medio de generalización y especialización.

Paso 2.2: Codificación. Involucra explícitamente representar el conocimiento adquirido en el paso anterior, en un lenguaje formal.

Paso 2.3: Integrar ontologías existentes. Durante los procesos de captura y codificación debe plantearse la pregunta de si corresponde usar ontologías que ya existen, y cómo hacerlo.

[1] La captura de los conceptos de la ontología se puede llevar a cabo siguiendo tres enfoques distintos [Usc96b]: desde los conceptos más concretos a los más abstractos (bottom-up), desde los más abstractos a los más concretos (top-down) o un enfoque intermedio que combina a ambos, también llamado desde dentro hacia fuera (middle-out).

Paso 3: Evaluación. Los autores adoptan la definición de Asunción Gómez Pérez, N. Juristo y J.Pazos: "hacer una evaluación de las ontologías, sus ambientes de software asociados, y documentación con respecto al marco de referencia ...El marco de referencia puede ser especificaciones de requerimientos, preguntas pertinentes, y/o el mundo real".

Paso 4: Documentación. Se recomienda el establecimiento de pautas para documentar la ontología construida, posiblemente diferentes de acuerdo al tipo y propósito de la ontología.

4.6.2.2 Ontologías Desarrolladas usando la Metodología

El proyecto más importante que se desarrolló usando esta metodología es The Enterprise Ontology [Usc98], que es una colección de términos y definiciones relevantes para el dominio de empresas de negocios. La ontología fue desarrollada en el marco del Enterprise Project del Artificial Intelligence Applications Institute de la Universidad de Edimburgo.

4.6.3 METHONTOLOGY

Esta metodología fue desarrollada dentro del laboratorio de Inteligencia Artificial de la Universidad Politécnica de Madrid [Fer97]. El marco de trabajo (framework) METHONTOLOGY permite la construcción de ontologías a un nivel de conocimiento. Dicha metodología provee un conjunto de guías de cómo se deberían llevar a cabo las actividades identificadas en el proceso de desarrollo de una ontología.

4.6.3.1 Tareas Recomendadas para la Construcción de la Ontología

Tarea 1: Establecimiento del proceso de desarrollo de la ontología. En esta etapa se identifican qué actividades se realizaran al construir la ontología. Se deben definir las **Actividades de gerenciamiento del proyecto** (Incluyen planificación, control y aseguramiento de la calidad), las **Actividades orientadas al desarrollo del proyecto** (incluyen especificación, conceptualización, formalización, implementación y mantenimiento) y las **Actividades de apoyo** que incluyen una serie de actividades ejecutadas al mismo tiempo que las actividades orientadas al desarrollo, sin las cuales la ontología podría no llegar a ser construida, (incluyen adquisición de conocimiento, evaluación, integración, documentación y administración de versiones).

Tarea 2: Desarrollo de la ontología. El ciclo de vida para el desarrollo de una ontología propuesto por METHONTOLOGY incluye las siguientes actividades orientadas al desarrollo [Fer99a] (ver figura 4.5):

Especificación: El propósito de la especificación es definir en un documento cuál es el principal objetivo de la ontología, con qué propósito se desarrolla (posibles aplicaciones), cuál es su nivel de generalidad y granularidad, y cuál es su alcance (establecimiento de los límites del dominio a cubrir).

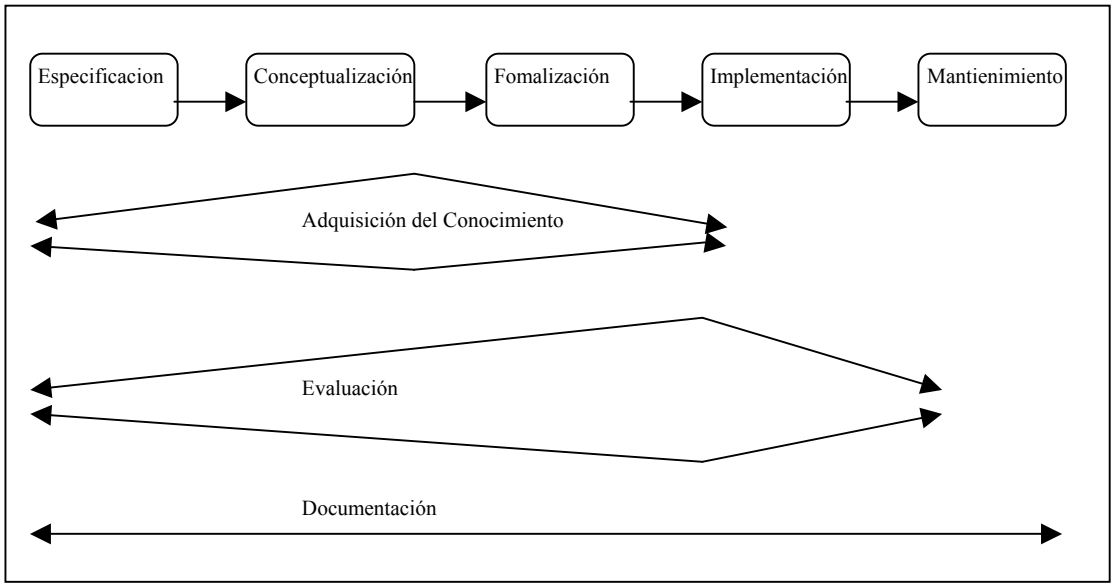

Figura 4.5 Ciclo de vida en METHONTOLOGY (Adaptada de [Fer99b])

Conceptualización. Estructura el conocimiento del dominio usando modelos conceptuales en un nivel de conocimiento. Una vez que se ha adquirido la mayoría del conocimiento del dominio en cuestión, los desarrolladores tienen un conjunto de conocimiento no estructurado que debe ser organizado. La actividad de conceptualización tiene como objetivo organizar y estructurar el conocimiento adquirido, usando un conjunto de *Representaciones Intermedias* (IR) que son modelos para representar conceptos independientes de los lenguajes y ambientes de desarrollo que se usarán en la implementación de la ontología. Gómez-Pérez y otros [Gom96], en su trabajo "Towards a Method to Conceptualize Domain Ontologies", proponen un conjunto de representaciones intermedias (como Diccionario de Datos, Árboles de Clasificación, Tablas de Atributos, entre otros) para la conceptualización de ontologías de dominio.

Formalización transforma el modelo conceptual en un modelo formal o semi-formal computable.

Implementación construye modelos computables en un lenguaje computacional, como podría ser Ontolingua, RDF/S (Resource Description Framework/Schema) u OWL (Ontology Web Language).

Mantenimiento actualiza y corrige la ontología.

Paralelamente al las actividades de desarrollo de la ontología se deben llevar a cabo actividades de apoyo al desarrollo, importantes para la construcción de la ontología, dichas actividades son:

Adquisición del conocimiento. Consiste en la adquisición del conocimiento del dominio. Esta actividad es independiente de las actividades de desarrollo de la ontología.

Evaluación consiste en hacer un valoración técnica de la ontología, su ambiente de software y documentación, con respecto al marco de referencia durante cada fase y entre fases del ciclo de vida.

Documentación. La documentación detallada, clara y exhaustiva es necesaria para cada una de las fases completadas y productos generados.

4.6.3.2 Ontologías Desarrolladas usando la Metodología

La ontología más citada construida con esta metodología es **CHEMICALS** [Fer99b], la cual contiene conocimiento sobre el dominio de elementos químicos y estructuras cristalinas.

Otra ontología desarrollada usando esta metodología es **The Reference-Ontology**, una ontología en el dominio de las ontologías (una meta-ontología) que juega el rol de una especie de páginas amarillas de ontologías. Reúne, describe y relaciona ontologías existentes, usando una organización lógica común.

4.6.4 Resumen

Sintetizando y comparando las metodologías analizadas en las secciones anteriores, presentamos un cuadro comparativo (ver tabla 4.1) basado en el trabajo realizado por Fernández [Fer99a], en donde se hace un estudio de las metodologías más difundidas de desarrollo de ontologías, estableciendo un conjunto de criterios para comparar dichas metodologías.

Algunos de los criterios seleccionados como base de comparación (se tuvieron en cuenta aquellos criterios del estudio de [Fer99a] considerados de interés para el desarrollo de nuestra ontología de Métricas e Indicadores) son los siguientes:

- **Detalle de Actividades y Técnicas:** Consideración del grado de detalle en el que están especificadas las actividades y técnicas propuestas por la metodología.

- **Recomendaciones para la formalización del conocimiento**: formalismos propuestos para representar los conocimientos del dominio.

- **Estrategia para construir ontologías**: se refiere a la relación existente entre la ontología y la aplicación que la va a explotar. Fernández las clasifica en: dependientes, semi-dependientes e independientes de la aplicación.

- **Estrategia para identificar conceptos**: Esta puede ser [Usc96b]: desde los conceptos más concretos a los más abstractos (bottom-up); desde los más abstractos a los más concretos (top-down) o un enfoque intermedio que combina a ambos, también llamado desde dentro hacia fuera (middle-out).

Criterio	**Grüninger y Fox**	**Unschold y King**	**METHONTOLOGY**
Detalle de Actividades y técnicas	Ningún	Muy poco	Muy detalladas.
Recomendaciones para la formalización del conocimiento	Lógica de primer orden	Ninguna	Ninguna
Estrategia para construir ontologías	Semidependiente de la aplicación	Independiente de la aplicación	Independiente de la aplicación
Estrategia para identificar conceptos	Middle-out	Middle-out	Middle-out
Ciclo de vida	No definido totalmente	Ningun	Prototipos evolutivos
Diferencias con el estándar IEEE 1074-1995	Faltan muchos procesos y actividades	Faltan muchos procesos y actividades	Faltan algunos procesos y actividades
Técnicas recomendadas	Desconocido	Desconocido	Recomienda algunas técnicas
Ontologías desarrolladas con la metodología	TOVE	Enterprise ontology	CHEMICALS. Ontología para medio ambiente. Reference Ontology

Tabla 4.1 Comparación de metodologías de desarrollo de ontologías

- **Ciclo de vida**: cuál es el ciclo de vida recomendado por la metodología.

- **Diferencias con el estándar IEEE 1074-1995**: El estándar IEEE 1074-1995 describe el proceso de desarrollo de software, las actividades que deben realizarse y las técnicas que se pueden usar en el desarrollo del software. En este ítem se analizan las diferencias con dicho estándar[2] (en los párrafos siguientes se muestran algunas diferencias entre las metodologías de desarrollo de Ingeniería de Software, y las metodologías para el desarrollo y construcción de ontologías).

- **Técnicas recomendadas**: si se proponen técnicas particulares para cada actividad recomendada del ciclo de vida.

- **Ontologías desarrolladas con la metodología**: Ejemplos de ontologías desarrolladas usando la metodología.

En conclusión, podemos observar que no existe ninguna metodología totalmente madura si se compara con el nivel de detalle especificado para los procesos en el estándar IEEE 1074 o en ISO 12207 [ISO12207]. Tampoco existe un consenso entre los desarrolladores de ontologías sobre qué metodología es conveniente usar para distintas necesidades. Se podría indicar que cada grupo aplica su propia metodología.

Según Fernández [Fer99a], METHONTOLOGY es la metodología más madura, además es la metodología recomendada por la "Fundación para los Agentes Físicos Inteligentes (FIPA), que promueve la interoperabilidad entre aplicaciones basadas en agentes (http://www.fipa.org). Sin embargo, algunas actividades y técnicas deberían ser especificadas con más detalle.

Cabe destacar que aunque las actividades propuestas por cada metodología, en el ciclo de vida de una ontología, tienen alguna influencia proveniente de las actividades especificadas en el proceso de Ingeniería de Software, existen diferencias, y las principales son:

- En Ingeniería de Software luego de las actividades de especificación de requerimientos y análisis, el diseño no es dividido en conceptualización y formalización como la mayoría de las metodologías proponen.

- Las actividades de adquisición de conocimiento no existen con tanto énfasis en el proceso de Ingeniería de Software, y son en cambio una parte esencial de cualquier proceso para la especificación y construcción de ontologías.

[2] De acuerdo a la definición de la IEEE [IEEE90], software es "Programas de computadoras, procesos, además de documentación y datos posiblemente asociados, que pertenecen al funcionamiento de un sistema de computadora"; las ontologías son parte de un producto de software, por lo tanto deberían ser desarrollados de acuerdo a los estándares propuestos para desarrollo de software general, el cual deberá ser adaptado a las características especiales del desarrollo de ontologías [Fer99a].

Para la construcción de la ontología de Métricas e Indicadores, que será detallada en el próximo capítulo, se usó la metodología METHONTOLOGY [Fer97] por ser una de las metodologías mas difundidas y maduras y por contar con actividades y técnicas detalladas para el proceso de desarrollo de ontologías.

4.7 Áreas de Aplicación de las Ontologías

A continuación se citan algunas de las áreas en las que actualmente se usan las ontologías.

- **Ingeniería del Conocimiento.** La ingeniería del conocimiento ha emergido como un disciplina diferente, pero estrechamente relacionada al de la Ingeniería de Software. Entre sus aspectos distintivos podemos enumerar a un conjunto de técnicas de elicitación (extracción) y modelado del conocimiento, un conjunto de formalismos para representar el conocimiento, y a un conjunto de mecanismos para automatizar razonamiento.

El aporte de las ontologías en el proceso de ingeniería del conocimiento está en las siguientes etapas:

Modelado conceptual, donde se crea un glosario de la terminología del dominio de la aplicación (los conceptos), se definen relaciones entre dichos términos y restricciones en su uso. Este modelo conceptual explícito es la ontología.

Construcción de la base de conocimiento. Usando la ontología definida en la etapa anterior como un conjunto de esquemas o contenedores de conocimiento, se genera la base de conocimiento con instancias del dominio bajo la forma de reglas, hechos, eventos y restricciones.

- **Procesamiento de lenguaje natural.** En el área de procesamiento de lenguaje natural, una ontología puede mantener la definición de elementos gramaticales del lenguaje y la relación entre ellos, permitiendo por ejemplo el análisis sintáctico de un texto.

- **Interoperabilidad entre sistemas de información heterogéneos.** En la interoperabilidad de sistemas de información, las ontologías se presentan como una importante solución para lograr una integración inteligente, en particular en el área de bases de datos, una ontología puede ser un elemento clave asociado a un mediador (puente), que integra datos provenientes de fuentes heterogéneas.

Con una ontología terminológica, se pueden organizar los términos que son usados en interacciones entre sistemas heterogéneos, de manera de reconocer cuando una aplicación está usando un término que es más general o más específico que otro que está en uso por otra aplicación.

Si la ontología es formal, se puede contar con una definición más completa de cómo se relaciona un término de un origen con el de otro, y eventualmente usar axiomas definidos que los vinculen por igualdad o que expresen un término exactamente en función del otro, lo que permitiría establecer correspondencias seguras y automáticas entre ellos.

- **Búsqueda semántica en sitios Web.** Usando ontologías asociadas a los motores de búsqueda se puede reemplazar la búsqueda tradicional basada en palabras claves, por una búsqueda semántica, encontrando páginas cuyo contenido difiere sintácticamente, pero semánticamente son similares. De esta forma las consultas no serán procesadas a un nivel terminológico, sino conceptual, con los beneficios que eso supone en el grado de acierto de las respuestas retornadas.

- **Modelado de empresas.** En el área de modelado de empresas, las ontologías desempeñan entre otros, el rol de mantener una memoria organizacional colectiva que permita a los distintos niveles de la empresa interoperar con un lenguaje común y con reglas únicas.

Las Ontologías TOVE (Toronto Virtual Enterprise) desarrolladas en la universidad de Toronto y la Enterprise Ontology desarrollada en la Universidad de Edimburgo, son ejemplos de este tipo de ontologías.

- **Aplicaciones en la Web Semántica.** Las ontologías se volverán un elemento clave en la Web Semántica, ya que permitirán explicitar la semántica de los contenidos dispersos en la web. Es decir, las ontologías proveen un vocabulario de marcas consensuado y con un significado bien definido, permitiendo su procesamiento automático y la inferencia de nuevos conocimientos.

4.8 Conclusiones

Diversos autores han señalado la importancia del desarrollo de ontologías como base para abordar la definición y diseño de sistemas de información basados en el conocimiento [Gru93, Gua95]. También la comunidad de investigadores de la Web Semántica [W3Csw], considera a las ontologías una pieza fundamental para publicar contenidos semánticos en la web, procesables por computadoras.

En este capítulo se presentaron los conceptos básicos relacionados con ontologías y algunas de sus principales aplicaciones.

Al igual que pasa con los sistemas software, es recomendable construir las ontologías utilizando un enfoque ingenieril, es decir, siguiendo alguna metodología adecuada para tal fin. Por tal razón se analizaron algunas de las metodologías más difundidas y se eligió la metodología METHONTOLOGY debido a que es una de las metodologías más maduras que imita el ciclo de vida del software propuesto en el estándar IEEE 1074, entre otras fortalezas, tal como se resumió en la tabla 4.1.

En el próximo capítulo se mostrará detalladamente el desarrollo de la ontología de métricas e indicadores construida usando dicha metodología.

Capítulo 5- Ontología para el Ámbito de Métricas e Indicadores de Software

5.1 Introducción

A medida que la Ingeniería de Software va progresando hacia una disciplina más madura, se van obteniendo importantes resultados de investigación relacionados al área de medición y evaluación de productos y procesos de software. De hecho ya se ha publicado una gran cantidad de información relacionada a atributos, métricas, escalas y métodos de medición para distintos propósitos y dominios del área de software.

Sin embargo, se observa que el desarrollo rápido y caótico de tanta información relacionada a métricas e indicadores, proveniente de fuentes heterogéneas, y con frecuencia, la falta de consenso en la terminología utilizada, dificulta su aplicación y reuso de forma eficiente en los procesos de evaluación, estimación y, en general de aseguramiento de calidad.

Por lo tanto se considera importante llegar a un acuerdo entre investigadores, profesionales de Ingeniería de Software y otros participantes relacionados al ámbito de medición y evaluación, sobre la terminología y significado de conceptos primitivos como son: atributo, métrica, medida, medición, método de cálculo, escala, entre otros. Con este propósito, en este capítulo se realiza una conceptualización del dominio de métricas e indicadores que se formaliza en una ontología para dicho ámbito.

La ontología propuesta, es usada como base para el diseño y construcción de un sistema de catalogación basado en la Web Semántica, que permitirá explorar, reusar y compartir información relacionada a métricas e indicadores. En el próximo capítulo se mostrará la arquitectura, el diseño navegacional y la construcción de un prototipo de dicho sistema de catalogación.

5.2 Fuentes de Conocimiento

Las fuentes de conocimiento que se tuvieron en cuenta para el desarrollo de la ontología de métricas e indicadores provienen de varios recursos a saber:

– La experiencia adquirida por los integrantes del grupo GIDIS (Grupo de Investigación y Desarrollo en Ingeniería de Software) *http://gidis.ing.unlpam.edu.ar/* en la elaboración de trabajos previos de investigación en el área de métricas, métodos y

procesos de software[Ols98]. Dicha experiencia fue volcada en discusiones de grupo tendientes a definir conceptos y terminología del ámbito de métricas e indicadores.

– Reuniones entre diferentes grupos ibero-americanos de investigadores en el ámbito de Ingeniería de Software; que tuvieron lugar con el objetivo de discutir y tratar de llegar a un consenso en la conceptualización de una ontología de la medición de software, como resultado de dichas reuniones se elaboró un informe técnico [Gar04] .

– Algunas publicaciones y trabajos previos de investigación reconocidos internacionalmente, en el área de medición y evaluación de software, como son:

- El libro escrito por Zuse [Zus98], el cual presenta un marco de validación para medidas de software.
- El trabajo de Kitchenham y otros [Kitch01], que presenta un modelo de datos conceptual para representar colecciones de conjuntos de datos relacionados a mediciones de software.
- La propuesta de Briand y otros [Bri02], que se basa en la definición de medidas dirigidas por metas.
- El libro de Fenton y Pfleeger [Fen97], que es una importante referencia ya que provee un extenso marco de trabajo para entender y usar los conceptos relacionados a mediciones y métricas de software.
- Propuestas de ontologías recientes, tal como la ontología de mediciones de software, documentada por Genero y otros [Gen03] inspirada principalmente en las definiciones del estándar ISO 15939 [ISO15939].
- El trabajo previo de Olsina [Ols00, Ols02a], autor del Método de Evaluación de Calidad de Sitios Web (WebQEM), el cuál está basado en el diseño, selección e implementación de métricas, indicadores y los criterios de decisión asociados, a partir de un concepto calculable.

– Principalmente, nuestra ontología de métricas e indicadores se ha inspirado en las definiciones de los estándares de organismos internacionales [Mar03a, Ols04], a saber:

- La terminología provista por el estándar ISO/IEC 15939 [ISO15939], el cual se refiere al proceso de medición de software.
- La terminología provista por el estándar ISO/IEC 9126-1 [ISO9126], el cual se refiere principalmente al modelo de calidad de productos de software.
- La terminología provista por el estándar ISO/IEC 14598-1 [ISO14598], el cual da una visión general de la evaluación de productos de software.

5.3 Ontología de Métricas e Indicadores

Para el desarrollo de la ontología se usó la metodología METHONTOLOGY (descripta en la sección 4.6.3), que provee guías para especificar ontologías en un nivel de conocimiento, particularmente provee técnicas para especificar la conceptualización. Los principales pasos propuestos por esta metodología son: Especificación, Conceptualización, Implementación y Evaluación.

Seguidamente, en la sección 5.3.1 se definen los principales requerimientos de la ontología de métricas e indicadores elaborados en la etapa de Especificación; luego, en la sección 5.3.2 se detalla la etapa de Conceptualización, para lo cual, se muestran las técnicas empleadas para tal fin (sección 5.3.2.1), se explican los conceptos involucrados en la ontología (sección 5.3.2.2), que luego se ilustran a través de un ejemplo (sección 5.3.2.3), se definen los términos de la ontología y sus relaciones (sección 5.3.2.4), y por último se muestran las representaciones semiformales y los axiomas obtenidos en esta etapa (sección 5.3.2.5); en la sección 5.3.3 se analizan aspectos técnicos de la etapa de implementación de la ontología y finalmente se exponen algunas conclusiones sobre el desarrollo de la ontología de métricas e indicadores (sección 5.4).

5.3.1 Especificación

En la etapa de especificación se deben considerar los principales requerimientos de la ontología, como son: el dominio modelado, el propósito, el alcance y las fuentes de conocimiento utilizadas, a continuación detallamos cada uno de estos puntos para nuestra ontología.

Dominio modelado: Elementos para la medición y/o estimación de artefactos de software (recursos, productos, procesos) para satisfacer distintas necesidades de información.

Propósitos de la ontología: Esta ontología tiene varios propósitos, los principales son:

- Facilitar el intercambio de información relacionada a métricas e indicadores de software entre los ingenieros, gerentes, usuarios y demás participantes en proyectos de medición y evaluación de artefactos software.

- Conformar la base de conocimiento para el diseño y construcción de un sistema de catalogación de métricas e indicadores [Ols03] que facilite la documentación, recuperación y búsqueda semántica de toda la información relacionada, sirviendo de apoyo a los procesos de aseguramiento de

calidad, definición y especificación de requerimientos no funcionales, definición de tests de calidad, entre otros.

- Esta ontología puede además ser útil como sub-ontología de una ontología de procesos de evaluación y medición más general.

Alcance: El alcance de la ontología comprende todos los conceptos, atributos y relaciones necesarios para la catalogación y explotación de la información relacionada a métricas e indicadores en el ámbito de Ingeniería de Software e Ingeniería Web. La ontología no incluye la especificación de instancias de los conceptos definidos.

Fuentes de conocimiento utilizadas: Las fuentes de conocimientos fueron detalladas en la sección 5.2, a continuación las resumimos en una lista:

- El libro de Zuse "A Framework of Software Measurement" [Zus98]
- El trabajo de Kitchenham y otros "Modeling Software Measurement Data" [Kitch01]
- La propuesta de Briand y otros "An Operational Process for Goal-driven Definition of Measures" [Bri02].
- El libro de Fenton y Pfleeger "Software Metrics: A Rigorous and Practical Approach"[Fen97].
- La propuesta de Genero y otros "An Ontology for Software Measurement" [Gen03].
- El trabajo de tesis de Olsina "Quantitative Methodology for Evaluation and Comparison of Web Site Quality" [Ols00].
- El estándar ISO/IEC 15939 ISO[15939].
- El estándar ISO/IEC 9126-1 [ISO9126].
- El estándar ISO/IEC 14598-1 [ISO14598].

5.3.2 Conceptualización

En esta sección se muestra la conceptualización del dominio de métricas e indicadores, presentando primero estrategias y técnicas usadas para tal fin, y una descripción detallada del dominio para facilitar su entendimiento.

5.3.2.1 Técnicas Empleadas en la Etapa de Conceptualización

Para la etapa de conceptualización se emplearon estrategias y técnicas como las tablas sugeridas para la metodología METHONTOLOGY [Gom96] y diagramas UML.

El uso de UML como lenguaje de representación de ontologías ha sido propuesto por diversos autores [Cra99, Cra01a, Kog03]. Algunos de ellos también proponen transformaciones de UML a RDF [Cra01b] y de UML a DAML [Bac02], para la implementación de ontologías modeladas con UML.

Si bien UML se utilizó originalmente para la comunicación entre seres humanos de modelos para la construcción de sistemas con un enfoque orientado a objetos, actualmente se está usando para modelar artefactos de Ingeniería de Software más declarativos tal como son las ontologías [Kog03], los esquemas XML [Car01], esquemas RDF y esquemas de bases de datos. Además existen otras buenas razones para usar el lenguaje UML como notación para la conceptualización de una ontología, a saber:

- Las ontologías incluyen taxonomías de conceptos (jerarquías clase/subclase), relaciones entre conceptos (asociaciones) y definición de atributos de conceptos. Esta información de las ontologías puede ser modelada con diagramas de clases del lenguaje UML.

- UML es una notación gráfica que surgió con el aporte de la experiencia de muchos años en análisis y diseño de software, por parte de una variedad de compañías en un amplio espectro de industrias y dominios.

- Es un estándar abierto administrado por el Object Management Group (OMG).

- Provee mecanismos para definir extensiones para aplicaciones específicas como es el modelado de ontologías.

- Está soportado por herramientas CASE ampliamente difundidas, y más accesibles a los profesionales del software que herramientas específicas para ontologías como Ontolingua y Protege, las cuales requieren experticia en la representación del conocimiento [Kog03].

La conceptualización o elaboración de la ontología de métricas e indicadores propiamente dicha ha sido realizada teniendo en cuenta los siguientes pasos en forma iterativa e incremental:

1. Definir el glosario de conceptos a partir de las fuentes de conocimiento citadas, teniendo en cuenta los objetivos y el alcance de la ontología.

2. Definir las interrelaciones semánticas entre dichos conceptos representándolas mediante un diagrama de clases UML, a la vez que se elabora una tabla de clases y de relaciones con las distintas clases que se van identificando.

3. Analizar los conceptos relacionados para identificar las partes comunes a dos o más conceptos. Decidir si estas partes son a su vez conceptos y, en su caso, incluirlos en el glosario de conceptos.

4. Identificar los atributos de cada concepto y construir la tabla de atributos, incluyéndolos a la vez al diagrama de clases UML.

5 Identificar y definir los axiomas estructurales y no estructurales que aporten conocimiento adicional a la ontología.

6. Comprobar la completitud y corrección de todas las tablas, analizando el modelo conceptual UML, los objetivos y alcance de la ontología, la correspondencia con las fuentes de información citadas y además analizando y resolviendo conflictos de interpretación conceptual.

5.3.2.2 Descripción Conceptual de la Medición y Evaluación

En esta sección presentamos una descripción conceptual mostrando qué se entiende por métrica, indicador, medición y demás conceptos de la ontología de métricas e indicadores, para un mejor entendimiento de los términos. Es decir, presentamos las estructuras y relaciones existentes entre las necesidades de información de un proyecto de evaluación y los elementos relevantes: entidades, atributos, métricas, métodos de cálculo y de medición, etc., que permiten obtener los informes que satisfagan dichas necesidades de información.

Debido a que muchas de las fuentes de información para la construcción de la ontología están escritas en idioma Inglés, hemos considerado más ilustrativo indicar los términos de la ontología en inglés, entre paréntesis, al lado de su correspondiente término en español.

La figura 5.1 muestra los conceptos y relaciones de la ontología de métricas e indicadores, en un diagrama UML, que describimos a continuación.

La selección o definición de las métricas adecuadas para satisfacer una necesidad de información (**Information Need**), comienza con la especificación de un concepto medible o calculable (**Calculable Concept)** a ser evaluado o estimado.

Un concepto calculable es una idea de cuáles son los atributos relacionados con dicho concepto para satisfacer una necesidad de información y cómo éstos están relacionados entre sí. Es decir, un concepto calculable es una relación abstracta entre atributos de entidades y necesidades de información.

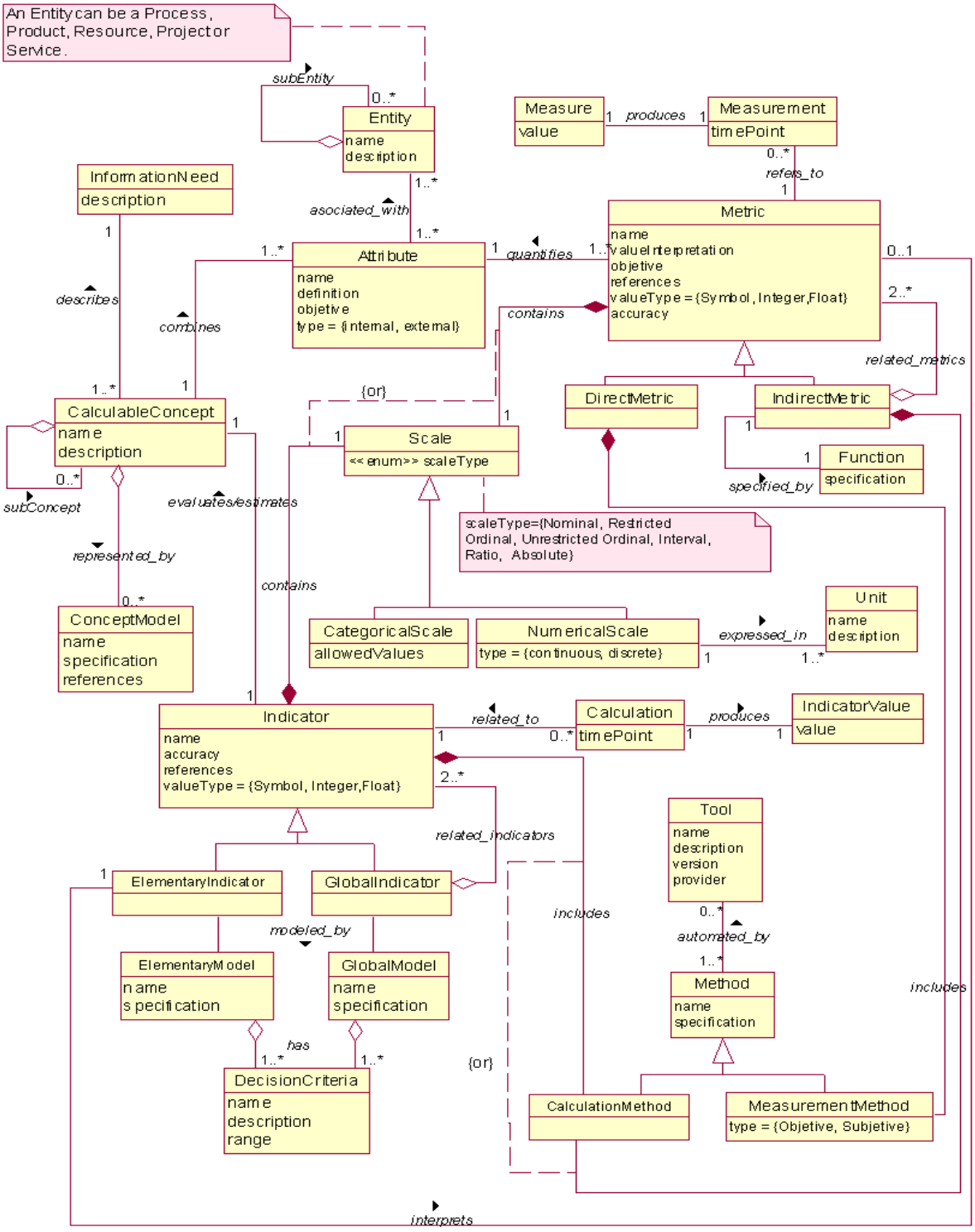

Fig. 5.1. Diagrama de clases UML modelando el dominio de Métricas e Indicadores

Por ejemplo, una necesidad de información podría ser "comparar la productividad de grupos de desarrollo en un proyecto de software", en este caso, el concepto calculable sería "Productividad". Y la entidad (**Entity**) sobre el que se aplica sería el "grupo de desarrollo". Para evaluar este concepto, podría requerirse medir atributos

como: el tamaño de los productos de software producidos, la cantidad de recursos requeridos, etc., dependiendo del modelo del concepto (**Concept Model**) elegido para "Productividad".

Otros ejemplos de conceptos calculables podrían ser calidad, accesibilidad, costo, entre otros. Considerando el nivel de abstracción del concepto calculable, puede ser descompuesto en sub-conceptos, conformando una jerarquía de conceptos calculables representada por un modelo de concepto (**Concept Model**). Por ejemplo, el estándar ISO 9126-1 especifica un modelo de calidad con características y subcaracterísticas. Uno o más atributos de entidades se asocian a cada concepto calculable.

Una entidad (**Entity**) es un objeto tangible o intangible, que se caracteriza mediante la medición de sus atributos. Tipos de entidades de interés en Ingeniería de Software podrían ser proyectos, productos, servicios, procesos o recursos (por ejemplo, módulos de programas, grupos de desarrollo, servicio de mantenimiento, podrían ser entidades a las cuales se les desea medir sus atributos).

Un atributo (**Attribute**) es una propiedad medible de una entidad. Una entidad puede tener muchos atributos, pero sólo algunos de ellos podrían ser de interés para evaluar un concepto calculable.

Un atributo de una entidad puede ser cuantificado por una o varias métricas (**Metric**). Cada observación o medición (**Measurement**) consiste en asignar (obtener) el valor de un atributo, en un determinado instante o punto en el tiempo, usando la definición de una de sus métricas asociadas. El valor de atributo obtenido es llamado medida (**Measure**).

Por lo tanto, una métrica **m** representa una correspondencia m: **A -> X**, siendo la flecha, la función de correspondencia, y que conlleva homomorfismo. **A** corresponde al atributo del mundo empírico; **X** es la variable (correspondiente al mundo formal) a la cual se le asignara el valor categórico o numérico que cuantifica al atributo. Para poder llevar a cabo esta correspondencia, se necesita una definición precisa de una actividad de medición específica, que establezca explícitamente un método y una escala (es decir se necesita la definición de la métrica).

Las métricas pueden ser directas o indirectas. Las métricas directas (**Direct Metric**) son aquellas cuya definición no depende de otras métricas. Las métricas indirectas (**Indirect Metric**), en cambio, dependen de otras métricas y están especificadas por una función (**Function**) de cálculo.

Cada métrica está definida por un determinado método (**Method**) que puede ser un método de medición (**Measurement Method**) (para el caso de las métricas directas), o un método de cálculo (**Calculation Method**) (para el caso de las métricas

indirectas). Estos métodos pueden estar soportados por herramientas de software (**Software Tool**) que automaticen el proceso de recolección de datos y/o cálculo.

Además cada métrica está asociada a una determinada escala (**Scale**). Las escalas pueden ser de 5 tipos diferentes: absoluta, nominal, ordinal, intervalo y proporción. Además las escalas se clasifican en dos grandes categorías: escala categórica (**Categorical Scale**) cuando los valores son categóricos, ya sean ordenados o no, y no pueden ser expresados en una unidad; y por otro lado escala numérica (**Numerical Scale**), cuando los valores son numéricos. En este último caso, la escala está expresada en una determinada unidad (**Unit**).

Para evaluar (o estimar) un concepto calculable se usa un indicador (**Indicator**). Y aquí conviene hacer la distinción semántica entre métrica e indicador. En párrafos anteriores habíamos señalado que una métrica representa una correspondencia (mapeo) entre un atributo del mundo empírico y una variable en el mundo formal. El indicador representa un nuevo mapeo que va del valor de una métrica (mundo formal), a otra variable de un nuevo mundo formal, a la cual se le asignan valores numéricos que representan la interpretación de dicho valor de métrica y que conlleva criterios de decisión para una necesidad o requerimiento específico de usuario. Por lo tanto, la correspondencia para indicadores elementales va de **X -> Ie** (**X** es la variable que tendrá el valor de la métrica; **Ie** es la variable que tendrá el valor del indicador elemental).

Un indicador especifica un método de cálculo y una escala, además de un modelo de análisis conteniendo criterios de decisión, para proveer una estimación o evaluación de un concepto calculable con respecto a una necesidad de información.

El valor de un indicador (**Indicator Value**) se obtiene aplicando un cierto método de cálculo y escala (definición de indicador), para interpretar los valores de una o varias métricas utilizando determinados criterios de decisión (**Decision Criteria**). A esta actividad de obtener un valor de indicador para un concepto calculable usando una determinada definición de indicador la llamamos Cálculo (**Calculation**).

Los indicadores pueden ser elementales o globales. Los indicadores elementales (**Elementary Indicator**) son aquellos que no dependen de otros indicadores para evaluar o estimar un concepto calculable y están modelados por un modelo de análisis elemental (**Elementary Model**). Los indicadores globales (**Global Indicator**) en cambio, dependen de otros indicadores y están modelados por un modelo de análisis global (**Global Model**).

Para ilustrar mejor estos conceptos, en la siguiente sección mostramos un ejemplo sencillo de una necesidad de información requerida y las métricas e indicadores relacionados.

5.3.2.3 Descripción Práctica a través de un Ejemplo

En este apartado desarrollamos un ejemplo concreto de aplicación de la descripción conceptual mostrada en el punto anterior. Para ello supongamos que una empresa dedicada al desarrollo de software quiere evaluar la eficiencia en desarrollo de sus grupos de trabajo. En este caso la **necesidad de información** sería "Evaluar la eficiencia en desarrollo de un grupo de trabajo", el **concepto calculable** es "Eficiencia" y la **entidad** es el grupo de desarrollo (en este caso un recurso).

Como podría ser de interés considerar distintos aspectos de la eficiencia en desarrollo, se elabora un **modelo de concepto** para el concepto calculable *Eficiencia*, que se muestra en la figura 5.2

1. *Eficiencia en Desarrollo*
   1.1. Productividad de Desarrollo (PD)
   1.2. Completitud de Desarrollo (CD)

Fig. 5.2. Un modelo de concepto para el concepto calculable Eficiencia.

A un concepto calculable se le puede asociar uno o más atributos de entidades, para nuestro ejemplo, los atributos de interés son: *Duración del desarrollo, Tamaño de los módulos desarrollados, Cantidad de módulos solicitados, Cantidad de módulos desarrollados, Productividad de desarrollo y Completitud de desarrollo.*

Por otra parte, para cada atributo dado existe al menos una relación empírica de interés que puede ser representada en el ámbito formal, por medio de una **métrica**. Para el caso de nuestro ejemplo, se consideran las siguientes **métricas directas**:

1. *Cantidad de horas de desarrollo* (#HD)
2. *Cantidad de líneas de código fuente del módulo* (#LDCF)
3. *Cantidad de módulos solicitados* (#MS)
4. *Cantidad de módulos completados* (#MC)

Y las siguientes **métricas indirectas**:

5. *Líneas de código producidas por hora,* (LCPH) que podría especificarse como el cociente entre la sumatoria de la *Cantidad de líneas de código fuente* de todos los módulos desarrollados y *Cantidad de horas de desarrollo* consumidas.

6. *Completitud de Desarrollo,* (CD) que podría especificarse como el cociente entre la *Cantidad de módulos solicitados* y *Cantidad de módulos completados*.

Cada métrica contiene información del **método de medición** (para el caso de una métrica directa) o el **método de cálculo** (para el caso de una métrica indirecta) definido y la **escala** que usa.

Por ejemplo para la métrica *Cantidad de líneas de código fuente del módulo,* el método de medición puede ser: "Contar el número de líneas de código fuente del módulo, sin considerar las líneas en blanco ni las líneas comentadas". Este es un método de medición de tipo objetivo.

Por otra parte, para la métrica indirecta *Completitud de Desarrollo,* el método de cálculo puede ser: "Evaluar la **función**: (#MC) * 100 / (#MS)". Ambos métodos podrían estar automatizados por una o varias **herramientas de software.**

Para las métricas *1,2,3 y 4,* la escala es absoluta, y las unidades son: horas, líneas de código fuente (LDCF), módulos y módulos respectivamente.

Para la métrica *Líneas de código producidas por hora* la escala es proporción (desde 0 a infinito), y la unidad es líneas de código fuente por hora (LDCF / H).

Para la métrica *Completitud de Desarrollo* la escala es proporción y la unidad es porcentaje de módulos completados (%MC).

Para evaluar el concepto calculable *Eficiencia*, el **indicador global** podría ser *Valoración de Eficiencia en desarrollo*. Como habíamos señalado en la sección anterior, un indicador además de especificar un **método de cálculo** y una escala, está asociado a un modelo de análisis conteniendo criterios de decisión, para proveer una estimación o evaluación de un concepto calculable con respecto a una necesidad de información.

Un indicador global puede estar compuesto de **indicadores elementales** (IE), en nuestro ejemplo podemos definir un indicador elemental para cada sub-concepto calculable, por ejemplo para el sub-concepto *Productividad de Desarrollo* se puede definir el indicador elemental *Preferencia en Productividad de Desarrollo* (PPD), que interprete la métrica *Líneas de código producidas por hora* (LCPH), y la especificación de su modelo elemental podría ser:

PPD = 100% si LCPH > Xmax ;

PPD = 0% si LCPH<Xmin ;

En otro caso

*PPD =((LCPH - Xmin) / (Xmax - Xmin)) * 100 if 0 <= LCPH <= X max*

donde Xmax es un umbral superior acordado, como por ejemplo 20

Los criterios de decisión que tiene un modelo de un indicador son los niveles del grado de aceptabilidad en una escala dada; por ejemplo podría ser: *No satisfactorio* si el rango es de 0 a 40, *Marginal* si es mayor que 40 y menor que 60, y *Satisfactorio* en otro caso.

Para el sub-concepto *Completitud de Desarrollo*, de la Fig. 5.2, se puede definir un modelo de indicador elemental similar, y sus criterios de decisión.

Un **indicador global** (IG) que evalúe el concepto calculable *Eficiencia,* podría ser especificado con un **modelo global**, representado por ejemplo a través de una función de puntuación lineal aditiva como la siguiente [Ols02a]:

$$IG_1 = (P_1\, IE_1 + P_2\, IE_2);$$

Dado que los indicadores elementales (IE_i) están expresados en escala porcentual, se cumple que $0 <= IE_i <= 100$; y los pesos P_i deben ser tal que:

$P_1 + P_2 + ...+ P_n = 1$ (n=2 en nuestro ejemplo).

En el modelo aditivo previo, los pesos pueden modelar la importancia relativa de los atributos para un concepto o sub-concepto dado. Los indicadores son de fundamental importancia en Ingeniería de Software para la evaluación y comparación de recursos, productos y procesos de software y la toma de decisiones.

5.3.2.4 Definición de Términos y sus Relaciones

En este apartado definimos cada término de la ontología de métricas e indicadores que luego se presentan en una tabla de conceptos, (ver tabla 5.1) ilustrando cada concepto con un ejemplo y mostrando cómo se relaciona con otros conceptos.

5.3.2.4.1 Attribute (Atributo)

Definición

Una propiedad mensurable, física o abstracta, de una entidad [ISO145981].

Relaciones

- Uno o más atributos se asocian a una o más entidades que caracterizan.
- Un atributo es cuantificado por una o más métricas.
- Uno o más atributos se combinan en un concepto calculable.

Ejemplos

Ejemplos de atributos de la entidad: "*Página web*", pueden ser: *Tamaño de Imágenes, Tamaño del código HTML.*

Nota

Esta definición coincide con la del estándar ISO/IEC 14598-1:1999. El estándar ISO 9126 hace mención de atributos internos y externos, que en esta ontología esa clasificación no fue considerada.

5.3.2.4.2 Calculable Concept (Concepto Calculable)

Definición

Relación abstracta entre atributos de entidades y necesidades de información.

Relaciones

- Un concepto calculable combina uno o más atributos de entidades.
- Uno o más conceptos calculables son definidos con el objetivo de satisfacer una necesidad de información concreta.
- Un concepto calculable es especificado por uno o más modelos de concepto.
- Un concepto calculable es evaluado o estimado por un indicador.

Ejemplos

Ejemplos de conceptos calculables son: *Usabilidad (de un producto de software)*, *Productividad (de un grupo de desarrollo), Calidad, Costo.*

Nota

Esta definición coincide con la del término "measurable concept" (concepto medible) del estándar ISO 15939:2002. La elección el término "Calculable Concept" en lugar de "measurable concept", está fundamentada en el hecho de que la evaluación o estimación de un concepto calculable está basada en una actividad de cálculo más que de medición, como se mostró en el ejemplo del punto 5.3.2.3 de conceptualización (ver también las diferencias entre las definiciones de "Calculation Method" 5.3.2.4.4 y "Measurement Method" 5.3.2.4.21). Estos dos términos se los podría considerar como sinónimos [Ols04].

5.3.2.4.3 Calculation (Cálculo)

Definición

Actividad que usa una definición de indicador para producir un valor de indicador

Relaciones

- Cada cálculo produce un valor de indicador específico.
- Cada cálculo está relacionado a la descripción de un indicador que usa para llevar a cabo la actividad.

Ejemplos

- Actividad que consiste en usar la especificación del indicador "*Preferencia en Productividad de Desarrollo"* (ver ejemplo de la sección 5.3.2.3), para obtener un valor en el campo numérico (formal), que estime o evalúe el atributo *Productividad de un Grupo de Desarrollo*. Esta acción dará como resultado un valor de indicador.

5.3.2.4.4 Calculation Method (Método de Cálculo)

Definición

Una particular secuencia de operaciones lógicas especificadas, para permitir la ejecución de una fórmula o descripción de indicador a través de un cálculo.

Relaciones

- Un método de cálculo específico es incluido en la descripción de un indicador.
- Un método de cálculo específico es incluido en la descripción de una métrica indirecta.
- Un método de cálculo es un tipo de método.

Ejemplos

Para calcular la métrica indirecta "*Tamaño promedio de los módulos*" (TPM). El método de cálculo es:

1. Medir el tamaño de cada módulo con la métrica #LCF (Cantidad de líneas de código fuente).
2. Realizar la sumatoria de todos los tamaños de módulos obtenidos = #LCFT (Cantidad de líneas de código fuente total).
3. Medir la cantidad de módulos del sistema con la métrica #M (Cantidad de módulos).
4. Evaluar la fórmula TPM = #LCFT / #M para obtener el tamaño promedio de los módulos.

Nota

Un método de cálculo está basado en métodos numéricos y es totalmente objetivo (si bien puede usar como datos de entrada valores de medidas obtenidas con tipos de métodos subjetivos). Usa como entrada datos obtenidos por otros cálculos o mediciones. Esto lo distingue de un método de medición, que no se basa en otros datos de entrada calculados, sino sobre una observación del atributo de la entidad que mide.

5.3.2.4.5 Categorical Scale (Escala Categórica)

Definición

Una escala donde los valores medidos o calculados son categorías y no se pueden expresar en unidades en un sentido estricto.

Relaciones

- Una escala categórica es un tipo de escala

Ejemplos

- Los valores (categorías) que puede tomar el atributo "Lenguaje de Programación" son por ejemplo: Pascal, C, Java. Esta escala es Nominal.
- Valores del nivel de madurez CMM: 1,2,3,4,5. Esta escala es Ordinal.

5.3.2.4.6 Concept Model (Modelo de Concepto)

Definición

El conjunto de sub-conceptos y las relaciones entre ellos, que proveen la base para especificar el requerimiento de concepto y su evaluación o estimación como un todo.

Relaciones

- Un modelo de concepto representa (modela) un concepto calculable

Ejemplos

- Modelo de calidad para productos software de ISO 9126, o modelo de calidad en uso del mismo estándar.

5.3.2.4.7 Decision Criteria (Criterios de Decisión)

Definición

Valores umbrales, objetivos o patrones, usados para determinar la necesidad de una acción o investigación posterior, o para describir el nivel de confianza de un resultado dado.

Relaciones

- Criterios de decisión están incluidos en un modelo de indicador.

Ejemplos

- Si PPD < 30 -> Productividad = “Baja”
- Si 30 <= PPD < 70 -> Productividad = “Normal”
- Si 70 < =PPD < 90 -> Productividad = “Alta”
- Si 90 <=PPD < =100 -> Productividad = “Muy alta”

Véase ejemplo de indicador elemental “*Preferencia en Productividad de Desarrollo* (PPD)” en sección 5.3.2.3.

Nota

El estándar ISO 15939 habla de nivel de “bondad” del valor de una métrica, con similar significado.

5.3.2.4.8 Direct Metric (Métrica Directa)

Definición

Una métrica de un atributo que no depende de una métrica de otro atributo.

Relaciones

- Un métrica directa incluye un método de medición específico.

Ejemplos

- LCF (cantidad de líneas de código fuente).
- ERI (cantidad de enlaces rotos internos).

Nota

El estándar ISO/IEC 14598-1:1999 define “direct measure” y no “direct metric”.

5.3.2.4.9 Elementary Indicator (Indicador Elemental)

Definición

Un indicador que no depende de otros indicadores para evaluar o estimar un concepto calculable.

Relaciones

- Un indicador elemental puede interpretar una métrica específica.
- Un indicador elemental es modelado por un modelo elemental.

Ejemplos

- Nivel de productividad (de un grupo de desarrollo).
- Nivel de completitud de desarrollo.

Nota

Este concepto no se encuentra especificado hasta el momento en ninguno de los estándares consultados. Está introducido en [Ols02a].

5.3.2.4.10 Elementary Model (Modelo Elemental)

Definición

Algoritmo o función con criterios de decisión asociados que modelan un indicador elemental.

Relaciones

- Un modelo elemental modela a un indicador elemental.
- Un modelo elemental tiene criterios de decisión.

Ejemplos

- El modelo elemental para el indicador *Preferencia en Errores por Módulo* (P_EM), podría ser:

  P_EM = 100% si #Errores = 0;
  P_EM = 0% si #Errores >0 ;

- Ver otro ejemplo en la sección 5.3.2.3

5.3.2.4.11 Entity (Entidad)

Definición

Un objeto que va a ser caracterizado mediante la medición de sus atributos [ISO15939].

Relaciones

- A una entidad se le pueden asociar uno o más atributos medibles.
- Una entidad puede estar compuesta por cero o varias sub-entidades, las cuales son a su vez entidades.

Ejemplos

- Un sitio web determinado.
- Una página web determinada (éste sería un ejemplo de sub-entidad que forma parte de la entidad "sitio web")
- El programa "HolaMundo.exe"
- El archivo "MisDatos.dat"

Notas

Esta definición coincide con la del estándar ISO/IEC 15939. Una entidad puede ser un proceso, un producto, un proyecto, un servicio o un recurso.

Una entidad puede ser tangible (por ejemplo una computadora) o abstracta por ejemplo un producto software.

5.3.2.4.12 Function (Función)

Definición

Algoritmo o fórmula que especifica cómo combinar dos o más métricas.

Relaciones

- Una función especifica una métrica indirecta.

Ejemplos

- HPT = $\sum_{dias}$ HPD (Horas de programador totales es la sumatoria de las horas de programador diarias durante los días de duración del proyecto).
- LCFH = LCF / HPT (La función de cálculo de Líneas de código fuente por hora es la simple división entre la cantidad de líneas de código fuente producidas y las horas de programador totales consumidas).

5.3.2.4.13 Global Indicator (Indicador Global)

Definición

Un indicador que se deriva de otros indicadores para evaluar o estimar un concepto calculable.

Relaciones

- Un indicador global se puede estructurar (componer) en base a otros indicadores relacionados.
- Un indicador global es modelado por un modelo global.

Ejemplos

- Calidad
- Costo

Nota

Este concepto no se encuentra especificado hasta el momento en ninguno de los estándares consultados. Está introducido en [Ols02a].

5.3.2.4.14 Global Model (Modelo Global)

Definición

Algoritmo o función con criterios de decisión asociados que modelan un indicador global.

Relaciones

- Un modelo global modela a un indicador global.
- Un modelo global tiene criterios de decisión.

Ejemplos

- El modelo global para el indicador "*Calidad Global"* puede ser un modelo aditivo de la forma:

$$CG = (P_1 IE_1 + P_2 IE_2);$$

Donde IE_i son los indicadores elementales asociados a *Calidad Global* y P_i son los pesos que modelan la importancia relativa de cada indicador elemental (ver ejemplo de la sección 5.3.2.3).

5.3.2.4.15 Indicator (Indicador)

Definición

El método de cálculo y la escala, además del modelo y los criterios de decisión definidos para proveer una estimación o evaluación de un concepto calculable, con respecto a una necesidad de información establecida.

Relaciones

- Un indicador evalúa (estima) un concepto calculable.
- Un indicador contiene una escala específica.
- Un indicador incluye un método de cálculo específico.
- Un indicador es usado por las actividades de cálculo.

Ejemplos

- Los ejemplos de indicadores elementales mostrados en la sección 5.3.2.4.9
- Los ejemplos de indicadores globales mostrados en la sección 5.3.2.4.13

5.3.2.4.16 Indicator Value (Valor de Indicador)

Definición

El número o categoría asignado a un concepto calculable por medio de la realización de un cálculo.

Relaciones

- Un valor de indicador específico es producido por una actividad de cálculo.

Ejemplos

- Nivel de productividad = 70%.
- Nivel de eficiencia =”Baja”
- Preferencia de calidad = 7 (en una escala de 0 a 10)

Nota

El valor de indicador sólo aparece cuando se realiza una actividad de cálculo, que estima un concepto calculable usando la definición (especificación) de un indicador. Esta actividad de cálculo interpretará el resultado de una métrica especifica (si se trata de un indicador elemental), o el resultado de otros indicadores (si se trata de un indicador global), de acuerdo a criterios asociados, produciendo un valor categórico o numérico para dicho concepto calculable.

5.3.2.4.17 Indirect Metric (Métrica Indirecta)

Definición

Una métrica de un atributo que se deriva de una o más métricas de otros atributos.

Relaciones

- Un métrica indirecta se puede estructurar en base a dos o más métricas relacionadas.
- Una métrica indirecta está especificada por una función (o fórmula) dada.
- Una métrica indirecta incluye un método de cálculo específico.

Ejemplos

- HPT (Cantidad total de horas de programador).
- LCFH (Cantidad de líneas de código fuente por hora).

Nota

El estándar ISO/IEC 14598-1:1999 define direct measure e indirect measure (en vez de direct o indirect metric). Sin embargo, y teniendo en cuenta las definiciones de un mismo estándar (ISO/IEC 14598-1:1999), los términos measure y metric no son sinónimos. Metric, está definido en [ISO14598] como "*the defined measurement method and the measurement scale*", mientras que el significado del término measure en el mismo estándar (significado adoptado en esta ontología) es "*the number or category assigned to an attribute of an entity by making a measurement*". Estas definiciones reflejan claramente el hecho de que una medida (measure) es el valor resultante (o producto de salida) de una actividad de medición, mientras que una métrica (metric) representa la definición específica y explícita de dicha actividad de medición.

5.3.2.4.18 Information Need (Necesidad de Información)

Definición

Información necesaria para gestionar objetivos, metas, riesgos y problemas [ISO15939].

Relaciones

- Una necesidad de información es descripta por uno o varios conceptos calculables.

Ejemplos

- Evaluar la eficiencia en desarrollo de un grupo de trabajo (ver ejemplo de sección 5.3.2.3).
- Evaluar si un producto de software cumple con los requerimientos de calidad del cliente.
- Estimar el costo de un proyecto de software.
- Evaluar la calidad global de un sitio web.

Nota

Esta definición coincide con la del estándar ISO/IEC 15939.

5.3.2.4.19 Measure (Medida)

Definición

El número o categoría asignado al atributo de una entidad, como resultado de hacer una medición [ISO14598].

Relaciones

- Una medida específica es producida por una actividad de medición.

Ejemplos

- 15 casos de uso.
- 0.2 errores por módulo.
- 10.000 líneas de código.
- 0,15 enlaces rotos por página.

Nota

La medida sólo aparece cuando se realiza una actividad de medición.

5.3.2.4.20 Measurement (Medición)

Definición

Actividad que usa la definición de una métrica para producir un valor medido.

Relaciones

- Una actividad de medición produce un valor medido específico.
- Una actividad de medición está relacionada a una descripción de métrica.
- Cero o más mediciones pueden ser realizadas usando una misma métrica.

Ejemplos

- Actividad consistente en usar la definición de la métrica directa "*Cantidad de líneas de código*" para obtener el valor del atributo "tamaño", del módulo "ventas.c".

5.3.2.4.21 Measurement Method (Método de Medición)

Definición

Una particular secuencia de operaciones lógicas y posibles heurísticas, especificadas para permitir la realización de una descripción de métrica por parte de una actividad de medición.

Relaciones

- Un método de medición está incluido en una métrica directa específica.

Ejemplos

- A continuación especificamos el método de medición para medir el atributo "cantidad de enlaces rotos" de la entidad "sitio web".

Descripción:

Recorrer el sito recursivamente, comenzando desde la página principal y analizando todos los enlaces de las páginas. Para cada enlace, ejecutar un comando GET del protocolo HTTP, si el comando devuelve un error 404 o 405, incrementar en uno la cantidad de enlaces rotos.

Algoritmo en pseudocódigo:

```
Precondiciones: URL = URL de la página inicial del sito Web a analizar
Enlaces_rotos (URL): Cantidad_Enlaces_Rotos
Enlaces_rotos = 0
Obtener la pagina URL
    Para cada enlace URLi de_la_pagina
        Si GET(URLi) devuelve error entonces
        Enlaces_rotos = Enlaces_rotos +1
    sino
        Enlaces_rotos = Enlaces_rotos(URLi)
        Fin si
    Fin para
Retornar (Enlaces_rotos )
Fin
```

Nota

Los métodos de medición pueden ser subjetivos u objetivos. **Subjetivo:** cuando el método de medición supone un juicio realizado por un ser humano. **Objetivo:** cuando el método de medición está basado en métodos numéricos.

5.3.2.4.22 Method (Método)

Definición

Una secuencia lógica de operaciones y posibles heurísticas, especificadas en forma genérica, para permitir la realización de una descripción de actividad.

Relaciones

- Uno o más métodos pueden estar automatizados por una o varias herramientas de software.

Ejemplos

- Los ejemplos de métodos de cálculo mostrados en la sección 5.3.2.4.4
- Los ejemplos de métodos de medición mostrados en la sección 5.3.2.4.21

5.3.2.4.23 Metric (Métrica)

Definición

El método de cálculo o de medición y la escala de medición definidos.

Relaciones

- Una o más métricas pueden cuantificar un atributo.
- Una métrica contiene una escala específica.
- Una métrica incluye un método de cálculo o de medición específico.
- Una métrica es referida por cero o más actividades de medición.
- Una métrica puede ser interpretada por un indicador elemental.
- Dos o más métricas relacionadas pueden estar estructuradas en una métrica indirecta.

Ejemplos

- Los ejemplos de métricas directas mostrados en la sección 5.3.2.4.8
- Los ejemplos de métricas indirectas mostrados en la sección 5.3.2.4.17

Notas

Esta definición es casi idéntica a la del estándar ISO 14598-1:1999: "*the defined measurement method and the measurement scale*". Se agregó "método de **cálculo** " a la

definición del estándar, porque en esta ontología distinguimos un método de cálculo de un método de medición. Una métrica directa usa siempre un método de medición, mientras que una métrica indirecta usa un método de cálculo, (una métrica indirecta está representada por una función o fórmula que especifica cómo combinar otras métricas).

Los estándares ISO 9126 e ISO 14598-1:1999 hacen referencia a métricas internas o externas. En esta ontología no se hace esa distinción, como tampoco se distingue entre atributos internos o externos.

5.3.2.4.24 Numerical Scale (Escala Numérica)

Definición

Una escala donde los valores medidos o calculados son números que pueden ser expresados en unidades en un sentido estricto.

Relaciones

- Una escala numérica es un tipo de escala.
- Una escala numérica está expresada en una unidad.

Ejemplos

- La temperatura expresada en grados centígrados.
- El tamaño de una imagen expresada en pixels.

5.3.2.4.25 Scale (Escala)

Definición

Un conjunto de valores con propiedades definidas [ISO14598].

Relaciones

- Una escala específica es parte de una métrica o indicador.

Ejemplos

- Los ejemplos de escalas categóricas mostrados en la sección 5.3.2.4.5.
- Los ejemplos de escalas numéricas mostrados en la sección 5.3.2.4.24.

5.3.2.4.26 Software Tool (Herramienta de Software)

Definición

Es una herramienta que automatiza total o parcialmente un método de medición o de cálculo.

Relaciones

- Una o más herramientas de software pueden automatizar uno o más métodos.

Ejemplos

- Una herramienta CASE que sirva para contar líneas de código, líneas comentadas, cantidad de procedimientos definidos, cantidad de invocaciones a procedimientos internos y cantidad de invocaciones a procedimientos externos.

5.3.2.4.27 Unit (Unidad)

Definición

Una cantidad particular, definida y adoptada por convención, contra la cual son comparadas otras cantidades de la misma clase, para expresar sus magnitudes en relación a esa cantidad [ISO15939].

Relaciones

- Una escala numérica debe ser expresada en una unidad.

Ejemplos

- Centímetros cuadrados, pixels.
- Líneas de código.
- Módulos.
- Microsegundos, minutos, horas.

Nota

Esta definición coincide con la del estándar ISO/IEC 15939.

En un sentido estricto, no existe la idea de unidad cuando la escala es categórica, por lo tanto las unidades sólo se relacionan a escalas numéricas.

5.3.2.5 Representación de la Ontología

Luego de haber definido los conceptos y relaciones de la ontología, se identificaron los atributos de cada concepto y se construyó una tabla de atributos, incluyéndolos a la vez al diagrama de clases UML (ver figura 5.1).

De esta manera y como resultado de la etapa de conceptualización, se obtuvo una primera versión de la ontología de métricas e indicadores que fue semi-formalizada usando algunas representaciones intermedias, independientes del dominio y de su implementación, como son el *Diccionario de Términos*, la *Tabla de Relaciones* y la ya mencionada *Tabla de Atributos* que mostramos a continuación en las tablas 5.1, 5.2 y 5.3 respectivamente.

Tabla 5.1 Diccionario de Términos de la Ontología de Métricas e Indicadores

Concepto	Descripción
Attribute	Una propiedad mensurable, física o abstracta, de una entidad [ISO14598].
Calculable Concept	Relación abstracta entre atributos de entidades y necesidades de información [ISO15939].
Calculation	Actividad que usa una definición de indicador para producir un valor de indicador.
Calculation Method	Una particular secuencia de operaciones lógicas especificadas, para permitir la ejecución de una fórmula o descripción de indicador a través de un cálculo.
Categorical Scale	Una escala donde los valores medidos o calculados son categorías y no se pueden expresar en unidades en un sentido estricto.
Concept Model	El conjunto de sub-conceptos y las relaciones entre ellos, que proveen la base para especificar el requerimiento de concepto y su evaluación o estimación como un todo.
Decision Criteria	Valores umbrales, objetivos o patrones, usados para determinar la necesidad de una acción o investigación posterior, o para describir el nivel de confianza de un resultado dado.
Direct Metric	Una métrica de un atributo que no depende de una métrica de otro atributo.
Elementary Indicator	Un indicador que no depende de otros indicadores para evaluar o estimar un concepto calculable.
Elementary Model	Algoritmo o función con criterios de decisión asociados que modelan un indicador elemental.

Concepto	Descripción
Entity	Un objeto que va a ser caracterizado mediante la medición de sus atributos [ISO15939].
Function	Algoritmo o fórmula que especifica cómo combinar dos o más métricas.
Global Indicator	Un indicador que se deriva de otros indicadores para evaluar o estimar un concepto calculable.
Global Model	Algoritmo o función con criterios de decisión asociados que modelan un indicador global.
Indicator	El método de cálculo y la escala, además del modelo y los criterios de decisión definidos para proveer una estimación o evaluación de un concepto calculable, con respecto a una necesidad de información establecida.
Indicator Value	El número o categoría asignado a un concepto calculable por medio de la realización de un cálculo.
Indirect Metric	Una métrica de un atributo que se deriva de una o más métricas de otros atributos.
Information Need	Información necesaria para gestionar objetivos, metas, riesgos y problemas [ISO15939].
Measure	El número o categoría asignado al atributo de una entidad, como resultado de hacer una medición [ISO14598].
Measurement	Actividad que usa la definición de una métrica para producir un valor medido.
Measurement Method	Una particular secuencia de operaciones lógicas y posibles heurísticas, especificadas para permitir la realización de una descripción de métrica por parte de una actividad de medición.
Method	Una secuencia lógica de operaciones y posibles heurísticas, especificadas en forma genérica, para permitir la realización de una descripción de actividad.
Metric	El método de cálculo o de medición y la escala de medición definidos.
Numerical Scale	Una escala donde los valores medidos o calculados son números que pueden ser expresados en unidades en un sentido estricto.
Scale	Un conjunto de valores con propiedades definidas [ISO14598].
Software Tool	Es una herramienta que automatiza total o parcialmente un método de medición o de cálculo.
Unit	Una cantidad particular, definida y adoptada por convención, contra la cual son comparadas otras cantidades de la misma clase, para expresar sus magnitudes en relación a esa cantidad [ISO15939].

Tabla 5.2 Tabla de relaciones de la Ontología de Métricas e Indicadores

Name	Descripción
associated_with	Uno o más atributos medibles se asocian con una o más entidades.
automated_by	Uno o mas métodos pueden ser automatizados por ninguna o varias herramientas de software.

combines	Un concepto calculable combina (asocia) uno o más atributos medibles.
contains	Una métrica o un indicador contienen una escala especifica.
describes	Uno o mas conceptos calculables se definen para satisfacer una necesidad de información concreta. De esta manera, un concepto calculable describe una necesidad de información concreta.
evaluates/estimates	Un indicador evalúa / estima un concepto calculable.
expressed_in	Una escala numérica se debe expresar en una unidad especifica. (En un sentido estricto, no existe la idea de unidad para escalas categóricas).
has	Un modelo de indicador tiene uno o más criterios de decisión.
includes	Una métrica incluye un método de cálculo o de medición específico. Un indicador incluye un método de cálculo específico.
interprets	Un indicador elemental puede interpretar cero o una métrica.
modeled_by	Un indicador elemental (o global) se modela por medio de un modelo elemental (o global).
produces	Una actividad de medición (o cálculo) produce un valor de medida (o indicador) específico.
quantifies	Una o mas métricas cuantifican un atributo.
refers-to	Una actividad de medición se relaciona a una métrica. Se pueden realizar cero o varias mediciones basándose en la misma métrica.
related_indicators	Un indicador global se puede estructurar (agregar) en base a dos o más indicadores relacionados.
related_metrics	Una métrica indirecta se puede estructurar en base a dos o más métricas relacionadas.
related-to	Una actividad de cálculo se relaciona a un indicador (descripción). Se pueden realizar cero o varios cálculos en base al mismo indicador.
represented_by	Un concepto calculable puede ser representado por cero o varios modelos de concepto.
specified_by	Una métrica indirecta se especifica por medio de una función (o fórmula) dada.
subConcept	Un concepto calculable puede ser compuesto por cero o varios sub-conceptos, los cuales son a su vez conceptos calculables.
subEntity	Una entidad se puede componer de cero o varias sub-entidades, las cuales son a su vez entidades.

Tabla 5.3 Tabla de Atributos de la Ontología de Métricas e Indicadores

Concepto	**Atributo**	**Descripción**
Attribute	Name	Nombre de un atributo para ser identificado.
	Definition	Una descripción no ambigua del significado del atributo.

Concepto	**Atributo**	**Descripción**
	Objective	Objetivo o propósito de medir el atributo.
	Type	Tipo de atributo, puede ser interno o externo [ISO9126].
Calculable Concept	Name	Nombre de un concepto calculable para ser identificado.
	Description	Una descripción no ambigua del significado del concepto calculable.
Calculation	timePoint	Instante del tiempo en se realiza el cálculo.
Categorical Scale	allowedValues	Lista de literales que indican los valores válidos de una escala categórica.
Concept Model	Name	Nombre de un modelo de concepto para ser identificado.
	Specification	Una representación formal o semiformal de un modelo de concepto.
	References	Referencias a recursos bibliográficos o URLs, donde se puede consultar información adicional y autoritativa referente al modelo de concepto.
Decision Criteria	Name	Nombre de un criterio de decisión para ser identificado.
	Description	Una descripción no ambigua del significado del criterio de decisión.
	Range	Valores numéricos especificando por ejemplo, los limites inferior y superior para un criterio dado.
Elementary Model	Name	Nombre de un modelo elemental para ser identificado.
	Specification	Una representación formal o semiformal de un modelo elemental. Puede ser por ejemplo, una representación lógica o matemática.
Entity	Name	Nombre de una entidad para ser identificada.
	Description	Una descripción no ambigua del significado de la entidad.
Function	Specification	Una representación formal o semiformal de una función. Sinónimo: fórmula.
Global Model	Name	Nombre de un modelo global para ser identificado.
	Specification	Una representación formal o semiformal de un modelo global. Puede ser por ejemplo, una representación lógica o matemática.
Indicator	Name	Nombre de un indicador para ser identificado.
	Accuracy	Una cuantificación del nivel de confianza inherente a la manera en que se produce el valor de un indicador.
	References	Referencias a recursos bibliográficos o URLs, donde se puede consultar información adicional y autoritativa referente al indicador.
	valueType	Tipo de valor que un indicador puede tomar. Puede ser un símbolo, un número entero o punto flotante.

Concepto	Atributo	Descripción
Indicator Value	Value	Resultado numérico o categórico asignado a un indicador [ISO15939]. Sinónimo: dato.
Information Need	Description	Una descripción no ambigua del significado de la necesidad de información.
Measure	Value	Resultado numérico o categórico asignado a una métrica. Sinónimo: dato.
Measurement	timePoint	Instante del tiempo en se realiza la medición.
Measurement Method	Type	Indica el tipo de método de medición, que depende de la naturaleza de las operaciones usadas para cuantificar un atributo. Se pueden distinguir dos tipos: sujetivo (la cuantificación se basa en un juicio humano), y objetivo (la cuantificación se basa en cálculos numéricos) [ISO15939].
Method	Name	Nombre de un método para ser identificado.
	Specification	Una descripción formal o semiformal de un método.
Metric	Name	Nombre de una métrica para ser identificada.
	valueInterpretation	Una declaración textual no ambigua para ayudar a los distintos participantes a entender el significado del valor obtenido. Por ejemplo, cuanto más cercano a cero mejor.
	Objective	Objetivo o propósito de aplicar una métrica especifica.
	References	Referencias a recursos bibliográficos o URLs, donde se puede consultar información adicional y autoritativa referente a la métrica.
	valueType	Tipo de valor que una métrica puede tomar. Puede ser un símbolo, un número entero o punto flotante.
	Accuracy	Una cuantificación del nivel de confianza inherente a la manera en que se produce el valor de una métrica.
Numerical Scale	Type	Una escala numérica puede ser continua o discreta.
Scale	scaleType	El tipo de escala depende de la naturaleza de la relación entre los valores de la escala [ISO15939]. Los tipos de escala comunmente definidos son: nominal, ordinal (restringida o no restringida), intervalo, proporción o absoluta.
Software Tool	Name	Nombre de una herramienta de software para ser identificada.
	Description	Una descripción no ambigua de la herramienta de software
	Version	Número que indica la versión de la herramienta de software.
	Provider	Indica el nombre (o URL) del proveedor de la herramienta de software.

Concepto	Atributo	Descripción
Unit	Name	Nombre de una unidad para ser identificada.
	Description	Una descripción no ambigua del significado de la unidad.

5.3.2.5.1 Axiomas

En esta sección definimos un conjunto de axiomas que proveen conocimiento adicional, a la conceptualización realizada en la sección anterior para el dominio de métricas e indicadores. Estos axiomas expresan otras relaciones entre los conceptos y/o restringen su interpretación, agregando restricciones estructurales y no estructurales al diagrama conceptual.

Los axiomas serán expresados en lenguaje natural y en lógica de primer orden, donde los términos de la ontología se expresan a través de predicados unarios o binarios, según sean conceptos o relaciones, respectivamente.

Por ejemplo, el predicado *Métrica(m)* expresa que *m* es una instancia que pertenece a la clase métrica. El predicado *Cuantifica(m,a),* expresa que la métrica *m* está relacionada con el atributo *a* a través de la relación *Cuantifica*.

Axioma 1: Los conceptos Direct Metric e Indirect Metric son disjuntos.

$$(\forall m)\ (\ Direct\ Metric(m) \leftrightarrow \sim IndirectMetric(m)\ )$$

Axioma 2: Los conceptos Elementary Indicator y Global Indicator son disjuntos.

$$(\forall i)\ (\ Elementary\ Indicator(i) \leftrightarrow \sim Global\ Indicator(i)\ )$$

Axioma 3: Todo indicador elemental (instancia de Elementary Indicator), puede interpretar una métrica (Metric), siempre y cuando dicha métrica cuantifique un atributo combinado por el concepto calculable (Calculable Concept) que dicho indicador elemental evalúa.

$$(\forall i,m)\ (\ interprets(i,m) \rightarrow (\exists a,c)\ (\ quantifies(m,a) \wedge combines(c,a) \wedge evaluates/estimates(i,c)\ )\ )$$

Axioma 4: Los conceptos Calculation Method y Measurement Method son disjuntos.

$$(\forall m)\ Calculation\ Method(m) \leftrightarrow \sim Measurement\ Method(m)$$

Axioma 5: Un método de cálculo que es parte de una métrica indirecta no es al mismo tiempo parte de un indicador y viceversa.

$$(\forall mc,m)\ (\ Indirect\ Metric(m) \wedge includes(m,mc) \rightarrow \sim\ (\exists i)\ (\ Indicator(i) \wedge\ includes(i,mc)\ )$$

$$(\forall mc,i)\ (\ Indicator(i) \wedge\ includes(i,mc) \rightarrow \sim\ (\exists m)\ (\ Indirect\ Metric(m) \wedge\ includes(m,mc)\ )$$

Axioma 6: Los conceptos Numerical Scale y Categorical Scale son disjuntos.

$$(\forall s)\ Numerical\ Scale(s) \leftrightarrow \sim Categorical\ Scale(s)$$

Axioma 7: Una escala que es parte de una métrica no es al mismo tiempo parte de in indicador y viceversa .

$$(\forall s,m)\ (\ Metric(m) \wedge contains(m,s) \rightarrow \sim\ (\exists i)\ (\ Indicator(i) \wedge\ contains(i,s)\ )$$

$$(\forall s,i)\ (\ Indicator(i) \wedge contains(i,s) \rightarrow \sim\ (\exists m)\ (\ Metric(m) \wedge\ contains(m,s)\ )$$

5.3.3 Implementación

Teniendo en cuenta que uno de los objetivos de la ontología es constituir la base de conocimiento para la construcción de un catálogo de métricas e indicadores con capacidades de web semántica, y además, dado que es importante que el conocimiento capturado en la ontología pueda ser reusado con otros fines, o en otros sistemas, se eligió el lenguaje RDFS, para la implementación de la ontología, ya que es uno de los lenguajes estándar recomendados por el w3c para tal fin, (como fue analizado en el capítulo 3).

Como Chang expone en una nota publicada por el W3C [Cha98], el modelo RDFS es equivalente a un subconjunto del modelo de clases en UML. RDFS usa un DLG (Directed Labeled Graph, Grafo dirigido etiquetado) para describir los esquemas. Los esquemas de clases expresados en UML también pueden verse como DLGs. Si se usa un DLG para un esquema del modelo de clases de UML, puede demostrarse que el DLG de RDFS es isomórfico a un subgrafo del DLG del esquema de clases de UML [Cha98].

```
<rdfs:Class rdf:ID="Attribute">
   <rdfs:label xml:lang="en">Attribute</rdfs:label>
   <rdfs:comment>A measurable physical or abstract property of an entity </rdfs:comment>
   <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Metric">
   <rdfs:label xml:lang="en">Metric</rdfs:label>
   <rdfs:comment>The defined measurement or calculation method and the measurement
  scale.</rdfs:comment>
   <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="DirectMetric">
   <rdfs:label xml:lang="en">DirectMetric</rdfs:label>
   <rdfs:comment>A metric of an attribute that does not depend upon a metric of any other
  attribute.</rdfs:comment>
   <rdfs:subClassOf rdf:resource="#Metric"/>
</rdfs:Class>

<rdf:Property rdf:ID="Quantifies">
   <rdfs:label xml:lang="en">Quantifies</rdfs:label>
   <rdfs:domain rdf:resource="#Metric"/>
   <rdfs:range rdf:resource="#Attribute"/>
    <rdfs:comment>One or more metrics can quantify an attribute.</rdfs:comment>
</rdf:Property>
```

Figura 5.3 Trozo de código RDFS que implementa parte la ontología de Métricas e Indicadores (algunos conceptos y relaciones)

Las estructuras y elementos en el modelo de clases de UML y de RDFS se corresponden unos con otros. Diagrama de clases UML puede transformarse a una representación equivalente en un esquema RDFS y viceversa. La citada nota de Chang también describe la relación entre los elementos de los diagramas de clase para UML y RDFS, que fue tenida en cuenta para la implementación de la ontología.

En las figuras 5.3 y 5.4 se muestran, a modo de ilustración, trozos del código RDFS que implementan la ontología de métricas e indicadores. El código completo puede consultarse en el apéndice A.

Si observamos el modelo conceptual de la figura 5.1, para su implementación se define una clase en RDF para cada término (ver Fig. 5.3), y se define una propiedad para cada relación, con restricciones de rango y dominio sobre las clases origen y destino respectivamente.

Un problema que necesitó ser analizado, fue la implementación de los atributos de clases, que pueden ser vistos como propiedades RDFS que tienen como rango la clase a la que pertenecen y como dominio un valor literal.

Debido a que las propiedades RDFS son en sí mismas una clase, y por lo tanto no pueden definirse en relación a una clase, no se puede definir una propiedad dada, con un rango particular cuando se aplica a objetos de una clase, y otro rango cuando se aplica a objetos de otra clase distinta.

```
<!-- Atributos de Metric -->

<rdf:Property rdf:ID="metricName">
    <rdfs:label xml:lang="en">metricName</rdfs:label>
    <rdfs:domain rdf:resource="#Metric"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
    <rdfs:comment>Name of an metric to be identified.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="metricValueInterpretation">
    <rdfs:label xml:lang="en">metricValueInterpretation</rdfs:label>
    <rdfs:domain rdf:resource="#Metric"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
    <rdfs:comment>An unambiguous textual statement for helping stakeholders to understand the
   obtained value meaning, e.g. the closer to zero the better.</rdfs:comment>
</rdf:Property>

<!-- Atributos de Indicator -->

<rdf:Property rdf:ID="indicatorName">
    <rdfs:label xml:lang="en">indicatorName</rdfs:label>
    <rdfs:domain rdf:resource="#Indicator"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
    <rdfs:comment>Name of an indicator to be identified.</rdfs:comment>
</rdf:Property>


<rdf:Property rdf:ID="indicatorAccuracy">
     <rdfs:label xml:lang="en">indicatorAccuracy</rdfs:label>
     <rdfs:domain rdf:resource="#Indicator"/>
     <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
     <rdfs:comment>A quantification of the accuracy level inherent to the way of producing the value of
      an indicator.</rdfs:comment>
</rdf:Property>
```

Figura 5.4 Trozo de código RDFS que implementa parte de la ontología de Métricas e Indicadores (algunos atributos)

Se han discutido varias soluciones a este problema en la lista de mail del www-rdf-interest [Hau00]. La opción que se eligió en esta implementación es crear propiedades con nombres de la forma: claseAtributo, de esta manera, cada atributo es único para cada clase. En la figura 5.4 mostramos algunos ejemplos.

Finalmente, las restricciones impuestas por los axiomas se dejó para ser implementadas por el código de la aplicación del catálogo, no formando parte de la base de conocimientos especificada en RDFS, teniendo en cuenta las limitaciones de RDFS para especificar axiomas (como se analizó en el capítulo 3).

En un trabajo futuro se puede traducir la ontología a una nueva versión implementada en OWL, que es un lenguaje estándar recientemente recomendado por el W3C, y tiene mayor expresividad para implementar las restricciones de los axiomas.

5.4 Conclusiones

En este capítulo hemos definido la ontología de métricas e indicadores, resultante de un arduo trabajo de consenso entre investigadores y profesionales del área de Ingeniería de Software, de algunos grupos de investigación participantes.

Llegar a acordar la semántica de todos los conceptos relacionados a métricas e indicadores no fue una tarea fácil, es más, algunos conceptos debieron ser definidos sin el consenso total. Cabe destacar la importancia que tiene, contar con una ontología para el ámbito de métricas e indicadores ya que provee una base conceptual consensuada, para compartir, reusar y organizar toda la información relacionada a métricas e indicadores.

Cabe destacar, que si bien existen varios estándares de gran utilidad relacionados al modelo de calidad de software [ISO9136], al proceso de medición [ISO15939], y al proceso de evaluación [ISO14598], que proveen un consenso sobre la terminología usada, estos estándares no constituyen en sí mismos una ontología formal o semiformal. Además, existen términos que están definidos con distinto significado en distintos estándares, y algunos conceptos importantes relacionados a métricas e indicadores se encuentran totalmente ausentes en dichos documentos.

La ontología de métricas e indicadores de software que se discutió en este capítulo se basó (tratando de acercarse lo más que se pudo) en los términos definidos por los estándares de la ISO. Específicamente, de los veintisiete conceptos definidos en la ontología, ocho coinciden exactamente en su término y significado, son los términos: *attribute, decision criteria, entity, information need, measurable concept* -en nuestra ontología usamos la palabra *calculable* en lugar de *measurable* (medible) como fue discutido en la sección 5.3.2.4.2-, *measure, scale*, y *unit*. Además para el término *metric*

se especificó casi el mismo significado que el del estándar [ISO14598], como fue discutido en la sección 5.3.2.4.23.

Por otro lado, se observó la ausencia de varios términos importantes en los estándares citados, como por ejemplo los términos *elementary indicator* y *global indicator* (si bien el término *indicator* está definido en [ISO14598] con similar significado pero no igual a nuestra propuesta). Otros términos ausentes son: *calculation* y *calculation method* . A nuestro criterio, el objetivo de usar un método de medición, es distinto de usar un método de cálculo (como se mostró en el ejemplo del punto 5.3.2.3 de conceptualización). Los estándares no distinguen entre estos dos conceptos.

Otros términos introducidos en la ontología, que se consideran importantes son los conceptos de *categorical scale* y *numerical scale* (que no están explícitamente especificados en los estándares ISO). Esta distinción es importante para comprender el concepto de unidad asociado a una escala. Particularmente, una escala categórica es una escala donde los valores son categorías y no pueden ser expresados en unidades, en un sentido estricto. En una escala numérica, en cambio, los valores son números que deben ser expresados en unidades (ver sección 5.3.2.4.27).

Desde el punto de vista de la implementación, es importante destacar que las ontologías implementadas con lenguajes para la web semántica, representan una tecnología muy útil al propósito de proveer conocimiento "entendible" en forma automática por computadoras.

En el caso de nuestra ontología de métricas e indicadores, sirve además como base conceptual para compartir y organizar información obtenida como resultado de diversos trabajos de investigación, en el área de medición y evaluación de software, de una forma extensible, interoperable y automáticamente procesable.

La ontología propuesta, fue usada como base para el diseño y construcción de un sistema de catalogación basado en la Web Semántica, que permitirá explorar, reusar y compartir información relacionada a métricas e indicadores. En el próximo capítulo se mostrará la arquitectura, el diseño navegacional y la implementación de un prototipo de dicho sistema de catalogación.

Capítulo 6- Caso de Estudio: Sistema de Catalogación con Potencia de Web Semántica

6.1 Introducción

En la actualidad, las ontologías se han vuelto una tecnología clave para proveer entendimiento automático del conocimiento. Éstas, cuando están implementadas con lenguajes formales, interrelacionan el entendimiento humano de los términos, con la procesabilidad automática por parte de las computadoras.

En este capítulo presentamos un caso de estudio que se basa en la ontología de métricas e indicadores definida en el capítulo anterior, para la construcción del sistema de catalogación, con aspectos de procesamiento automático de su semántica [Mar02].

Este sistema de catalogación fue diseñado como una aplicación web, debido a la importancia que tiene hoy Internet, tanto para compartir el conocimiento como para su explotación. Además, las tecnologías que existen en la actualidad para la construcción de este tipo de aplicaciones, proveen el soporte necesario a un costo relativamente bajo, si se usan estándares abiertos.

En este sentido, y tal como fueron analizadas en el capítulo 3, es importante destacar las ventajas que ofrecen las nuevas tecnologías, que podríamos llamarlas estándares de facto, para la Web Semántica, en cuanto al procesamiento automático, la interoperabilidad y la búsqueda semántica en los sistemas basados en conocimiento. Por esta razón, el sistema de catalogación se implementará usando estas tecnologías, cuya potencialidad también será analizada en las próximas secciones.

Este capítulo está organizado como sigue: en la siguiente sección presentamos la arquitectura general del sistema de catalogación que se compone de dos subsistemas: el *Sistema Semántico de Consultas de Métricas e Indicadores* y el *Sistema de Revisión del Catálogo* (sección 6.2). El resto del capítulo está dedicado a mostrar detalladamente el diseño e implementación del *Sistema Semántico de Consultas*, para lo cual, en la sección 6.3 se analiza la importancia del sistema semántico de consultas y la necesidad de contar para su formalización e implementación con un lenguaje de consultas a nivel semántico. En las secciones 6.4 y 6.5 se describen las principales tecnologías usadas para la implementación del catálogo, a saber: el lenguaje de consultas RQL -sección 6.4, y la arquitectura Sesame -sección 6.5. En la sección 6.6 se discute la arquitectura del *Sistema Semántico de Consultas* mostrando cómo interactúan los distintos componentes usados en la construcción de un prototipo para consultar métricas e indicadores, en tanto que en la sección siguiente (6.7) se muestra el diseño navegacional para dicho prototipo. En las secciones 6.8 y 6.9 se muestran aspectos de implementación de la navegación y

búsqueda semántica respectivamente. Finalmente, en la sección 6.10, se remarcan las principales conclusiones sobre la potencia que los aspectos ontológicos y de procesamiento semántico le confieren al sistema de catalogación.

6.2 Arquitectura del Sistema de Catalogación

El sistema de catalogación de métricas e indicadores [Mar03a] provee, por un lado, un mecanismo colaborativo para discutir, consensuar, y agregar información de métricas e indicadores al repositorio, y por otro lado, funcionalidades de búsqueda y navegación semántica para propósitos de consulta y reuso. En el diseño de la arquitectura del sistema se pretende utilizar la potencia de las tecnologías de web semántica para definir las consultas sobre la información del catálogo y el uso de servicios web para simplificar y facilitar el acceso a la funcionalidad de búsqueda semántica, revisión y mantenimiento del catálogo.

Desde el punto de vista del diseño de las funcionalidades para los distintos tipos de usuarios, se definieron cuatro roles de usuario con distintas responsabilidades y privilegios de acceso, estos roles son: *Administrador, Moderador, Revisor y Usuario Registrado.*

El *Administrador* es el responsable final del Sistema de Revisión y tiene acceso total al catálogo. Está habilitado para agregar instancias de métricas y/o indicadores aprobadas por consenso al repositorio, y eliminar instancias si fuera necesario. Además, es el encargado del gerenciamiento y la coordinación de los moderadores (los que son responsables de discutir en los foros por potenciales métricas y/o indicadores).

El *Moderador* es el responsable de conformar y coordinar el grupo de revisores, a quienes someterá para su discusión, la métrica y/o indicador candidata y el cronograma de trabajo. Tanto el moderador como los revisores trabajaran sobre un área de datos temporal privada, con sus respectivos permisos de acceso. Se usaran mecanismos de colaboración centrados en la web, para facilitar la comunicación entre los miembros del grupo tanto en forma síncrona como asíncrona.

El *Revisor* es responsable de contribuir en la discusión y elaboración de la definición de las métricas y/o indicadores. Cada revisor tendrá derechos de acceso de lectura al área de revisores, y, en definitiva será el moderador del grupo quien administre la versión final de la métrica acordada, notificando al administrador de este evento, quien a su vez será el encargado de incluirla en el catálogo.

Con respecto al *Usuario Registrado*, puede ser tanto un ser humano, una aplicación de software (herramienta) o un agente automático, que usa los servicios de navegación y búsqueda de la información contenida en el repositorio. Los usuarios

humanos estarán habilitados para acceder al catálogo de métricas e indicadores por medio de las funcionalidades de búsqueda y navegación con permisos de acceso de sólo lectura. Los agentes y otras aplicaciones podrán acceder al repositorio de la misma manera, por medio de interfaces SOAP (Simple Object Access Protocol)[New02] , y servicios Web.

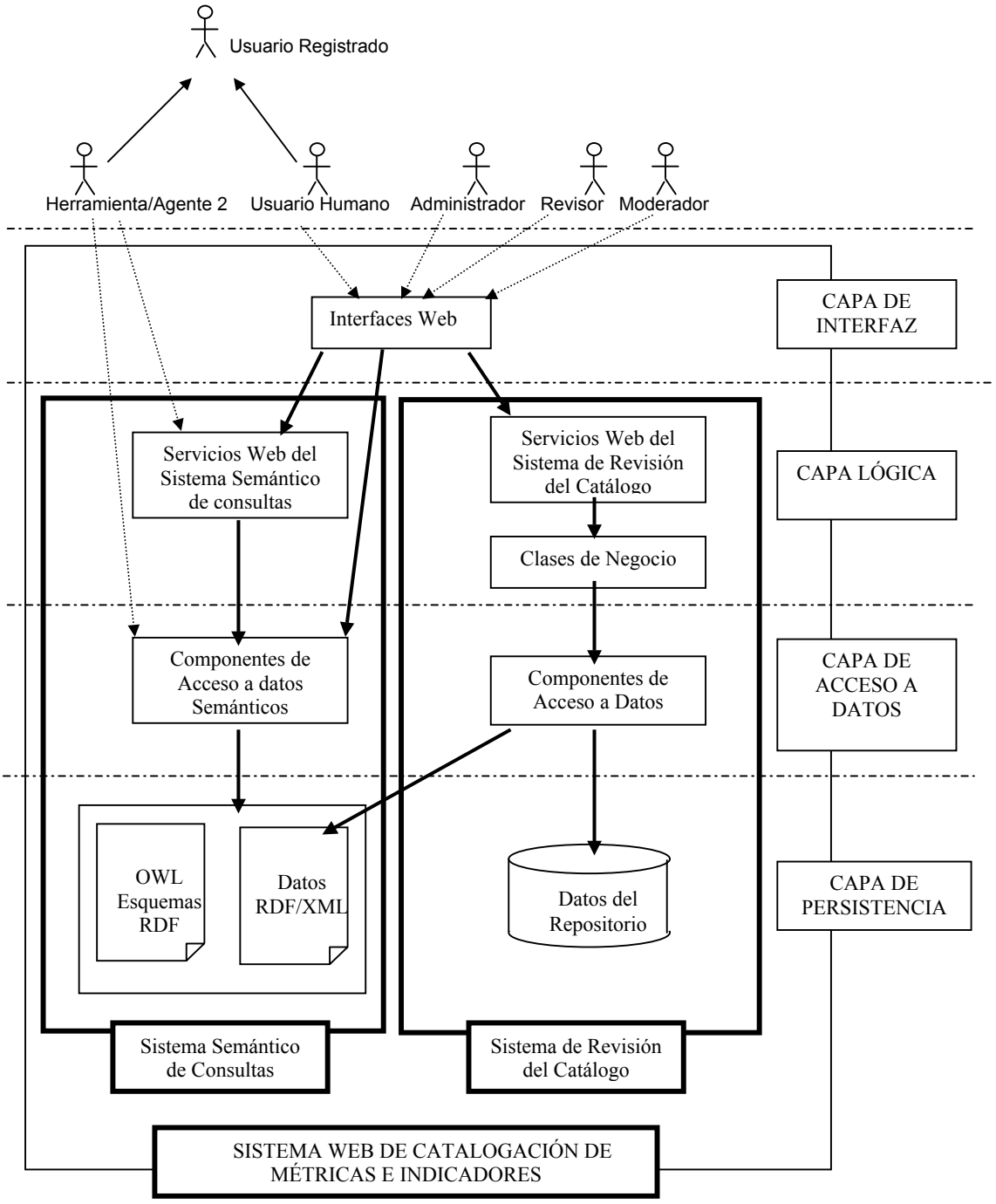

Figura 6.1 Arquitectura del Sistema Web de Catalogación de Métricas e Indicadores.

Para el diseño del sistema de catalogación, se ha elegido un estilo arquitectónico multinivel o de n-capas (ver Fig. 6.1). Este estilo arquitectónico posee como inconveniente su complejidad, pero a su vez proporciona a la aplicación una estructura clara y comprensible, siendo más fácilmente modificable y escalable. Las capas de la arquitectura son:

La capa de interfaz. Proporciona acceso personalizado al sistema para los distintos roles de usuario (Administradores, Revisores, Moderadores y Usuarios registrados). Para acceder a la funcionalidad, esta capa redirigirá las peticiones a la capa lógica .

La capa lógica. Implementa la funcionalidad que el sistema debe proporcionar a los distintos usuarios. Utiliza servicios web [Cur02, New02] para 'publicar' su funcionalidad. Estos servicios se implementan mediante tecnologías estándares (SOAP/HTTP, WSDL/XML y UDDI), facilitando y simplificando su uso a través de Internet por el propio sistema y por otras aplicaciones software, independientemente de la plataforma.

La capa lógica de acceso a datos. Se integra de un conjunto de componentes software que independizan y facilitan a las capas superiores el acceso a los datos del repositorio almacenados en la capa de persistencia.

La capa de persistencia. Almacena de manera persistente la información de métricas e indicadores.

Para estructurar de manera adecuada el sistema, éste ha sido dividido en dos subsistemas que son : *El Sistema de Revisión del Catálogo de Métricas e Indicadores,* y el *Sistema Semántico de Consultas de Métricas e Indicadores.*

6.2.1 Sistema de Revisión del Catálogo

El Sistema de Revisión del Catálogo de Métricas e Indicadores, es el encargado de gestionar y mantener el catálogo. Esta aplicación debe proporcionar funcionalidad fácilmente accesible desde la web para la revisión y aceptación de nuevas métricas y/o indicadores por los Administradores, Revisores y Moderadores, así como la funcionalidad para la extracción de datos por las Herramientas/Agentes. Se compone de las siguientes capas:

Capa lógica. Proporciona la funcionalidad requerida por los Administradores, Revisores, Moderadores y Herramientas/Agentes. Está dividida en dos paquetes: *Servicios Web del sistema de revisión del catálogo y Clases de Negocio.*

Servicios Web del sistema de revisión del catálogo es un conjunto de servicios que publicitan mediante tecnología estándar, la funcionalidad del sistema, haciéndola accesible desde cualquier plataforma. Estos servicios realizarán el control de privilegios de acceso en función del tipo de usuario y usarán las clases de negocio para resolver las peticiones de las capas superiores.

Las Clases de Negocio implementan la política de gestión del catálogo y proporcionan la funcionalidad necesaria para los usuarios del Sistema de Revisión. Usan los componentes de la capa lógica de datos para recuperar y almacenar los datos.

Componentes de acceso a datos. Proporciona un conjunto de componentes software basados en SQL para facilitar e independizar el acceso a los datos (capa de persistencia).

Datos del Repositorio. Base de datos específica construida mediante tecnología estándar basada en SQL que almacena el repositorio de métricas, indicadores e información adicional para su gestión.

6.2.2 Sistema Semántico de Consultas

El Sistema Semántico de Consultas de Métricas e Indicadores, es el encargado de publicar el catálogo haciendo uso de tecnologías de la web semántica. Este sistema no implementa funcionalidad de gestión del catálogo. Sin embargo, debe brindar la funcionalidad de búsqueda y navegación on-line de documentos y repositorios semánticos, que estén basados en la ontología de métricas e indicadores. Está compuesto de tres capas principales:

Capa lógica. Implementa las capacidades de consulta, búsqueda y navegación semántica a través de servicios web. Podrá ser utilizada por la capa de interfaz de usuario del sistema web y por otras aplicaciones software (representado en la fig. 6.1 por Herramienta/Agente 2).

Capa de acceso a los datos. Contiene un conjunto de componentes para acceder a distintos repositorios con información semántica de métricas e indicadores. Esta capa usa el núcleo de la arquitectura SESAME [Bro01a], para independizar la funcionalidad de acceso a los datos de la capa de persistencia.

Capa de persistencia. Conjunto de páginas web y documentos con información semántica de métricas e indicadores (especificados en OWL, RDF/RDFS y XML/XMLS).

En esta tesis nos centraremos en el diseño y la implementación de un prototipo, para el *Sistema Semántico de Consultas de Métricas e Indicadores*. En las secciones siguientes mostraremos más en detalle la arquitectura y las tecnologías usadas en este subsistema, y aspectos de navegación y búsqueda semántica del prototipo.

6.3 Descripción del Sistema Semántico de Consultas

Una de las principales funcionalidades del Sistema de Catalogación de métricas e indicadores es la explotación de la información del repositorio por parte de agentes humanos o automáticos. Para implementar esta funcionalidad se usaron tecnologías de Web Semántica y un paquete de software abierto desarrollado recientemente para dar soporte a la búsqueda y navegación semántica, como es la arquitectura Sesame que será descripta en la sección 6.5. En esta sección mostraremos dichas tecnologías y cómo fueron usadas en el sistema de catalogación.

6.3.1 Navegación y Búsqueda Semántica

En las aplicaciones web tradicionales, la navegación y exploración de la información está dirigida por la interacción del usuario. Generalmente se le muestra al usuario la información de más alto nivel de la jerarquía del mapa navegacional; y él, interactivamente, elige qué parte quiere explorar más exhaustivamente. El usuario debe navegar a través de un camino en la jerarquía y tiene que encontrar por sí mismo lo que realmente está buscando. El usuario debe hacer una serie de elecciones antes de encontrar la información de su interés, desconociendo si está navegando por el camino correcto. Muchas veces debe volver atrás para intentar diferentes caminos alternativos a través de la jerarquía o grafo, hasta encontrar la información que busca o determinar que no existe. Si bien la mayoría de las aplicaciones web, estructuran la información en categorías para su navegación, ocurre que el punto de vista del diseñador del sistema no coincide necesariamente con el punto de vista del usuario, lo que resulta en visualizaciones confusas para el usuario final.

Una forma potencial de resolver este problema es proveer mecanismos de navegación basados en ontologías, las cuales proveen la base conceptual y el metamodelo consensuado, para estructurar la información conforme a su semántica. De esta manera se le puede brindar al usuario una forma más efectiva de navegación y exploración de toda la información disponible sobre el tópico de su interés.

Con respecto a la búsqueda de información, el problema es similar, los motores de búsqueda que existen actualmente en la web, recuperan una lista de los documentos que contienen las palabras claves ingresadas por el usuario, en forma exacta o aproximada.

El inconveniente en este caso, es que los resultados obtenidos de estas búsquedas suelen contener demasiados documentos, no todos relacionados al tópico de interés, y se torna difícil analizar todos los documentos recuperados a fin de encontrar lo que uno desea.

El problema es que la búsqueda por palabras claves cuando no se tiene en cuenta su semántica, puede resultar en la recuperación de muchos documentos no deseados. Por ejemplo, si un usuario quiere buscar información relacionada a indicadores para evaluar productos de software e ingresa la palabra clave "Indicadores", podría recuperar miles de documentos relacionados a indicadores económicos, sociales, análisis estadísticos, etc. Con el agravante de que no recuperaría los documentos que contengan sinónimos de las palabras buscadas y que sí podrían ser de su interés.

Las ontologías representan la tecnología adecuada para facilitar la búsqueda semántica y proveer "entendimiento" automático de la información, facilitando la explotación del conocimiento contenido en documentos y repositorios de una forma más eficiente.

El caso de estudio que se presenta en este capítulo hace hincapié en la búsqueda y navegación semánticas, basado en la ontología de métricas e indicadores desarrollada en el capítulo anterior.

6.3.2 Necesidad de un Lenguaje de Consultas para RDFS

Una vez que la ontología de métricas e indicadores fuera formalizada e implementada en el lenguaje RDFS, instancias de métricas, indicadores y demás información relacionada se codifica en el lenguaje de marcado provisto por la ontología, y se almacena en repositorios (documentos RDF/XML). Cabe destacar que el prototipo construido del sistema de catalogación permite recuperar información de métricas e indicadores tanto del propio repositorio, como de repositorios externos, distribuidos en la web.

Como ya fue analizado en el capítulo 3, se pueden considerar los documentos RDF y RDFS en tres niveles distintos de abstracción: nivel sintáctico, nivel estructural y nivel semántico [Dav03].

6.3.2.1 Consultas a un Nivel Sintáctico

A nivel **sintáctico**, los documentos RDF y RDFS son a su vez documentos XML, (si bien actualmente existen versiones de RDF que no están específicas en XML, no es el caso de implementación adoptada para nuestro sistema de catalogación). Por lo tanto

cualquier modelo RDF serializado en XML podría ser consultado (queried by) usando un lenguaje de consulta para XML (por ejemplo Xquery)[Cha01].

Sin embargo, este tipo de lenguajes que operan a nivel sintáctico, desconocen el modelo de datos RDF que se encuentra subyacente en la sintaxis, y que es diferente de la estructura de árbol con nodos rotulados del modelo XML. Las relaciones en los datos RDF que no son advertidas en la estructura de árbol de XML son muy difíciles de consultar con lenguajes de consulta a nivel sintáctico.

Por ejemplo, consideremos el documento RDF/XML de la figura 6.2, cuyo modelo de datos RDF se muestra en la figura 6.3. Con un lenguaje de consultas que operan a nivel sintáctico del tipo de XQuery [Cha01], las consultas están limitadas a expresiones que atraviesen la estructura de árbol de XML, del tipo: "recuperar todos elementos contenidos en un elemento *Description* cuyo atributo **about** tenga el valor *Metrica1* (recupera todos los datos de la métrica *Metrica1*).

```
<?xml version="1.0" >
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:m="http://gidis.edu.ar/Esquema#">

  <rdf:Descritpion rdf:about="DensidadDeErrores">
    <rdf:type rdf:resource="http://gidis.edu.ar/Esquema#Attribute"/>
    <m:attibuteName>Densidad de Errores</m:attributeName>
  </rdf:Descritpion>

    <rdf:Descritpion rdf:about="Métrica1">
    <rdf:type rdf:resource="http://gidis.edu.ar/Esquema#IndirectMetric"/>
    <m:metricName>Errores por módulo </m:metricName>
    <m:Quantifies rdf:resource="http://gidis.edu.ar/datos#DensidadDeErrores"/>
  </rdf:Descritpion>

  </rdf:RDF>
```

Figura 6.2 Ejemplo de datos RDF serializados en XML.

Sin embargo, el modelo de datos RDF es un grafo, no un árbol; es más, tanto sus nodos (objetos) como sus arcos (propiedades) son rotulados. Las consultas sobre las **relaciones** entre recursos RDF, con un lenguaje de consulta a nivel sintáctico, quedan como un duro ejercicio para el constructor de la consulta, con el inconveniente adicional de que además debe conocer qué sintaxis (esquema RDF) se usó para codificar los datos en XML.

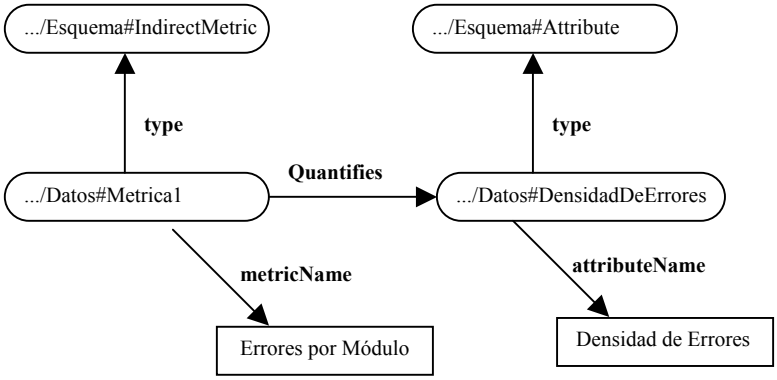

Figura 6.3 Ejemplo de grafo RDF.

Por ejemplo, si queremos consultar qué relación existe entre la métrica *Metrica1* y el atributo cuyo nombre es *Densidad de Errores*, uno tendría que formular la compleja consulta "Recuperar todos los elementos contenidos en un elemento **Description** cuyo atributo **about** tenga el valor *Metrica1,* y para los cuales el valor de su atributo **resource** ocurra como el valor del atributo **about** de otro elemento **Description,** que contiene un elemento attributeName cuyo valor es 'Densidad de Errores'".

6.3.2.2 Consultas a un Nivel Estructural

A nivel **estructural**, los documentos RDF y RDFS son un conjunto de tri-uplas que pueden ser fácilmente transformadas en tablas de una base de datos relacional para su procesamiento. Cada tri-upla representa una sentencia de la forma (Predicado, Sujeto, Objeto). Se han propuesto e implementado varios lenguajes de consulta que consideran a los documentos RDF como un conjunto de tri-uplas, y permiten la búsqueda de tales tri-uplas de varias maneras (ver http://perso.enst.fr/˜ta/web/rdf/rdf-query.html).

Si analizamos de nuevo el ejemplo de la figura 6.2, el modelo de datos se corresponde con el siguiente conjunto de tri-uplas:

```
(type  .../Datos#DensidadDeErrores  .../Esquema#Attribute)
(attributeName  .../Datos#DensidadDeErrores  "Densidad de Errores")
(type     .../Datos#Metrica1 .../Esquema#IndirectMetric)
(metricName  .../Datos#Metrica1 "Errores por módulo")
(Quantifies  .../Datos#Metrica1 .../Datos#DensidadDeErrores)
```

En un lenguaje de consulta RDF como por ejemplo Squish [Mil01], podríamos consultar qué métricas cuantifican al atributo DensidadDeErrores de la siguiente manera:

```
SELECT  ?x
FROM (Quantifies ?x DensidadDeErrores)
```

La ventaja de esta consulta es que tiene en cuenta el modelo de datos RDF, y por lo tanto es independiente de la sintaxis especifica que se usó para representar el dato. Sin embargo, una desventaja de los lenguajes de consulta a nivel estructural, es que interpretan cualquier modelo RDF sólo como un conjunto de tri-uplas, incluyendo aquellos elementos que tienen una semántica especial en RDFS, como son las propiedades subClassOf y type.

Por ejemplo, como *.../Datos#Metrica1* es de tipo *.../Esquema#IndirectMetric,* y además *.../Esquema#IndirectMetric* es una sub-clase de *.../Esquema#Metric ,* por lo tanto, *.../Datos#Metrica1* es también de tipo *.../Esquema#Metric,* en virtud de la semántica establecida para type y subClassOf . Sin embargo, la consulta:

```
SELECT  ?x
FROM (type ?x Metric)
```

no recuperara todas las métricas, porque la consulta sólo busca en las tri-uplas explícitamente guardadas en el almacenamiento. En este caso, la tri-upla (*type .../Datos#Metrica1 .../Esquema#Metric*) no está explícitamente guardada, a pesar de estar implícitamente interpretada por la semántica de RDFS. El lector puede notar, que si extendemos la consulta anterior a otra del tipo:

```
SELECT  ?x ?c1 ?c2
FROM (type ?x ?c1)
     (subClassOf ?c2 Metric)
WHERE ?c1 = ?c2
```

resolveríamos el problema para nuestro ejemplo específico, pero no sería una solución general, para cadenas de tri-uplas subClassOf de longitud indeterminada.

6.3.2.3 Consultas a Nivel Semántico

Se puede observar claramente, que para una explotación total del conocimiento almacenado en datos RDF/RDFS, se requiere un lenguaje de consulta a nivel semántico. Es decir, que sea sensible a la semántica de las primitivas RDFS, teniendo en cuenta todo el conocimiento que las descripciones RDF/RDFS implican, implícita o explícitamente, no sólo las sentencias aseveradas explícitamente.

RQL [Ale01, Kar01] es la primer propuesta de un lenguaje de consulta declarativo que explícitamente captura esta semántica en su diseño. RQL fue

desarrollado en el instituto ICS-FORTH (http://athena.ics.forth.gr:9090/RDF/), y su potencia semántica está basada en la evaluación de caminos de expresiones sobre grafos RDF, permitiendo el uso de variables tanto para denotar nombres de nodos (es decir, clases), como arcos (es decir propiedades). Una de las principales características de RQL, y que lo distingue de otros lenguajes de consulta RDF es su capacidad de consultar esquemas RDF y descripciones RDF (es decir, instancias) en una misma consulta.

RQL permite realizar consultas sobre esquemas RDF, tal como *Class* (recupera todas las clases); *Property* (recupera todas las propiedades), o sobre instancias RDF como por ejemplo *Metric* (recupera todas las instancias de la clase Metric). Es importante destacar que esta última consulta devuelve también todas las instancias de las sub-clases de Metric, en virtud de la semántica de RDF, y que está formalmente definida [Bro01b] para el lenguaje de consulta RDF. Esto lo distingue de la mayoría de los lenguajes de consulta RDF, que se han diseñado para la consulta simple de una base de datos de tri-uplas y no tienen una funcionalidad específica o semántica que permita explotar la información implícitamente definida en los esquemas RDF.

6.4 Un lenguaje de Consulta a Nivel Semántico: RQL

En esta sección describiremos brevemente el lenguaje de consulta RQL, que fue utilizado en el sistema de catalogación de métricas e indicadores dado su potencia en la explotación semántica del repositorio. RQL permite recuperar no sólo información de métricas e indicadores y sus relaciones, sino también los metadatos de la ontología y recursos relacionados que estén disponibles en el repositorio local o en otros repositorios de la web. Una descripción detallada del lenguaje, su sintaxis y semántica formal se puede consultar en la documentación en línea disponible en http://athena.ics.forth.gr:9090/RDF/RQL/.

El lenguaje RQL está definido por medio de un conjunto de consultas básicas, e iteradores que se usan para construir otras consultas a través de una composición funcional. Como fue mencionado en la sección anterior, RDF soporta expresiones de caminos generalizadas, permitiendo el uso de variables tanto para denotar nodos rotulados (es decir, clases), como arcos (es decir, propiedades). Esta combinación de esquemas y datos en las expresiones, es una característica clave para satisfacer las necesidades de muchas aplicaciones de la web semántica, como es el caso de nuestro sistema de catalogación.

6.4.1Consultas Básicas

El núcleo del lenguaje de consultas RQL provee esencialmente una forma de acceder a las bases de descripciones RDF con un conocimiento mínimo del esquema empleado. Estas consultas pueden ser usadas para implementar interfaces de navegación semántica sobre bases de datos codificadas en RDF, en una forma muy simple.

Por ejemplo, en el caso de nuestro sistema de catalogación, para cada concepto (es decir, clase), se puede navegar a sus sub-conceptos (es decir, sub-clases) y eventualmente ver todos los recursos que están directamente catalogados dentro de ellos. Para recorrer las jerarquías de clases/propiedades definidas en un esquema RDF, RQL provee funciones tales como *subClassOf* (para subclases en forma transitiva) y *subClassOf^*(para subclases directas). Por ejemplo, la consulta *subClassOf^*(Metric) devuelve una bolsa (*bag*) con los nombres de clase *DirectMetric* e *IndirectMetric*. Existen funciones similares para la jerarquía de propiedades.

También se puede acceder a todas las instancias de una clase, con sólo escribir su nombre. Por ejemplo la consulta *DirectMetric,* devuelve una bolsa que contiene todos los URI´s de las métricas directas del catálogo, ya que estos recursos son la extensión de la clase *DirectMetric.*

RQL soporta un conjunto de operadores comunes que se aplican sobre conjuntos de elementos de un mismo tipo (como son: *union, intersect, minus*). Por ejemplo, la consulta "*Metric* union *Indicator*" devuelve una bolsa con los URI´s de todas las métricas e indicadores.

Para realizar filtros sobre los datos, RQL provee los operadores lógicos estándares como =, < , > , y like (para comparación de cadenas de caracteres). Todos estos operadores se pueden aplicar sobre valores literales (por ejemplo sobre cadenas, enteros, reales, fechas) o recursos (URIs). Además RQL provee un conjunto de funciones de agregación (como son: *min, max, avg, sum* y *count*).

Para concluir esta subsección, queremos señalar que RQL permite recuperar el contenido de cualquier documento con colecciones de datos RDF o información de esquemas RDF. Provee un iterador de la forma *select-from-where*, que se aplica sobre estas colecciones e introduce variables. Debido a que la totalidad de las descripciones o esquemas RDF, se pueden ver como colecciones de nodos y arcos, se pueden usar filtros con expresiones de caminos para recorrer grafos RDF, de una longitud arbitraria.

6.4.2 Consultas sobre Esquemas

En esta subsección describiremos a través de algunos ejemplos sencillos, las consultas RQL sobre esquemas RDF (sin tener en cuenta las instancias), mostrando

cómo se puede usar la notación de expresiones de camino, para recorrer las jerarquías de clases (o propiedades), con el objetivo de implementar navegación o filtrado de información semántica usando las condiciones apropiadas.

Por ejemplo, si se quiere obtener los nombres de todas las clases que son dominio y rango de una propiedad dada, (por ejemplo, la propiedad *Quantifies*), la consulta RQL podría ser la siguiente:

Q1: obtener los nombres de las clases que son dominio y rango de la propiedad Quantifies.

```
select  $C1, $C2
from {$C1}Quantifies{$C2}
```

Esta consulta dará como resultado el siguiente conjunto de duplas:

$C1	$C2
Metric	Attribute
DirectMetric	Attribute
IndirectMetric	Attribute

Tabla 6.1 Ejemplo de resultado de la consulta Q1.

Como se puede observar en la consulta, en la cláusula *from* se usó una expresión de camino compuesta por el nombre de propiedad *Quantifies* (es decir un rótulo de arco), y dos variables de clase, $C1 y $C2 (es decir variables sobre nodos o clases). Debido a que las propiedades RDF se aplican a cualquier sub-clase de su dominio y rango, la expresión *{$C1}Quantifies{$C2}* denota que las variables *$C1* y *$C2* iteran sobre *subClassOf(domain(Quantifies))* y *subClassOf(range(Quantifies))* respectivamente, incluyendo las raíces de la jerarquía.

La cláusula *select* define la proyección sobre las variables de interés (en el ejemplo anterior $C1, $C2). Esta proyección construirá una bolsa de t-uplas ordenadas, cuya aridad depende del número de variables usadas. Consideremos ahora otro ejemplo, supongamos que queremos obtener los nombres de todas las propiedades y sus rangos de la clase *DirectMetric*, la consulta RQL sería en este caso la siguiente:

Q2: obtener los nombres de las propiedades y sus rangos que tienen como dominio la clase *DirectMetric*.

```
select  @P, range(@P)
from {$C1}@P
where $C1 = DirectMetric
```

Esta consulta dará como resultado el siguiente conjunto de duplas:

@P	Range(@P)
Quantifies	Attribute
Includes	Method
Contains	Scale
metricName	Literal
metricValueInterpretation	Literal
metricObjective	Literal
metricReferences	Literal
metricValueType	Literal
metricAccuracy	Literal

Tabla 6.2 Ejemplo de resultado de la consulta Q2.

En la cláusula *from* de la consulta Q2 se usa una expresión de camino compuesta por una variable de clase $C1 y una variable de propiedad @P. En general, las variables de clase se prefijan con el carácter $ y las variables de propiedad con el carácter @.

Para cada posible evaluación *p* de @P, la variable $C1 itera sobre subClassOf(domain(p)). La expresión en la cláusula *where* filtra los valores de @P quedándose sólo con aquellas propiedades para las cuales la clase *DirectMetric* es igual a su dominio, o es una subclase válida de su dominio.

Podemos observar en este último ejemplo que la expresión de camino, atraviesa los arcos (propiedades) *rdfs:domain* y *rdfs:range* en forma conjunta con los arcos *rdfs:subClassOf*, en el grafo RDFS. Otro aspecto importante a destacar es que en el resultado de la consulta, los valores de *range(@P)* son de tipo unión entre el tipo *class* y el tipo *Literal*. Esto se debe a que las propiedades de los datos (en el caso del catálogo de métricas e indicadores) pueden tener como rango clases (cuando ellas representan relaciones, como es el caso de *Quantifies*), o tipos literales (cuando representan atributos, como es el caso de *metricName*).

6.4.3 Consultas sobre Instancias

A continuación ilustraremos algunas consultas RQL sobre descripciones RDF (instancias de clases y propiedades), usando también expresiones de caminos generalizadas. Consideremos por ejemplo la siguiente consulta:

Q3: Encontrar todas las instancias de métricas directas y los atributos que cuantifica

```
Select X,Y
From DirectMetric{X}.Quantifies{Y}
```

En la cláusula *from* se usó una expresión de camino de datos con el nombre de clase *DirectMetric* y el nombre de propiedad *Quantifies*. Las variables de datos X e Y iteran respectivamente sobre la extensión de la clase *DirectMetric* (es decir, recorren los arcos *rdf:type*, conectando los grafos de datos y de esquemas), y los valores destinos de la propiedad *Quantifies* (es decir, recorriendo los arcos en los grafos de datos RDF). El símbolo "." que se usa para concatenar las dos componentes del camino, implican una operación de join (unión de tablas), entre los valores fuentes de la propiedad *Quantifies* y X.

También es posible en RQL usar varios caminos de expresión de longitud arbitraria, en una misma consulta, como se mostrará en el siguiente ejemplo:

Q4: Obtener las métricas directas y sus nombres, que cuantifiquen atributos de la entidad cuyo nombre es "Web_site" .

```
select X,N
from DirectMetric{X}.Quantifies.associeted_with{Y}.entityName{Z},
     {X}.metricName{N}
where Z="Web_site"
```

RQL permite también realizar consultas más complejas, inclusive con caminos de expresión que involucran variables esquemas y datos en forma combinada [Kar02].

En la siguiente sección, describiremos la arquitectura Sesame [Bro01a], un módulo genérico que da soporte al almacenamiento de datos y esquemas RDF y a su recuperación, mediante un lenguaje de consultas basado en RQL (el lenguaje SeRQL).

6.5 Breve Descripción de la Arquitectura Sesame

Sesame es una arquitectura genérica, implementada con software abierto y basada en la web. Permite el almacenamiento persistente de datos y esquemas RDF y su posterior consulta en línea [Dav03]. En la figura 6.4 mostramos una vista general de esta arquitectura.

Para el almacenamiento persistente de datos y esquemas RDF, Sesame fue diseñado para trabajar con un repositorio escalable. Además, para que fuera independiente de los DBMS usados en la administración del repositorio, todo el código

que hace referencia a un DBMS específico, fue concentrado en una única capa de la arquitectura Sesame: la capa RAL (Repository Abstraction Layer).

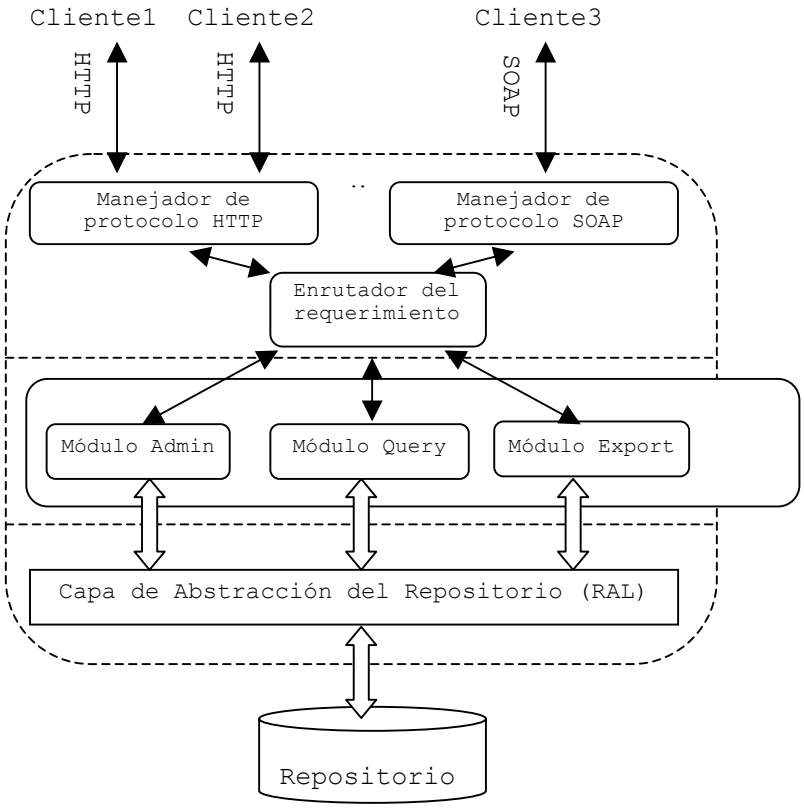

Figura 6.4 Arquitectura Sesame (adaptada de [Dav03])

La capa RAL es una API (Application Programming Interface) que ofrece métodos específicos RDF a sus clientes y los traduce en invocaciones a distintos métodos de los DBMS apropiados. Una ventaja importante de contar con dicha capa separada, es la posibilidad de implementar Sesame arriba de una gran variedad de repositorios en distintos DBMS, sin tener que cambiar ninguna componente Sesame.

Los módulos funcionales de Sesame son clientes de la capa RAL, y actualmente son tres los módulos implementados:

- *El módulo de consultas RQL (módulo Query):* Este módulo evalúa consultas RQL sobre el repositorio persistente.

- *El módulo de administración de RDF (módulo Admin.):* Este módulo permite la actualización incremental de los datos RDF e información de esquemas, como también su eliminación.

- *El módulo de exportación de RDF (módulo Export):* Este módulo permite la exportación de esquemas completos y/o datos desde repositorios con formato RDF serializados XML.

Como sus autores lo indican [Dav03], dependiendo del ambiente donde sea utilizado, se puede necesitar distintas maneras de interactuar con los módulos Sesame. Por ejemplo, para construir una aplicación web sería conveniente acceder a Sesame a través de una comunicación sobre HTTP, pero en otro tipo de aplicaciones podría ser necesario contar con protocolos como RMI (*Remote Method Invocation*) o SOAP (*Simple Object Access Protocol*). Para permitir una mayor flexibilidad y extensibilidad, los manejadores de estos protocolos se han sacado afuera del núcleo Sesame, que implementa las principales funcionalidades. Los manejadores de protocolos conforman una capa intermedia entre los módulos principales de Sesame y sus clientes, cada uno maneja un protocolo específico, traduciendo las solicitudes del cliente a invocaciones a métodos de la API de Sesame adecuadas.

En conclusión, la introducción de la capa de abstracción del repositorio y los distintos manejadores de protocolos, convierten a Sesame en una arquitectura genérica para almacenamiento y consulta de datos y esquemas RDF, más que una implementación particular de un determinado sistema. Esta fue una de las razones por la que se consideró apropiado para la implementación del prototipo del sistema de catalogación de métricas e indicadores.

6.6 Componentes del Prototipo del Sistema de Catalogación de Métricas e Indicadores

El sistema semántico de consultas de nuestro sistema web de catalogación (introducido en la sección 6.2.2) debe permitir la navegación y consulta en línea de la información de métricas e indicadores contenida en el repositorio. Como fuera mencionado en la sección anterior, el ambiente Sesame es un paquete de software abierto que provee módulos de funcionalidades genéricas para el almacenamiento y recuperación de datos y esquemas RDF, que además está basado en un lenguaje de consultas a nivel semántico (SeRQL – Sesame RQL). Por esta razón, consideramos a la arquitectura Sesame una solución atractiva para la implementación del sistema de catalogación, en términos del reuso de software abierto existente, que además provee flexibilidad e interoperabilidad.

La figura 6.5 muestra la arquitectura del sistema semántico de consultas donde se puede ver la interoperación entre los módulos de Sesame y el sistema de consultas.

La arquitectura permite el uso del repositorio en forma transparente por parte de agentes o herramientas de software. Cada aplicación puede interactuar con el repositorio

ya sea a través de los módulos de la API Sesame, o ya invocando los servicios web que brindan funciones más especializadas de búsqueda y consultas.

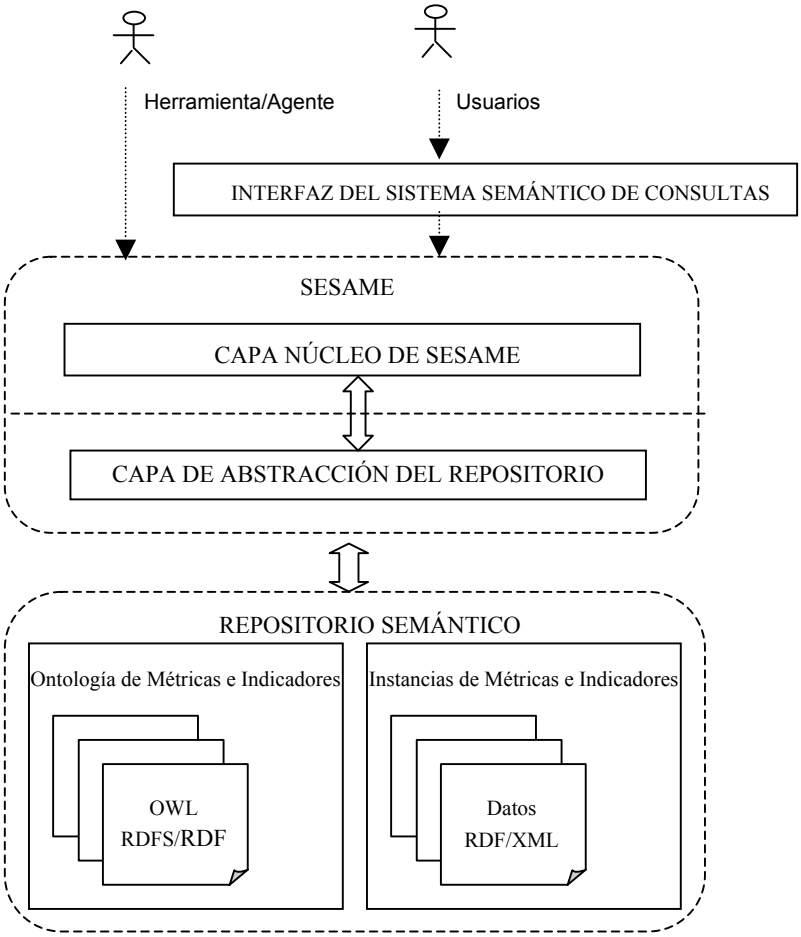

Figura 6.5 Arquitectura del Sistema Semántico de Consultas

Por otra parte, el repositorio contiene la totalidad del conocimiento relacionado a métricas e indicadores que se puede manipular y referir en forma integral. Dicho repositorio contiene tanto la meta información de la ontología como las instancias de datos.

La búsqueda y consulta de instancias y esquemas de la ontología son implementadas por el sistema Sesame, soportando para tal fin el lenguaje SeRQL, una versión ligeramente ampliada de RQL, que conserva sus principales características (vistas en la sección 6.4).

6.7 Diseño Navegacional del Sistema de Catalogación

El proceso de modelado conceptual es fundamental, si se quiere desarrollar aplicaciones web en forma correcta y con el soporte de la documentación necesaria. Particularmente, consideramos que el modelado navegacional es un aspecto crítico e importante a la hora de diseñar aplicaciones web. En esta sección se analiza el modelo navegacional usando el enfoque OOWS (Object-Oriented Web Solutions) [Pas01], que es un método para especificar requerimientos navegacionales (entre otros) de una aplicación web a nivel conceptual. Este enfoque usa estereotipos UML para representar sus primitivas de abstracción.

Las próximas sub-secciones presentan el modelo OOWS aplicado a la definición de los requerimientos navegacionales del sistema de catalogación de métricas e indicadores para el usuario registrado.

6.7.1 Modelo Navegacional para el Usuario Registrado

OOWS es un método de modelado conceptual que usa un *Modelo Navegacional* para capturar de una forma conveniente los requerimientos de navegación de una aplicación web. Provee un conjunto de constructores conceptuales para definir las capacidades de navegación (estructura, navegabilidad, etc.) de un sistema web. El modelo navegacional se compone de un conjunto de mapas navegacionales, uno para cada tipo de usuario del sistema.

Un *Mapa Navegacional* (ver figura 6.6) representa la vista personalizada que tiene cada clase de usuario sobre el sistema y define cómo estos usuarios pueden acceder a la información del sistema y su funcionalidad. Este mapa es un grafo dirigido donde los nodos representan unidades de interacción del usuario (contextos navegacionales), y los arcos (enlaces navegacionales) representan la alcanzabilidad del nodo (que definen caminos navegacionales válidos). De esta manera, un *Contexto Navegacional* es un nodo en el grafo (representados gráficamente como paquetes UML estereotipados con la palabra reservada <<context>>), y representa una vista sobre un conjunto de atributos de clases y servicios, y también relaciones entre clases, extraídas del modelo de objetos (ver modelo UML en la figura 5.1).

Existen dos tipos de contextos: a) *Contextos de Exploración* (rotulados con una "E"), que representan nodos navegacionales que pueden ser alcanzados en cualquier momento, independientemente del contexto actual; y b) *Contextos de Secuencia* (rotulados con una "S"), los cuales sólo pueden ser alcanzados siguiendo un camino navegacional predefinido (secuencia de contextos navegacionales).

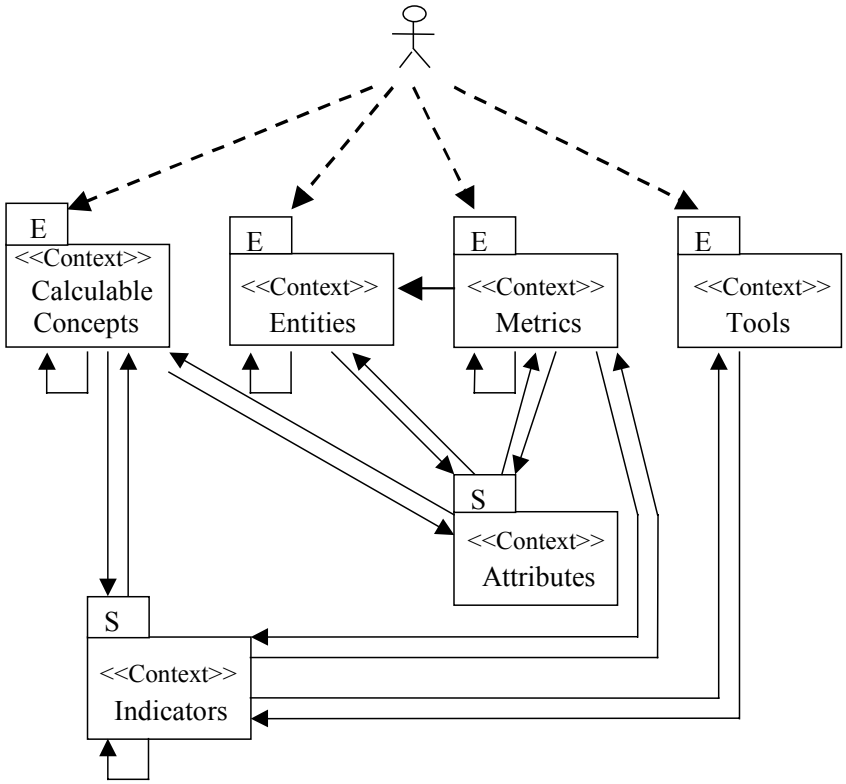

Figura 6.6 Mapa Navegacional para el Usuario Registrado del Sistema de Catalogación

La figura 6.6 muestra el mapa navegacional para el usuario registrado del sistema de catalogación de métricas e indicadores. Se definen seis contextos navegacionales: *Calculable Concepts*, *Entities*, *Metrics* y *Tools* (estos son contextos de exploración, a los cuales llega una flecha punteada directamente desde el usuario), y por otro lado *Indicators* y *Attributes* (estos son contextos de secuencia, sólo alcanzables siguiendo un camino navegacional predefinido). Por ejemplo, el contexto *Indicators* no se puede alcanzar desde el contexto *Entities*. Cada uno de estos contextos definen una vista sobre el modelo de objetos, proveyendo acceso a la información relevante (atributos de clases), y funcionalidad (métodos de clases) para el usuario registrado.

El paso siguiente es definir en detalle cada contexto especificado en el mapa navegacional. Un contexto navegacional se compone de *Clases Navegacionales* y *Relaciones Navegacionales*. Una clase navegacional representa una vista de una clase en el modelo de objetos, que define un subconjunto de atributos y servicios visibles. Las clases navegacionales son representadas gráficamente como clases UML estereotipadas con la palabra reservada <<view>> (ver figura 6.7). Además, una clase navegacional representa una consulta que resulta en la recuperación de todas las instancias de dicha clase, especificando los atributos recuperados. Se puede especificar un filtro respecto a

una clase navegacional estableciendo una condición a cumplir para las instancias a ser recuperadas.

A modo de ejemplo, en la figura 6.7 mostramos la definición del contexto *Metrics*. Este contexto es responsable de proveer la información relevante sobre las instancias de métricas para el usuario registrado. El contexto *Metrics* tiene catorce clases navegacionales (por ejemplo *Metric*, *Entity*, *Attribute*, entre otras), con sus atributos (*name*, *objective, references*, etc).

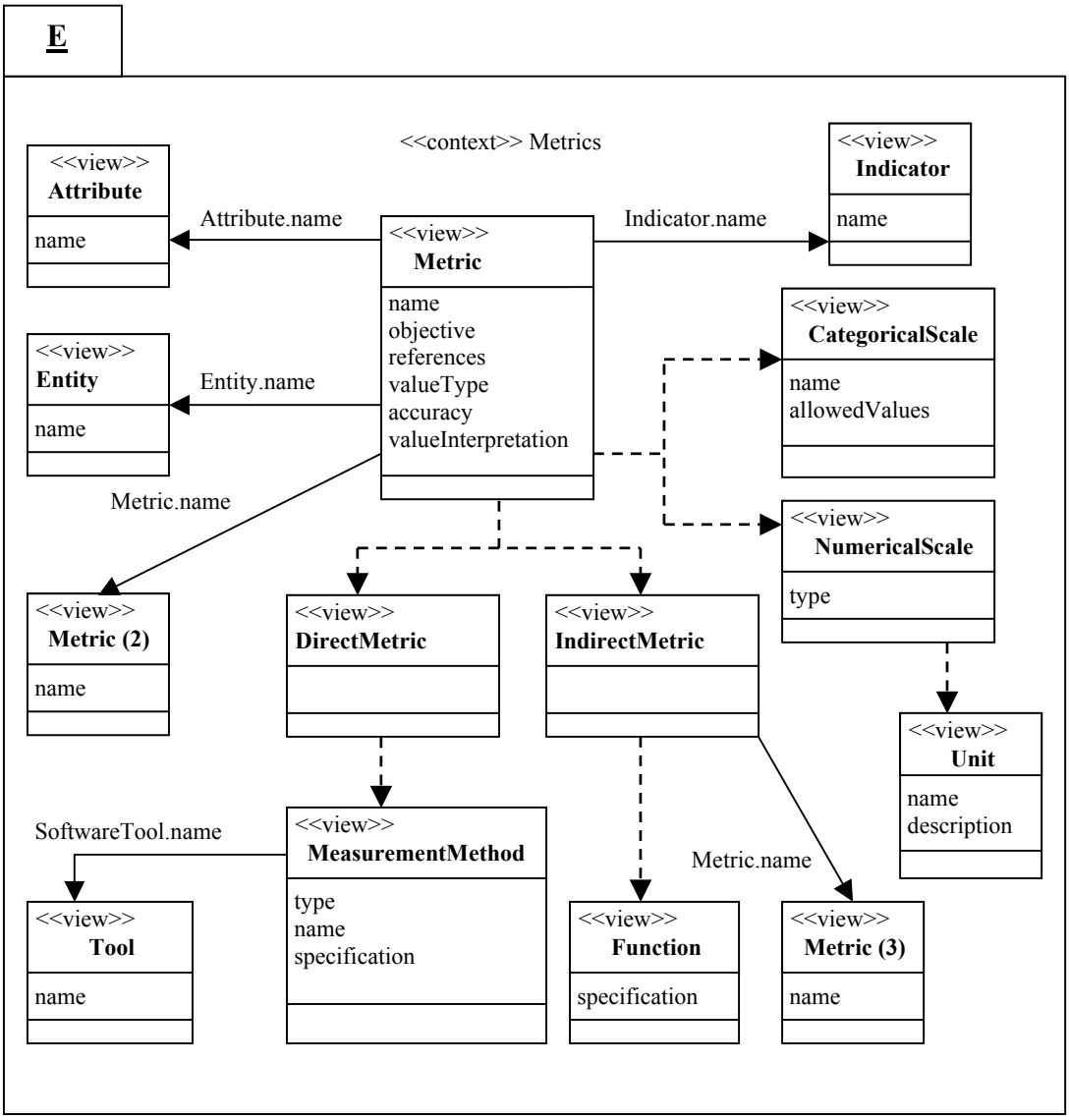

Figura 6.7 Contexto Navegacional *Metrics* para el Usuario Registrado

Todas las clases navegacionales se deben relacionar por medio de relaciones binarias unidireccionales, llamadas *Relaciones Navegacionales* (dibujada con una flecha). Una *Relación Navegacional* se define sobre relaciones existentes de agregación, asociación, composición o especialización en el modelo de objetos y especifica la

recuperación de todas las instancias sobre la clase navegacional destino, relacionada a la clase original. Por ejemplo, en el caso del contexto *Metrics* de la figura 6.7*,* para cada métrica consultada, se mostrarán en el mismo contexto, las instancias de *Attributes* (name), *Entities* (name), *Indicators* (name), etc.

Existen dos tipos de relaciones navegacionales: 1) *Relaciones de dependencia de contexto* (graficadas con flechas punteadas), y 2) *Relaciones de contexto* (graficadas con flechas llenas). Las primeras sólo definen un conjunto de recuperación. Las últimas en cambio, definen una navegación hacia un *[contexto destino],* usando como vínculo un atributo de la clase navegacional destino. Por ejemplo, luego de recuperar todos los nombres de entidades (*Entities*) relacionadas a una métrica dada, es posible navegar al contexto destino *[Entities],* simplemente mediante la selección de una entidad, y usando el atributo *name* de esa entidad como vínculo, (de esta forma se define el enlace navegacional entre contextos *Metrics -> Entities* especificado en el mapa navegacional – ver figura 6.6). Los restantes contextos navegacionales del sistema de catalogación, se definieron de manera similar.

6.8 Implementación de la Navegación Semántica

En esta sección y en la siguiente se muestran aspectos de navegación y búsqueda semántica, que fueron implementadas por Molina H. y Papa F. (integrantes becarios del grupo GIDIS), en un prototipo del sistema de catalogación, el cual fue probado para un conjunto de instancias de métricas e indicadores [Mol04] .

Como ya fue comentado en el capítulo 5, para implementar la información ontológica de métricas e indicadores, se usaron los lenguajes estándares RDF y RDFS. Para explotar dicha información del repositorio, se implementaron en el prototipo dos tipos de navegación semántica: 1) La simple navegación semántica a través de la taxonomía de conceptos de la ontología y, 2) Una navegación mas “amigable” al usuario humano, que si bien está basada en la semántica de la ontología, fue diseñada teniendo en cuenta otros aspectos de calidad como son usabilidad y navegabilidad.

6.8.1 Implementación de la Navegación Semántica usando la Taxonomía de Conceptos

Para implementar la navegación semántica sobre los conceptos de la ontología de métricas e indicadores, se usaron los módulos de la API del servidor Sesame, que ejecuta consultas SeRQL, sobre repositorios RDF/RDFS cargados en él.

Las consultas recorren las propiedades *subClassOf* para obtener todos los elementos de la ontología en forma jerárquica. La figura 6.8 muestra la pantalla desplegada, cuando se usa esta opción de navegación semántica.

6.8.2 Implementación de la Navegación Semántica para el Usuario Registrado

Para implementar el segundo tipo de navegación semántica para el usuario registrado, se tuvo en cuenta el diseño navegacional presentado en la sección 6.7. Además se debieron formalizar en RDFS, algunas primitivas del modelo navegacional OOWS, para automatizar la navegación a partir del mapa navegacional y de la semántica proporcionada por la ontología del repositorio.

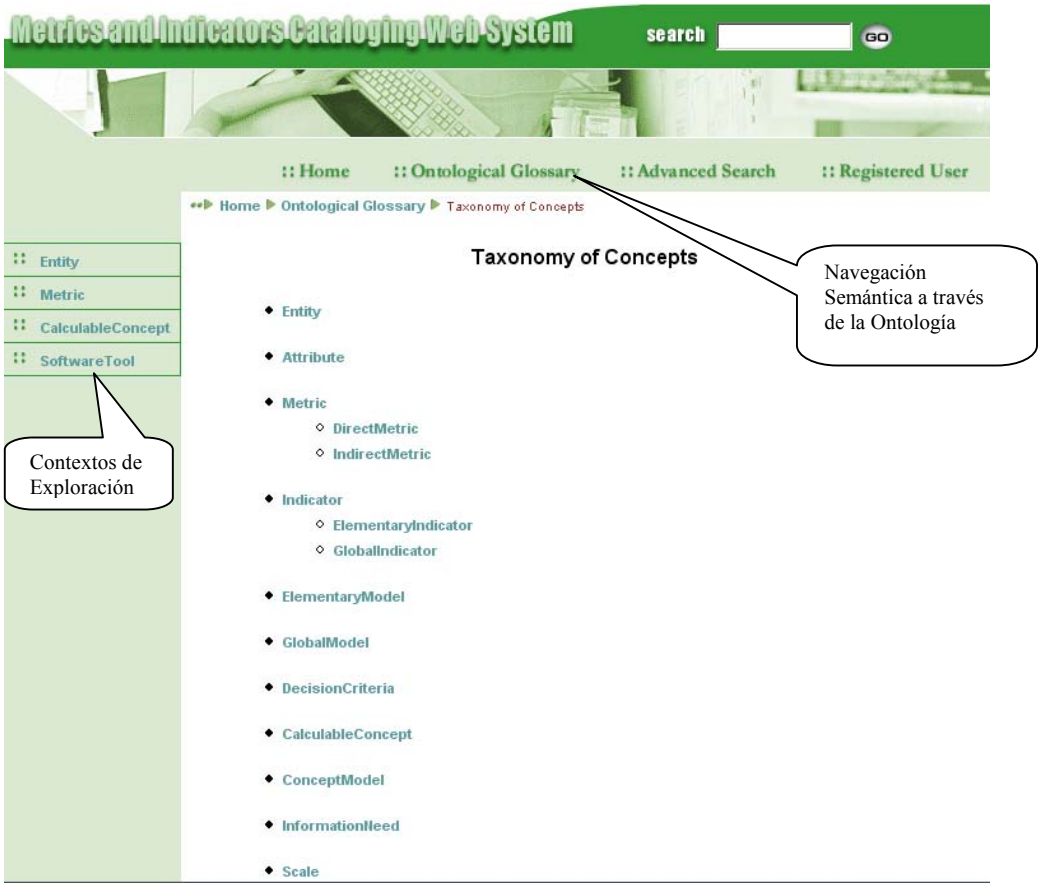

Figura 6.8 Navegación Semántica a través de la Taxonomía de Conceptos

Las primitivas RDFS definidas (ver figura 6.9), permiten contar con un lenguaje de marcado, que hace posible guardar la información del modelo navegacional de una manera procesable por la computadora.

En un documento de esquemas RDF se definieron los recursos *View*, *ExplorationContext*, *NavigationalLink* y *ContextDependencyRelationship.* En el código RDFS de la figura 6.9 se muestra la especificación de dichos recursos. Los nodos del mapa navegacional fueron implementados como clases RDFS (como el caso de *View* y

ExplorationContext) y los arcos como propiedades RDFS (como el caso de *NavigationalLink* y *ContextDependencyRelationship*).

Es importante hacer notar que no fue necesario implementar la primitiva *Relación de contexto* (*ContextRelationship*), debido a que este tipo de relaciones ya están explícitamente especificadas en la ontología.

```
<rdfs:Class rdf:ID="View">
   <rdfs:label xml:lang="en">View</rdfs:label>
   <rdfs:comment>Entry point for a navigational map</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="ExplorationContext">
   <rdfs:label xml:lang="en">Exploration Context</rdfs:label>
   <rdfs:comment>Destination to an Entry point</rdfs:comment>
</rdfs:Class>

<rdfs:Property rdf:ID="NavigationalLink">
   <rdfs:label xml:lang="en">Navigational Link</rdfs:label>
   <rdfs:comment>Navigational Link from a given user View to an
                 Exploration Context</rdfs:comment>
   <rdfs:domain rdf:resource="#View"/>
   <rdfs:range rdf:resource="#ExplorationContext"/>
</rdfs:Property>

<rdfs:Property rdf:ID="ContextDependencyRelationship">
    <rdfs:label xml:lang="en">Context Dependency
                 Relationship</rdfs:label>
    <rdfs:comment>It allows to show the information in the same
                 view</rdfs:comment>
</rdfs:Property>
```

Figura 6.9 Ejemplo de definición de primitivas OOWS en RDFS

Nuestro modelo de navegación semántica, se basa fundamentalmente en la semántica provista por la ontología, con el agregado de alguna información del mapa navegacional para enriquecer la navegación entre distintos contextos. La línea de investigación con el fin de realizar una ontología de navegación para los conceptos que involucra OOWS, no fue el objetivo de esta tesis. Un trabajo inicial en esta dirección fue presentado recientemente en IWWOST'04 [Tor04] .

En el trozo de código RDF de la figura 6.10, se muestra a modo de ejemplo, cómo quedan instanciados algunos elementos del modelo navegacional para el caso del contexto navegacional *Metrics.* En dicha figura, se muestran sólo tres descripciones de recursos RDF.

```
<rdf:RDF ... xmlns:navmod="http://gidis/rdf/navmod.rdf#">
      ...
<rdf:Description
            rdf:about="http://gidis/rdf/met_ind.rdf#Metric">
  <rdf:type
     rdf:resource="http://gidis/rdf/navmod.rdf#ExplorationContext"/>
</rdf:Description>

<rdf:Description rdf:ID="RegisteredUserView">
    <rdf:type rdf:resource="http://gidis/rdf/navmod.rdf#View"/>
    <navmod:NavigationalLink
            rdf:resource="http://gidis/rdf/met_ind.rdf#Metric"/>
            ...
</rdf:Description>
...
<rdf:RDF>

<rdf:Description
      rdf:about="http://gidis/rdf/met_ind.rdf#indicatorIncludes">
    <rdf:type
rdf:resource="http://gidis/rdf/navmod.rdf#ContextDependencyRelationshi
p"/>
</rdf:Description>
```

Figura 6.10 Ejemplo de Instancias del Esquema de Mapa Navegacional para el Usuario Registrado

En la primera descripción (*rdf:Description*), se asocia un contexto de exploración a la clase Metric, similares descripciones se codificaron para cada contexto de exploración.

En la segunda descripción se define la vista *RegistredUserView*, y se asocia un enlace navegacional entre dicha vista y el contexto navegacional *Metrics.* Enlaces similares se definieron para cada contexto navegacional accesible para el perfil *RegistredUserView.*

Finalmente, se debieron definir todas las relaciones de dependencia de contexto, estas son relaciones que no están explícitamente definidas en la ontología y que son necesarias para poder navegar de un contexto navegacional a otro, como es el caso de la relación *IndicatorIncludes,* que permite navegar desde el contexto navegacional *Metrics* al contexto navegacional *Indicators* (ver figura 6.7).

Para poder implementar los caminos de navegación dentro de los contextos navegacionales se utilizaron las relaciones de la ontología implementadas en el esquema RDF. De esta manera, toda capacidad de navegación del sistema de catalogación, está dirigida por la información semántica almacenada en el repositorio.

6.9 Implementación de la Búsqueda Semántica

Para el sistema de catalogación se implementaron dos tipos de búsqueda semántica: 1) Una búsqueda rápida, disponible desde cualquier página del sistema, que se basa en la búsqueda semántica de palabras claves, y 2) una búsqueda avanzada accedida desde el menú principal, que si bien se basa en la búsqueda semántica de palabras claves, también permite personalizar la búsqueda especificando dónde buscar (sólo en el nombre del recurso, en la descripción completa o en ambos) y qué tipos de recursos recuperar.

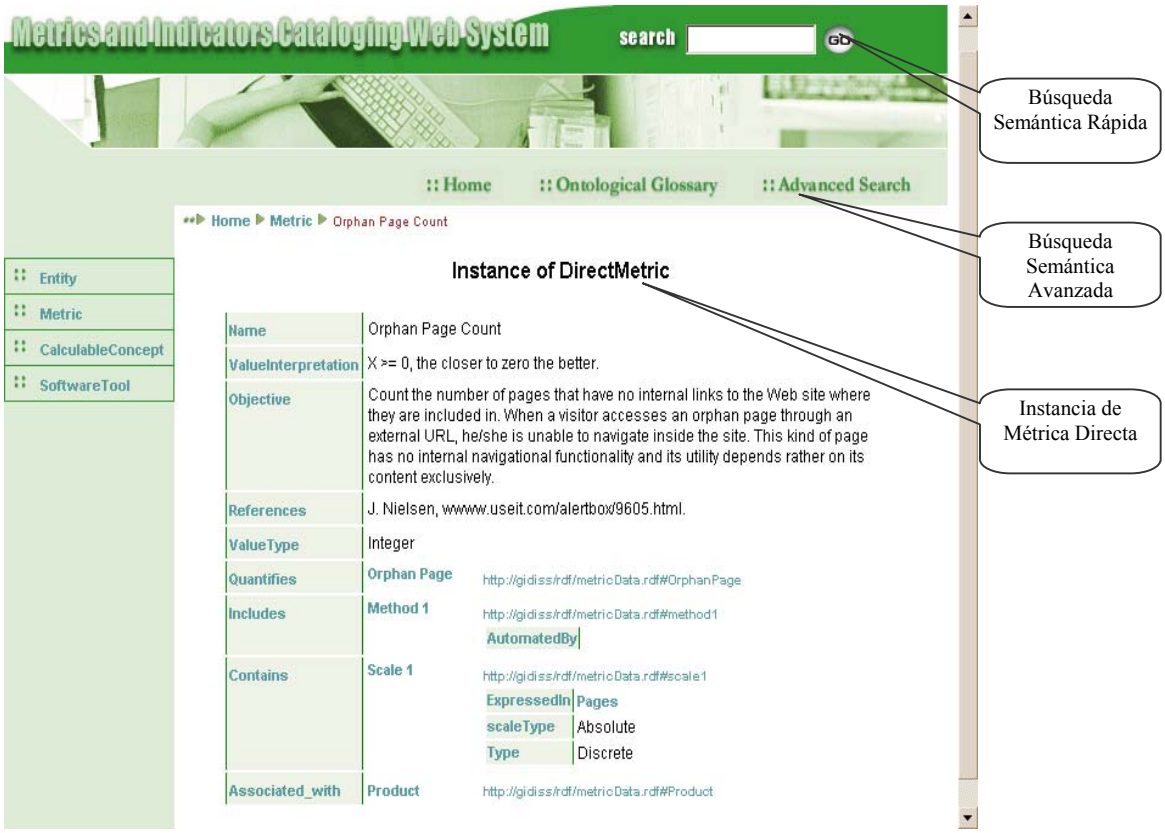

Figura 6.11 Exploración y Búsqueda del SCMI por parte del Usuario Registrado

Para ambos tipos de consulta, se usan invocaciones al motor de consulta SeRQL a través de la API del Servidor de Sesame. En la figura 6.11 se muestran los botones de acceso a sendos tipos de búsqueda, y la exploración de los datos específicos de una instancia de métrica directa.

Para el caso de la búsqueda rápida, se aceptan cadenas de caracteres conteniendo palabras claves separadas por espacios y opcionalmente encerradas por comillas dobles. Dichas palabras se buscan en las propiedades *rfds:label* y *rdfs:comment* , de las distintas definiciones RDF del catálogo. La propiedad *rfds:label* contiene el nombre del recurso y *rdfs:comment* contiene la descripción del mismo. Por ejemplo, la figura 6.12 muestra la consulta SeRQL generada si se ingresa la cadena de búsqueda: Enlace "Sitio Web". El resultado devuelto es un conjunto de URI´s correspondiente a recursos que tienen la palabra "Enlace" o "Sitio Web" dentro de su descripción.

```
SELECT Item, Name, Description
FROM {Item} <rdfs:label> {Name}; <rdfs:comment> {Description}
WHERE (Description like "*Enlace*" OR Name like "*Enlace*") AND
      (Description like "*Sitio Web*" OR Name like "*Sitio Web*")
USING NAMESPACE
       metr = <!http://gidis/rdf/metind.rdf#>
```

Figura 6.12 Ejemplo de consulta SeRQL para Implementar una Búsqueda Rápida

Para la implementación de la búsqueda avanzada se diseñó una interfaz que solicita los siguientes elementos de entrada:

- Palabra o palabras a buscar (¿Qué buscar?)
- Dónde buscar (puede ser en el nombre del recurso, descripción o todas las propiedades)
- Qué elementos recuperar (de las clases definidas en el modelo conceptual)

Esta consulta se procesa de forma análoga a la búsqueda simple pero permite una búsqueda más refinada. Por ejemplo, la figura 6.13 muestra la consulta SeRQL generada si se desea buscar: todas las Métricas y Atributos que contengan la palabra "Enlace" ya sea en su nombre o su descripción:

```
SELECT Item, Name, Description
FROM {Item} <rdf:type> {Class};
            <rdfs:label> {Name};
            <rdfs:comment> {Description}
WHERE (Name like "*Enlace*"
       OR Description like "*Enlace*") AND
      (Class = <metr:Metric> OR Class = <metr:Attribute>)
USING NAMESPACE
    metr = <!http://gidis/rdf/metind.rdf#>
```

Figura 6.13 Ejemplo de consulta SeRQL para Implementar una Búsqueda Avanzada

6.10 Conclusiones

En este capítulo se presentaron el diseño arquitectural y aspectos de implementación del sistema de catalogación de métricas e indicadores, con capacidades de Web Semántica. Teniendo en cuenta las tecnologías y lenguajes disponibles para la web semántica, y luego de analizar la importancia de usar un lenguaje de consulta que opere a nivel semántico para la explotación de catálogo, se eligió la arquitectura Sesame y el lenguaje de consultas SeRQL (una extensión de RQL), para la implementación de un prototipo de dicho sistema.

Como fue analizado en este capítulo, RQL es un lenguaje de consultas declarativo tipado, que permite construir consultas potentes en forma sencilla, debido a su propiedad de analizar estructuras semánticas, es decir, tiene en cuenta no sólo la información explícitamente almacenada en repositorios RDF/RDFS sino también la semántica implícita contenida en los esquemas (metadatos).

Además, su capacidad para hacer corresponder variables tanto a nodos (clases), como a arcos (propiedades) de la información RDF/RDFS, facilita la explotación de los documentos semánticos del repositorio, permitiendo no sólo recuperar información de métricas e indicadores y sus relaciones, sino también la información descriptiva (metadatos) de los recursos del catálogo.

A través de la implementación del prototipo, se puedo comprobar las ventajas de la búsqueda y navegación semántica. Con los actuales algoritmos de los buscadores de información web, al ingresar una palabra clave (o varias), con más o menos éxito se recupera información que puede ser o no de interés para el usuario. Esto es debido a que los agentes actuales de búsqueda no se diseñan para “comprender” (procesar) la información que reside en la web, precisamente porque es prácticamente imposible

conocer la semántica de los datos ubicados en las diferentes páginas. Con la búsqueda y navegación semántica, en cambio, el acceso a la información deseada es mas eficaz.

Otro beneficio importante de la capa semántica, es que la información bien definida semánticamente puede ser entendida no sólo por seres humanos, sino también por agentes, motores de búsqueda y herramientas automáticas y de esta manera puede ser aprovechada en forma más eficiente para facilitar las tareas de aseguramiento de calidad.

Por ejemplo, se podría programar un agente para que recorra la web (semántica) buscando información de métricas (y sus métodos de medición) que midan atributos de calidad de páginas web. Dicho agente podría analizar los contenidos de cada sitio ejecutando las consultas RQL apropiadas sobre los documentos RDF/RDFS y luego aplicar la información obtenida, invocando métodos de medición automáticos.

Capítulo 7- Conclusiones y Trabajos Futuros

7.1 Conclusiones

En esta tesis se presentó el diseño arquitectural y aspectos de implementación del sistema de catalogación de métricas e indicadores, con capacidades de Web Semántica, basado en la ontología de métricas e indicadores, también especificada en este trabajo.

Para el desarrollo de dicho sistema se debieron integrar conocimientos y tecnologías de distintas áreas de investigación, como son el área de Ingeniería de Software, el área de Ontologías y el área de Web Semántica, entre otras. Como consecuencia, los aportes realizados son interdisciplinarios.

Para una mejor síntesis de las conclusiones de esta tesis, a continuación se analizan los resultados obtenidos teniendo en cuenta las principales contribuciones realizadas en las distintas áreas.

7.1.1 Sistema de Catalogación de Métricas e Indicadores

La Ingeniería de Software, al igual que otras ingenierías, necesita de información de métricas e indicadores para poder especificar, predecir, evaluar y analizar distintos atributos y características de los entes (productos, procesos, etc.) que participan en el desarrollo y mantenimiento del software.

Observando el volumen cada vez mayor y heterogéneo de dicha información, y teniendo en cuenta su importancia como base fundamental para las actividades de medición y evaluación en el área de Ingeniería de Software, se puede ver la importancia de disponer de repositorios genéricos y soporte de catalogación que faciliten de un modo extensible, el almacenamiento, consulta, explotación y reuso de toda la información relacionada a métricas e indicadores.

El Sistema de Catalogación de Métricas e Indicadores (SCMI) que se propuso en esta tesis (capítulo 6), da soporte a la necesidad planteada, posibilitando que la información obtenida de distintas fuentes y relacionada a distintos ámbitos de aplicación pueda ser compartida y reusada, en pos de mediciones más objetivas, precisas, reproducibles y significativas.

Este sistema de catalogación fue diseñado como una aplicación web, debido a la importancia que tiene hoy Internet, tanto para compartir el conocimiento como para su explotación.

Además, al estar implementado con tecnologías de web semántica, la información del catálogo tiene un significado bien definido formalmente, facilitando su procesamiento automático, y permitiendo, de esta manera, que los datos y meta-datos de métricas e indicadores puedan ser explotados tanto por agentes automáticos como por herramientas de medición y/o evaluación.

7.1.2 Ontología de Métricas e Indicadores

Para que la información de métricas e indicadores pueda ser compartida, reusada, interpretada y aplicada uniformemente a distintos ámbitos, es necesario contar con una terminología común, y una estructura de metadatos (datos acerca de cómo están estructurados los datos), que permita tanto el procesamiento automatizado de la información, como su correcta interpretación.

En el capítulo 5 se especificó la ontología de métricas e indicadores, y se implementó con lenguajes recomendados para la web semántica, de esta manera, provee los metadatos necesarios para una representación declarativa del conocimiento de métricas e indicadores. Además, la ontología proporciona una definición formal de conceptos consensuados que asegura la interpretación correcta del conocimiento compartido y, por último, brinda un vocabulario común, bien definido para el intercambio de información.

La ontología de métricas e indicadores se basó principalmente en los términos definidos por los estándares de la ISO, y en un arduo trabajo de discusión entre investigadores y profesionales del área de Ingeniería de Software, de algunos grupos de investigación participantes. Acordar la semántica de todos los conceptos definidos en la ontología no fue una tarea fácil, es más, algunos conceptos debieron ser definidos sin el consenso total.

Es importante destacar la importancia que tienen las ontologías en los sistemas basados en conocimiento, como es el caso del Sistema de Catalogación de Métricas e Indicadores, ya que proveen una comprensión común y compartida de un dominio que puede ser comunicado entre personas y computadoras. Además, se han vuelto una tecnología clave para proveer entendimiento automático del conocimiento. Éstas, cuando están implementadas con lenguajes formales, interrelacionan el entendimiento humano de los términos, con la procesabilidad automática por parte de las computadoras.

7.1.3 Búsqueda y Navegación Semántica para el SCMI

En las aplicaciones web tradicionales, la navegación y exploración de la información está dirigida por la interacción del usuario, quien muchas veces debe intentar caminos alternativos de navegación hasta encontrar la información que busca o determinar que no existe. Si bien la mayoría de las aplicaciones web, estructuran la información en categorías para su navegación, estas categorías no son taxonomías consensuadas, y raramente coincide con el punto de vista del usuario, quien debe encontrar la información que busca mediante el mecanismo de "prueba y error". La navegación semántica a través de la taxonomía de conceptos, que se propuso en este trabajo, por el contrario, facilita la exploración de la información por parte del usuario, al estar organizada en estructuras taxonómicas basadas en una ontología.

Con respecto a la búsqueda de información, el problema es similar, los motores de búsqueda que existen actualmente en la web, buscan documentos por la coincidencia de palabras claves ingresadas por el usuario, en forma exacta o aproximada. El inconveniente en este caso, es que los resultados obtenidos se refieren a tópicos relacionados a los diversos significados de las palabras claves ingresadas, resultando en la recuperación de demasiados documentos, no todos relacionados al tema de interés.

La solución a este problema es la búsqueda semántica, es decir, buscar documentos que contengan las palabras claves ingresadas con su misma semántica. Las ontologías representan la tecnología adecuada para facilitar la búsqueda semántica y proveer "entendimiento" automático de la información, facilitando la explotación del conocimiento contenido en documentos y repositorios de una forma más eficiente.

En el capítulo 6, sección 6.3.2 se analizaron los lenguajes de consultas apropiados para la búsqueda y navegación semántica, y se mostró sus ventajas a través de la implementación del prototipo.

Otro beneficio importante de nuestro sistema de catalogación con potencia de web semántica, es que la información bien definida semánticamente puede ser entendida no sólo por seres humanos, sino también por agentes, motores de búsqueda y herramientas automáticas y, de esta manera, puede ser aprovechada en forma más eficiente para automatizar las tareas de aseguramiento de calidad.

7.2 Trabajos Futuros

Una línea de avance sobre la implementación de la ontología es la construcción de una nueva versión de la misma, codificada en OWL (nueva recomendación del W3C para especificar ontologías en la web semántica). La implementación de la ontología de métricas e indicadores en OWL permitirá la especificación formal de los axiomas y

restricciones definidos en la etapa de conceptualización, y el procesamiento automático de dichas restricciones.

Por otra parte, para tener una real eficacia en el procesamiento automático de la semántica de los datos, es importante proveer a la ontología de reglas de inferencia especificadas formalmente, de manera que agentes automáticos puedan deducir nuevo conocimiento a partir de la ontología y las reglas consistentemente especificadas. De ahí la importancia de adicionar una capa lógica por encima de la ontológica, para proveer al Sistema de Catalogación de capacidad de razonamiento sobre sus datos y metadatos. Esta es también una futura línea de investigación.

Finalmente, se está avanzando, en nuestro grupo de investigación y desarrollo (GIDIS), en el diseño e implementación de diferentes herramientas integradas para dar soporte a los procesos de aseguramiento de calidad, siendo el sistema de catalogación de métricas e indicadores la base de conocimiento sobre la que se alimentan dichas herramientas.

General Pico, Septiembre de 2004

Referencias

[Alb83] Albrecht A.J. and Gaffney J.E., 1983, “Software function, source lines of code and development effort prediction: a software science validation”, *IEEE Transactions*, 6(1983), pp. 639-648.

[Ale01] Alexaki, S.; Christophides, V.; Karvounarakis; Plexousakis, D. and Tolle K. ,2001. “The RDFSuite: Managing Voluminous RDF Description Bases”. Technical report, Institute of Computer Science, FORTH, Heraklion, Greece.

[Ang81] Angeles P. J., 1981. “Dictionary of Philosophy”. Harper Reference, New York.

[Bac02] Baclawski, K.; Kokar, M.; Kogut, P.; Hart, L.; Smith, J.; Holmes, W.; Letkowski, J.; Aronson, M. and Emery, P.; 2002. "Extending the UML for Ontology Development", SOSYM 2002, Sofware System Model (2002) 1: 1-15, Springer-Verlag.

[Bat95] Bateman, J. A.; Henschel R. and Rinaldi F., 1995. “Generalized Upper Model 2.0: documentation”,Technical Report, GMD/IPSI, Darmstadt, Germany.

[Bec02] Beck, H. and Pinto, S.; 2002. “Overview of Approach, Methodologies, Standards, and Tools for Ontologies”. Disponible en http://it.ifas.ufl.edu/AOS/BackgroundPaper/.

[Bech04] Bechhofer S., van Harmelen F., Hendler J., Horrocks I., McGuinness D., Patel-Schneider P. and Stein L., 2004. “OWL Web Ontology Language Reference”, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-owl-ref-20040210/.

[Ben96] Benjamins, V. R. and Gomez-Perez A., 1996. “Knowledge-System Technology: Ontologies and Problem-Solving Methods”. Disponible en http://www.swi.psy.uva.nl/usr/richard/pdf/kais.pdf.

[Ber01] Berners-Lee, T.; Hendler, J.; Lassila, O., 2001, “The Semantic Web”. Scientific American, http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html.

[Bla98] Blazquez, M.; Fernandez, M.; Garcia-Pinar, J.; Gomez-Perez, A. 1998. “Building Ontologies at the Knowledge Level using the Ontology Design Environment”. Knowledge Acquisition Workshop (KAW’98).

[Bor97] Borst, W. N., 1997. “Construction of Engineering Ontologies”. PhD thesis, University of Twente, Enschede, 1997.

[Bra98] Bray T.; Paoli J. and Sperberg-Mcqueen C., 1998. "Extensible Markup Language (XML) 1.0.". W3C (World-Wide Web Consortium) Recommendation 10 February 1998. http://www.w3.org/TR/REC-xml .

[Bri00] Brickley, D. and Guha, R., 2000. "Resource Description Framework (RDF) Schema Specification 1.0.", Technical report, W3C, 27 March 2000. http://www.w3.org/TR/2000/CR-rdf-schema-20000327.

[Bri02] Briand, L., Morasca, S. and Basili, V., 2002, "An Operational Process for Goal-driven Definition of Measures". *IEEE Transactions on Software Engineering* 28(12), pp. 1106-1125.

[Bro01a] Broekstra J. and Kampman A.; 2001. "Sesame: A generic Architecture for Storing and Querying RDF and RDF Schema". Deliverable 9, On-To-Knowledge project. http://www.ontoknowledge.org/downl/del10.pdf.

[Bro01b] Broekstra, J. and Kampman, A.; 2001. "Query Language Defnition. On-To-Knowledge (IST-1999-10132) Deliverable 9, Aidministrator Nederland b.v. See http://www.ontoknowledge.org/.

[Cap91] Capers, J. , "Applied Software Measurement", McGraw-Hill Inc., New York, 1991.

[Car01] Carlson, D.; 2001. "Modeling XML Applications with UML", Addison-Wesley.

[Cha01] Chamberlin, D.; Florescu, D.; Robie, J.; Simeon, J. and Stefanescu, M., 2001. "XQuery: A Query Language for XML". Working draft, World Wide Web Consortium. See http://www.w3.org/TR/xquery/.

[Cha98] Chang, W.; 1998. "A Discussion of the Relationship Between RDF-Schema and UML", http://www.w3.org/TR/1998/NOTE-rdf-uml-19980804/.

[Cha99] Chandrasekaran, B. 1999. "What Are Ontologies, and Why Do We Need Them?. *IEEE Inteligent Systems*, Janeiro 7 Fevereiro 99, EUA.

[Charis] Charismatek Software Metrics, http://www.charismatek.com.au.

[Con86] Conte S.D., Shen V.Y. and Dunsmore H.E., 1986, "Software Engineering Metrics and Models".

[Cor01] Corcho, O.; Fernández-López, M. and Gomez-Perez A. 2001. "Technical Roadmap v 1.0".Deliverable 1.1.1 del Consorsio OntoWeb , disponible en: http://ontoweb.aifb. uni-karlsruhe.de/About/Deliverables/deliverable_view.

[Cra99] Cranefield, S. and Purvis, M. 1999. "UML as an ontology modelling language". In Proceedings of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI-99).

[Cra01a] Cranefield, S. 2001. "Networked knowledge representation and exchange using UML and RDF". *Journal of Digital Information*, 1(8), . http://jodi.ecs.soton.ac.uk/.

[Cra01b] Cranefield, S. (2001) "UML and the Semantic Web". Discussion Paper 2001/04, Department of Information Science, University of Otago, New Zealand http://www.otago.ac.nz/informationscience/publctns/complete/papers/dp2001-04.pdf.gz

[Cur02] Curbera, F.; Duftler, M.; Khalaf, R.; Nagy, W.; Mukhi, N. and Weerawarana, S.; 2002. "Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI". *IEEE Internet Computing* V.6, N 2, pp. 86-93.

[Cyc02] OpenCyc Selected Vocabulary and Upper Ontology site, http://www.cyc.com/cycdoc/vocab/vocab-toc.html.

[DamlOil] daml+oil directory site http://www.daml.org/2001/03/daml+oil-index.html

[Dav03] Davies, J.; Fensel, D. and Van Harmelen, F.;2003. "Towards the Semantic Web: Ontology-driven Knowledge Management", John Willey & Sons.

[DavCG] David Consulting Group, http://www.davidconsultinggroup.com.

[Dublin] The Dublin Core Metadata Initiative. Available from http://www.dublincore.org

[Edr02] Japan Electronic Dictionary Research Institute , http://www.iijnet.or.jp/edr/.

[Fen97] Fenton, N.E.; Pfleeger, S.L., 1997, "Software Metrics: A Rigorous and Practical Approach", 2nd Ed., PWS Publishing Company.

[Fer97] Fernández López, M.; Gómez-Pérez, A. and Juristo, N., 1997, "METHONTOLOGY: From Ontological Art Towards Ontological Engineering". Proceed. of the AAAI Symposium. University of Stanford; P.A., California, US, pp. 33-40.

[Fer99a] Fernández-López, M. 1999. "Overview Of Methodologies for Building Ontologies". In Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5) Stockholm, Sweden, August 2, 1999.

[Fer99b] Fernández-López, M.; Gómez-Pérez, A. and Pazos-Sierra, J. 1999. "Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design

Environment". *IEEE Intelligent Systems & their applications*. January/February, pp. 37-46.

[Fra99] Frank van Harmelen, Dieter Fensel. "Practical Knowledge Representation for the Web", Proceedings of the IJCAI-99 workshop on Intelligent Information Integration Stockholm, Sweden,1999.

[Gar04] García, F.; Ruiz, F.; Bertoa, M. F.; Calero, C.; Genero, M.; Olsina, L.; Martín, M.; Quer, C.; Tondori, N.; Abrahao, S.; Vallecillo, A. and Piattini, M. 2004. "Una Ontología de la Medición del Software". Informe Técnico, Universidad de Castilla-La Mancha. Ciudad Real. España.

[Gen03] Genero, M.; Ruiz, F.; Piattini, M.; García, F.; and Calero, C.; 2003, "An Ontology for Software Measurement", Proceed. of SEKE'2003, 15th International Conference on Software Engineering and Knowledge Engineering, San Francisco, US.

[Gom96] Gomez-Perez, A.; Fernandez, M. and De Vicente, A. 1996. "Towards a Method to Conceptualize Domain Ontologies". Working notes of the workshop Ontological Engeniering. ECAI'96 pp. 41-52.

[Gom99] Gómez-Pérez, A., 1999. "Ontological Engineering: A State Of The Art". Expert Update. British Computer Society. Autumn. Vol. 2. Nº 3.

[Gru93] Gruber T. R. . "A translation approach to portable ontology specifications". Knowledge Acquisition, 5:199–220, 1993.

[Gru94a], Gruber T. R. and Olsen G. R. 1994. "An Ontology for Engineering Mathematics", Knowledge Systems Laboratory, Stanford University disponible en: http://www.ksl.stanford.edu/knowledge-sharing/papers/engmath.html.

[Gru94b] Gruber T. and Gerbaux F., 1994. "Frame ontology". ftp://ftp.ksl.stanford.edu/pub/knowledge-sharing/ontologies/html/frame-ontology/index.html

[Gru95] Gruber, T. "Towards Principles for the Design of Ontologies used for Knowledge Sharing". *International Journal of Human-Computer Studies*, 43(5/6), pp. 907-928. (1995).

[Grü95] Grüninger, M. and Fox, M.S. 1995. "Methodology for the design and evaluation of ontologies". Workshop on Basic Ontological Issues in Knowledge Sharing. Montreal, Canada.

[Gua95] Guarino, N. 1995. "Formal Ontology, Conceptual Analysis and Knowledge Representation: The Role of Formal Ontology in the Information Technology",

International Journal of Human-Computer Studies, Vol. 43, Nos. 5/6, 1995, pp. 625-640.

[Hau00] Haustein, S.; 2000. "RDF for object serialization". Start of thread on www-rdf-interest mailing list. http://lists.w3.org/Archives/Public/www-rdf-interest/2000Feb/0157.html .

[Hay04] Hayes P. (editor). WWW Consortium, "RDF Semantics",W3C Recommendation 10 February 2004, http://www.w3.org/TR/rdf-mt/.

[Hei97] van Heijst, G.; Schreiber, A.Th. and Wielinga, B.J. , 1997. "Using Explicit Ontologies in KBS Development", International Journal of Human-Computer Studies, 46(2/3): pp. 183–292.

[IEEE90] IEEE 610.12:1990 International Standard, Glossary of Software Engineering Terminology.

[ISO9126] ISO/IEC 9126-1: 2001 International Standard, Software Engineering - Product Quality - Part 1: Quality Model.

[ISO12207] ISO/IEC 12207-1:1994 International Standard, Information technology - Software life cycle processes .

[ISO14598] ISO/IEC 14598-1:1999 International Standard, Information technology - Software product evaluation - Part 1: General Overview.

[ISO15939] ISO/IEC 15939: 2002 International Standard, Software Engineering - Software Measurement Process.

[Kar01] Gregory Karvounarakis, Vassilis Christophides, Dimitris Plexousakis, and Sofia Alexaki. Querying community web portals. Technical report, Institute of Computer Science, FORTH, Heraklion, Greece, 2001.

[Kar02] Karvounarakis, G.; Alexaki, S.; Christophides, V.; Plexousakis, D. and Sholl, M.; 2002. "RQL A Declarative Query Language for RDF". Technical report, Institute of Computer Science, FORTH, Heraklion, Greece.

[Kitch01] Kitchenham B.A.; Hughes R.T.; Linkman S.G.; 2001, "Modeling Software Measurement Data". *IEEE Transactions on Software Engineering*, 27(9), pp. 788-804.

[Kly04] Klyne G.; Reynolds F.; Woodrow C.; Ohto H.; Hjelm J.; Butler M. and Tran L., 2004. "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0". W3C Recommendation 15 January 2004, http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/.

[Kog03] Kogut, P.; Cranefield, S.; Hart, L.; Dutra, M.; Baclawski, K.; Kokar, M. and Smith, J.; 2003. "UML for Ontology Development". http://www.sandsoft.com/docs/ker4.pdf .

[Las99] Lassila O. and Swick R. R. (editores). "Resource description framework (RDF): Model and syntax specification", 1999. W3C Recommendation 1999-02-22, http://www.w3.org/TR/REC-rdf-syntax.

[Law03] Lawler J. and Kitchenham B.A., 2003. "Measurement Modeling Technology". *IEEE Software*, 20(3), pp. 68-75.

[Man02] Manola, F.; Miller E., WWW Consortium, March 2002, "RDF Primer", http://www.w3.org/TR/2002/WD-rdf-primer-20020319/.

[Mar02] Martín, M.A.; Bertoa, M.F.; Vallecillo, A. and Olsina, L., 2002. "Hacia un Enfoque Semántico para la Catalogación de Métricas". Proceedings (en CD-ROM) del VII Congreso Argentino de Ciencias de la Computación, (CACIC), BsAs, Arg.

[Mar03a] Martín, M.A.; Molina, H.; Papa, F.; Fons, J.; Pastor, O. and Olsina, L. 2003. "Aspectos de Diseño Arquitectural y Semántico para un Sistema Web de Catalogación de Métricas". in Proceedings del VI Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software (IDEAS), Asunción, Paraguay, pp. 224-236.

[Mar03b] Martín M. and Olsina, L. 2003. "Towards An Ontology for Software Metrics and Indicators as the Foundation for a Cataloging Web System" . In IEEE C. Press Proceeding of the 1st Latin American Web Congress (LA-Web), Santiago de Chile, 10-12 Nov 2003, pp 103-113.

[McC76] McCabe T.J., 1976, "A complexity measure", *IEEE Transactions on Software Engineering*, 2(4), pp. 308-320.

[Men01] Mendes, E.; Mosley, N.; Counsell, S.; 2001, "Web Metric –Estimating Design and Authoring Effort", *IEEE Multimedia*, V. 8, Nº 1, pp. 50-57.

[Mil01] Miller, L., 2001. "RDF Squish query language and Java implementation". Public draft, Institute for Learning and Research Technology, 2001. Ver http://ilrt.org/ discovery/2001/02/squish/.

[Miz95] Mizoguchi, R.; Vanwelkenhuysen, J. and Ikeda M., 1995. "Task ontology for reuse of problem solving knowledge". In 2nd International Conference on Very Large-Scale Knowledge Bases., pages 46–57. IOS Press, Amsterdam, NL.

[Mol04] Molina, H.; Papa, F.; Martín, M.A. and Olsina, L., 2004. "Semantic Search and Navigation Capabilities for the Metrics and Indicators Cataloging Web System". To appear in Proceedings JAIIO, 33th. Argentine Conference on Computer Science and Operacional Research (JAIIO), Córdoba, Argentina.

[New02] Newcomer, E.; 2002, "Understanding Web services: XML, WSDL, SOAP, and UDDI". Addison-Wesley, Boston, MA.

[Oil]] Oil directory site. http://www.ontoknowledge.org/oil/.

[Ols98] Olsina, L. 1998. "Hypermedia Engineering: Flexible Process Model to support Hypermedia Application Development", Master Thesis on Software Engineering (in Spanish), UNLP, La Plata, Ar, Feb 98.

[Ols00] Olsina L., "Quantitative Methodology for Evaluation and Comparison of Web Site Quality", Doctoral Thesis, (in Spanish), Ciencias Exactas School, UNLP, La Plata, Argentina, Abril-2000.

[Ols01] Olsina, L.; González Rodríguez, J.; Lafuente, G.J.; Pastor, O., 2001, "Towards Automated Web Metrics", VIII Quality Brazilian Workshop, RJ-Br, pp. 74-86.

[Ols02a] Olsina L., Rossi G., "Measuring Web Application Quality with WebQEM", *IEEE Multimedia*, 9(4), pp. 20-29, 2002.

[Ols02b] Olsina, L.; Bertoa M. F.; Lafuente G. J., Martín M. A., Katrib M., Vallecillo A., . 2002, "Un Marco Conceptual para la Definición y Explotación de Métricas de Calidad", In proceed. of Jornadas de Ingeniería de Software y Bases de Datos (JISBD'02), Madrid, Spain, pp. 189-199.

[Ols03] Olsina, L.; Martin, M. A.; Fons, J.; Abrahao, S. and Pastor, O; 2003. "Towards the Design of a Metrics Cataloging System by Exploiting Conceptual and Semantic Web Approaches". In proceed. of Springer. Int'l Conference on Web Engineering 2003 (ICWE'03), Oviedo, Spain, pp. 324-333.

[Ols04] Olsina, L. and Martín, M., 2004. "Ontology for Software Metrics and Indicators". To appear in *Journal of Web Engineering*, Rinton Press, US, Vol 3 Nº X, 2004, ISSN 1540-9589.

[Pas01] Pastor, O.; Abrahão, S. M.; Fons, J. J., 2001. "Object-Oriented Approach to Automate Web Applications Development, 2nd Int'l Conference on Electronic Commerce and Web Technologies (EC-Web'01), Munich, Germany, Springer Verlag, pp. 16-28.

[SMLab] Laboratorio de Medición de Software (SMLab), http://irb.cs.uni-magdeburg.de/sw-eng/us/index.shtml.

[Sow00] Sowa, J. F. "Knowledge Representation: Logical, Philosophical, and Computational Foundations", Brooks Cole Publishing Co., Pacific Grove, CA 2000. Disponible parcialmente en http://www.jfsowa.com/ontology/.

[Sta00] Staab S., Erdmann M., Maedche A. And Decker S. "An extensible approach for modeling ontologies in RDF(S)". First Workshop on the Semantic Web at the Fourth European Conference International Workshop on Research and Advanced Technology for Digital Libraries, Lisbon, Portugal 18-20 September, 2000.

[Ste00] Stein, L. A.; Connolly, D.; McGuinness, D. (editors). "DAML Ontology language specification", released in October 2000. http://www.daml.org/2000/10/daml-ont.html.

[Stu98] Studer, R.; Benjamins V. R. and Fensel D. "Knowledge engineering: principles and methods". Data and Knowledge Engineering, 25(1-2):161–197, 1998.

[Tor04] Torres V.; Fons, J.; Asensi, O; Palechano V. , 2004. "Gettin Ready Web Engineering Methods for the Semantic Web Putting Ontologies into Pratice". In Proceedings of Fourth International Workshop on Web-Oriented Software Technologies (IWWOST'04), Munich, Germany.

[TotMe] Total Metrics, http://www.totalmetrics.com.

[TOVE] TOVE Ontologies site . http://www.eil.utoronto.ca/tove/toveont.html

[Usc95] Uschold, M. and King, M. 1995. "Towards a Methodology for Building Ontologies". Workshop on Basic Ontological Issues in Knowledge Sharing.

[Usc96a] Uschold, M. and Grüninger, M. 1996. "Ontologies: Principles Methods and Applications" Knowledge Engineering Review. Vol. 2.

[Usc96b] Uschold, M. 1996. "Building ontologies: Towards a unified methodology". In Expert Systems 96.

[Usc98] Uschold, M.; King, M.; Moralee, S. and Zorgios, Yannis, 1998. "The Enterprise Ontology". The Knowledge Engineering Review , Vol. 13, Special Issue on Putting Ontologies to Use (eds. Mike Uschold and Austin Tate). Disponible en: http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html

[Wan04] Wang, X. and Chan, C.W., 2001. "Ontology Modeling Using UML". 7th International Conference on Object Oriented Information Systems Conference (OOIS'2001), pp. 59-68.

[W3Csw] W3C, WWW Consortium, 2001, Semantic Web, http://www.w3.org/2001/sw/

[WebOnt] Web Ontology Working Group of W3C, WWW Consortium, 2001, http://www.w3.org/2001/sw/WebOnt

[WordN] WordNet site, http://www.cogsci.princeton.edu/~wn/

[ZDMIS] Zuse and Drake Measure Information System, http://home.t-online.de/home/horst.zuse/zdmis.html.

[Zus98] Zuse H., "A Framework of Software Measurement", Walter de Gruyter, Berlín-NY (1998).

Apéndice A

A.1 Listado del código RDF de la Ontología de Métricas e Indicadores

```
<!-- Esquema RDF de la Ontología de Metricas e Indicadores->
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <!-- Conceptos -->

        <rdfs:Class rdf:ID="Entity">
                <rdfs:label xml:lang="en">Entity</rdfs:label>
                <rdfs:comment>Object that is to be characterised by measuring its attributes
                </rdfs:comment>
                <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
        </rdfs:Class>

        <rdfs:Class rdf:ID="Attribute">
                <rdfs:label xml:lang="en">Attribute</rdfs:label>
                <rdfs:comment>A measurable physical or abstract property of an entity </rdfs:comment>
                <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
        </rdfs:Class>

        <rdfs:Class rdf:ID="Metric">
                <rdfs:label xml:lang="en">Metric</rdfs:label>
                <rdfs:comment>The defined measurement or calculation method and the measurement
                scale.</rdfs:comment>
                <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
        </rdfs:Class>

        <rdfs:Class rdf:ID="DirectMetric">
                <rdfs:label xml:lang="en">DirectMetric</rdfs:label>
                <rdfs:comment>A metric of an attribute that does not depend upon a metric of any  other
                attribute.</rdfs:comment>
                <rdfs:subClassOf rdf:resource="#Metric"/>
        </rdfs:Class>

        <rdfs:Class rdf:ID="IndirectMetric">
                <rdfs:label xml:lang="en">IndirectMetric</rdfs:label>
                <rdfs:comment>A metric of an attribute that is derived from metrics of one or more other
                attributes.</rdfs:comment>
                <rdfs:subClassOf rdf:resource="#Metric"/>
        </rdfs:Class>

        <rdfs:Class rdf:ID="Indicator">
                <rdfs:label xml:lang="en">Indicator</rdfs:label>
                <rdfs:comment>The defined calculation method and scale in addition to the model   and
                decision criteria in order to provide an estimate or evaluation of a calculable concept with
                respect to defined information needs. </rdfs:comment>
                <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
        </rdfs:Class>
```

```
<rdfs:Class rdf:ID="ElementaryIndicator">
        <rdfs:label xml:lang="en">ElementaryIndicator</rdfs:label>
        <rdfs:comment>An indicator that does not depend upon other indicators to evaluate  or
        estimate a calculable concept. </rdfs:comment>
        <rdfs:subClassOf rdf:resource="#Indicator"/>
</rdfs:Class>

<rdfs:Class rdf:ID="GlobalIndicator">
        <rdfs:label xml:lang="en">GlobalIndicator</rdfs:label>
        <rdfs:comment>An indicator that is derived from other indicators to evaluate or   estimate a
        calculable concept. </rdfs:comment>
        <rdfs:subClassOf rdf:resource="#Indicator"/>
</rdfs:Class>

<rdfs:Class rdf:ID="ElementaryModel">
        <rdfs:label xml:lang="en">ElementaryModel</rdfs:label>
        <rdfs:comment>Algorithm or function with associated decision criteria that model  an
        elementary indicator. </rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="GlobalModel">
        <rdfs:label xml:lang="en">GlobalModel</rdfs:label>
        <rdfs:comment>Algorithm or function with associated decision criteria that model  a global
        indicator.</rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="DecisionCriteria">
        <rdfs:label xml:lang="en">DecisionCriteria</rdfs:label>
        <rdfs:comment>Thresholds, targets, or patterns used to determine the need for action or
        further investigation, or to describe the level of confidence in a given
        result.</rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="CalculableConcept">
        <rdfs:label xml:lang="en">CalculableConcept</rdfs:label>
        <rdfs:comment>Abstract relationship between attributes of entities and information needs
        </rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="ConceptModel">
        <rdfs:label xml:lang="en">ConceptModel</rdfs:label>
        <rdfs:comment>The set of sub-concepts and the relationships between them, which
        provide the basis for specifying the concept requirement and its further evaluation or
        estimation. </rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="InformationNeed">
        <rdfs:label xml:lang="en">InformationNeed</rdfs:label>
        <rdfs:comment>Insight necessary to manage objectives, goals, risks, and
        problems.</rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
```

```
</rdfs:Class>

<rdfs:Class rdf:ID="Scale">
        <rdfs:label xml:lang="en">Scale</rdfs:label>
        <rdfs:comment>A set of values with defined properties </rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="CategoricalScale">
        <rdfs:label xml:lang="en">CategoricalScale</rdfs:label>
        <rdfs:comment>A scale where the measured or calculated values are categories, and can not
        be expressed in units, in a strict sense.</rdfs:comment>
        <rdfs:subClassOf rdf:resource="#Scale"/>
</rdfs:Class>

<rdfs:Class rdf:ID="NumericalScale">
        <rdfs:label xml:lang="en">NumericalScale</rdfs:label>
        <rdfs:comment>A scale where the measured or calculated values are numbers that can be
        expressed in units, in a strict sense.</rdfs:comment>
        <rdfs:subClassOf rdf:resource="#Scale"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Unit">
        <rdfs:label xml:lang="en">Unit</rdfs:label>
        <rdfs:comment>Particular quantity, defined and adopted by convention, with which  other
        quantities of the same kind are compared in order to express their magnitude relative to that
        quantity </rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Function">
        <rdfs:label xml:lang="en">Function</rdfs:label>
        <rdfs:comment>Algorithm or formula performed to combine two or more metrics.
        </rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Measurement">
        <rdfs:label xml:lang="en">Measurement</rdfs:label>
        <rdfs:comment>Activity that uses a metric definition in order to produce a measure's
        value.</rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Measure">
        <rdfs:label xml:lang="en">Measure</rdfs:label>
        <rdfs:comment>The number or category assigned to an attribute of an entity by making a
        measurement.</rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Calculation">
        <rdfs:label xml:lang="en">Calculation</rdfs:label>
        <rdfs:comment>Activity that uses an indicator definition in order to produce an indicator's
        value.</rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
```

```
        </rdfs:Class>

        <rdfs:Class rdf:ID="IndicatorValue">
                <rdfs:label xml:lang="en">IndicatorValue</rdfs:label>
                <rdfs:comment>The number or category assigned to a calculable concept by making a
                calculation.</rdfs:comment>
                <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
        </rdfs:Class>

        <rdfs:Class rdf:ID="Method">
                <rdfs:label xml:lang="en">Method</rdfs:label>
                <rdfs:comment>Logical sequence of operations and possible heuristics, specified
                generically, for allowing the realisation of an activity description.</rdfs:comment>
                <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
        </rdfs:Class>

        <rdfs:Class rdf:ID="MeasurementMethod">
                <rdfs:label xml:lang="en">MeasurementMethod</rdfs:label>
                <rdfs:comment>The particular logical sequence of operations and possible heuristics
                specified for allowing the realisation of a metric description by a
                measurement. </rdfs:comment>
                <rdfs:subClassOf rdf:resource="#Method"/>
        </rdfs:Class>

        <rdfs:Class rdf:ID="CalculationMethod">
                <rdfs:label xml:lang="en">CalculationMethod</rdfs:label>
                <rdfs:comment>The particular logical sequence of operations specified for allowing the
                realisation of a formula or indicator description by a calculation.</rdfs:comment>
                <rdfs:subClassOf rdf:resource="#Method"/>
        </rdfs:Class>

        <rdfs:Class rdf:ID="SoftwareTool">
                <rdfs:label xml:lang="en">SoftwareTool</rdfs:label>
                <rdfs:comment>It is a tool that automates partially or totally a measurement or calculation
                method.</rdfs:comment>
                <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
        </rdfs:Class>


<!-- Atributos de Entity -->

        <rdf:Property rdf:ID="entityName">
                <rdfs:label xml:lang="en">entityName</rdfs:label>
                <rdfs:domain rdf:resource="#Entity"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Name of an entity to be identified.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="entityDescription">
                <rdfs:label xml:lang="en">entityDescription</rdfs:label>
                <rdfs:domain rdf:resource="#Entity"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>An unambiguous description of the entity meaning</rdfs:comment>
        </rdf:Property>
```

```
<!-- Atributos de Attribute -->

        <rdf:Property rdf:ID="attributeName">
                <rdfs:label xml:lang="en">attributeName</rdfs:label>
                <rdfs:domain rdf:resource="#Attribute"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Name of an attribute to be identified.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="attributeDefinition">
                <rdfs:label xml:lang="en">attributeDefinition</rdfs:label>
                <rdfs:domain rdf:resource="#Attribute"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>An unambiguous description of the attribute meaning.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="attributeObjective">
                <rdfs:label xml:lang="en">attributeObjective</rdfs:label>
                <rdfs:domain rdf:resource="#Attribute"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Goal or purpose to measuring this attribute.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="attributeType">
                <rdfs:label xml:lang="en">attributeType</rdfs:label>
                <rdfs:domain rdf:resource="#Attribute"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Attributes can be internal or external.</rdfs:comment>
        </rdf:Property>

<!-- Atributos de Metric -->

        <rdf:Property rdf:ID="metricName">
                <rdfs:label xml:lang="en">metricName</rdfs:label>
                <rdfs:domain rdf:resource="#Metric"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Name of an metric to be identified.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="metricValueInterpretation">
                <rdfs:label xml:lang="en">metricValueInterpretation</rdfs:label>
                <rdfs:domain rdf:resource="#Metric"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>An unambiguous textual statement for helping stakeholders to understand
                the obtained value meaning, e.g. the closer to zero the better.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="metricObjective">
                <rdfs:label xml:lang="en">metricObjective</rdfs:label>
                <rdfs:domain rdf:resource="#Metric"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Goal or purpose for applying the specific metric.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="metricReferences">
                <rdfs:label xml:lang="en">metricReferences</rdfs:label>
```

```
        <rdfs:domain rdf:resource="#Metric"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
        <rdfs:comment>References to bibliographical or URL resources, where additional and
        authoritative information of the given metric can be consulted.</rdfs:comment>
    </rdf:Property>

    <rdf:Property rdf:ID="metricValueType">
        <rdfs:label xml:lang="en">metricValueType</rdfs:label>
        <rdfs:domain rdf:resource="#Metric"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
        <rdfs:comment>Type of value that a metric can assume. It can be a symbol, integer or float.
        </rdfs:comment>
    </rdf:Property>

    <rdf:Property rdf:ID="metricAccuracy">
        <rdfs:label xml:lang="en">metricAccuracy</rdfs:label>
        <rdfs:domain rdf:resource="#Metric"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
        <rdfs:comment>A quantifier of the accuracy level inherent to the way of producing the
        value of a metric.</rdfs:comment>
    </rdf:Property>


<!-- Atributos de Indicator -->

    <rdf:Property rdf:ID="indicatorName">
        <rdfs:label xml:lang="en">indicatorName</rdfs:label>
        <rdfs:domain rdf:resource="#Indicator"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
        <rdfs:comment>Name of an indicator to be identified.</rdfs:comment>
    </rdf:Property>

    <rdf:Property rdf:ID="indicatorAccuracy">
        <rdfs:label xml:lang="en">indicatorAccuracy</rdfs:label>
        <rdfs:domain rdf:resource="#Indicator"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
        <rdfs:comment>A quantification of the accuracy level inherent to the way of producing the
        value of an indicator.</rdfs:comment>
    </rdf:Property>

    <rdf:Property rdf:ID="indicatorReferences">
        <rdfs:label xml:lang="en">indicatorReferences</rdfs:label>
        <rdfs:domain rdf:resource="#Indicator"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
        <rdfs:comment>References to bibliographical or URL resources, where additional and
        authoritative information of the given indicator can be consulted.</rdfs:comment>
    </rdf:Property>

    <rdf:Property rdf:ID="indicatorValueType">
        <rdfs:label xml:lang="en">indicatorValueType</rdfs:label>
        <rdfs:domain rdf:resource="#Indicator"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
        <rdfs:comment>Type of value that an indicator can assume. It can be a symbol, integer or
        float.</rdfs:comment>
    </rdf:Property>
```

```
<!-- Atributos de ElementaryModel -->

        <rdf:Property rdf:ID="elementaryModelName">
                <rdfs:label xml:lang="en">elementaryModelName</rdfs:label>
                <rdfs:domain rdf:resource="#ElementaryModel"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Name of an elementary model to be identified.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="elementaryModelSpecification">
                <rdfs:label xml:lang="en">elementaryModelSpecification</rdfs:label>
                <rdfs:domain rdf:resource="#ElementaryModel"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>A formal or semiformal representation of an elementary model. It can be
                e.g. a mathematical or logical representation.</rdfs:comment>
        </rdf:Property>

<!-- Atributos de GlobalModel -->

        <rdf:Property rdf:ID="globalModelName">
                <rdfs:label xml:lang="en">globalModelName</rdfs:label>
                <rdfs:domain rdf:resource="#GlobalModel"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Name of an global model to be identified.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="globalModelSpecification">
                <rdfs:label xml:lang="en">globalModelSpecification</rdfs:label>
                <rdfs:domain rdf:resource="#GlobalModel"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>A formal or semiformal representation of a global model. It can be e.g. a
                mathematical or logical representation.</rdfs:comment>
        </rdf:Property>


<!-- Atributos de DecisionCriteria -->

        <rdf:Property rdf:ID="decisionCriteriaName">
                <rdfs:label xml:lang="en">decisionCriteriaName</rdfs:label>
                <rdfs:domain rdf:resource="#DecisionCriteria"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Name of an decision criteria to be identified.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="decisionCriteriaDescription">
                <rdfs:label xml:lang="en">decisionCriteriaDescription</rdfs:label>
                <rdfs:domain rdf:resource="#DecisionCriteria"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>An unambiguous description of the decision criterion
                meaning.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="decisionCriteriaRange">
                <rdfs:label xml:lang="en">decisionCriteriaRange</rdfs:label>
                <rdfs:domain rdf:resource="#DecisionCriteria"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
```

```
            <rdfs:comment>Numerical values specifying e.g. the lower/upper thresholds for a given
            criterion.</rdfs:comment>
        </rdf:Property>

<!-- Atributos de CalculableConcept -->

        <rdf:Property rdf:ID="calculableConceptName">
            <rdfs:label xml:lang="en">calculableConceptName</rdfs:label>
            <rdfs:domain rdf:resource="#CalculableConcept"/>
            <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
            <rdfs:comment>Name of an calculable concept to be identified.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="calculableConceptDescription">
            <rdfs:label xml:lang="en">calculableConceptDescription</rdfs:label>
            <rdfs:domain rdf:resource="#CalculableConcept"/>
            <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
            <rdfs:comment>An unambiguous description of the calculable concept
            meaning.</rdfs:comment>
        </rdf:Property>

<!-- Atributos de ConceptModel -->

        <rdf:Property rdf:ID="conceptModelName">
            <rdfs:label xml:lang="en">conceptModelName</rdfs:label>
            <rdfs:domain rdf:resource="#ConceptModel"/>
            <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
            <rdfs:comment>Name of an concept model to be identified.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="conceptModelSpecification">
            <rdfs:label xml:lang="en">conceptModelSpecification</rdfs:label>
            <rdfs:domain rdf:resource="#ConceptModel"/>
            <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
            <rdfs:comment>A formal or semiformal representation of a concept
            model.</rdfs:comment>
        </rdf:Property>

<!-- Atributos de InformationNeed -->

        <rdf:Property rdf:ID="InformationNeedDescription">
            <rdfs:label xml:lang="en">InformationNeedDescription</rdfs:label>
            <rdfs:domain rdf:resource="#InformationNeed"/>
            <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
            <rdfs:comment>An unambiguous textual statement describing the information
            needs.</rdfs:comment>
        </rdf:Property>

<!-- Atributos de Scale -->

        <rdf:Property rdf:ID="scaleType">
            <rdfs:label xml:lang="en">scaleType</rdfs:label>
            <rdfs:domain rdf:resource="#Scale"/>
            <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
            <rdfs:comment>The type of scales depends on the nature of the relationship between values
```

```
                of the scale [7]. These types of scales are commonly defined: nominal, ordinal (restricted or
                unrestricted), interval, ratio, and absolute. </rdfs:comment>
        </rdf:Property>

<!-- Atributos de CategoricalScale -->

        <rdf:Property rdf:ID="categoricalScaleAllowedValues">
                <rdfs:label xml:lang="en"> categoricalScaleAllowedValues</rdfs:label>
                <rdfs:domain rdf:resource="#CategoricalScale"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>List of literals indicating the valid values of a categorical scale.
                </rdfs:comment>
        </rdf:Property>

<!-- Atributos de NumericalScale -->

        <rdf:Property rdf:ID="numericalScaleType">
                <rdfs:label xml:lang="en">numericalScaleType</rdfs:label>
                <rdfs:domain rdf:resource="#NumericalScale"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>A numerical scale can be continuous or discrete. </rdfs:comment>
        </rdf:Property>

<!-- Atributos de Function -->

        <rdf:Property rdf:ID="functionSpecification">
                <rdfs:label xml:lang="en">functionSpecification</rdfs:label>
                <rdfs:domain rdf:resource="#Function"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>A formal or semiformal representation of a function. Synonymous:
                formula.</rdfs:comment>
        </rdf:Property>

<!-- Atributos de Measurement -->

        <rdf:Property rdf:ID="measurementTimePoint">
                <rdfs:label xml:lang="en">measurementTimePoint</rdfs:label>
                <rdfs:domain rdf:resource="#Measurement"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Instant when a measurement is performed.</rdfs:comment>
        </rdf:Property>

<!-- Atributos de Measure -->

        <rdf:Property rdf:ID="measureValue">
                <rdfs:label xml:lang="en">measureValue</rdfs:label>
                <rdfs:domain rdf:resource="#Measure"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Numerical or categorical result assigned to an attribute. Synonymous:
                data.</rdfs:comment>
        </rdf:Property>
```

```
<!-- Atributos de Calculation -->

        <rdf:Property rdf:ID="calculationTimePoint">
                <rdfs:label xml:lang="en">calculationTimePoint</rdfs:label>
                <rdfs:domain rdf:resource="#Calculation"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Instant when a calculation is performed.</rdfs:comment>
        </rdf:Property>

<!-- Atributos de IndicatorValue -->

        <rdf:Property rdf:ID="indicatorValueValue">
                <rdfs:label xml:lang="en">indicatorValueValue</rdfs:label>
                <rdfs:domain rdf:resource="#IndicatorValue"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Numerical or categorical result assigned to an indicator Synonymous:
                data.</rdfs:comment>
        </rdf:Property>

<!-- Atributos de Method -->

        <rdf:Property rdf:ID="methodName">
                <rdfs:label xml:lang="en">methodName</rdfs:label>
                <rdfs:domain rdf:resource="#Method"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Name of a method to be identified.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="methodSpecification">
                <rdfs:label xml:lang="en">methodSpecification</rdfs:label>
                <rdfs:domain rdf:resource="#Method"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>A formal or semiformal description of a method.</rdfs:comment>
        </rdf:Property>

<!-- Atributos de MeasurementMethod -->

        <rdf:Property rdf:ID="measurementMethodType">
                <rdfs:label xml:lang="en">measurementMethodType</rdfs:label>
                <rdfs:domain rdf:resource="#MeasurementMethod"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Indicates the type of measurement method that depends on the nature of the
                operations used to quantify an attribute.  Two types may be distinguished: subjective
                (quantification involving human judgement), and objective (quantification based on
                numerical rules) .</rdfs:comment>
        </rdf:Property>

<!-- Atributos de SoftwareTool -->

        <rdf:Property rdf:ID="softwareToolName">
                <rdfs:label xml:lang="en">softwareToolName</rdfs:label>
                <rdfs:domain rdf:resource="#SoftwareTool"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Name of a software tool to be identified.</rdfs:comment>
        </rdf:Property>
```

```
        <rdf:Property rdf:ID="softwareToolDescription">
                <rdfs:label xml:lang="en">softwareToolDescription</rdfs:label>
                <rdfs:domain rdf:resource="#SoftwareTool"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>An unambiguous description of a software tool.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="softwareToolVersion">
                <rdfs:label xml:lang="en">softwareToolVersion</rdfs:label>
                <rdfs:domain rdf:resource="#SoftwareTool"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Number that indicates a software tool version.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="softwareToolProvider">
                <rdfs:label xml:lang="en">softwareToolProvider</rdfs:label>
                <rdfs:domain rdf:resource="#SoftwareTool"/>
                <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
                <rdfs:comment>Indicates the name (or URL) of a software tool supplier.</rdfs:comment>
        </rdf:Property>


<!-- Relaciones -->

        <rdf:Property rdf:ID="SubEntity">
                <rdfs:label xml:lang="en">SubEntity</rdfs:label>
                <rdfs:domain rdf:resource="#Entity"/>
                <rdfs:range rdf:resource="#Entity"/>
                <rdfs:comment>An entity may be composed of none or several subentities, which are in
                turn entities.  </rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="Quantifies">
                <rdfs:label xml:lang="en">Quantifies</rdfs:label>
                <rdfs:domain rdf:resource="#Metric"/>
                <rdfs:range rdf:resource="#Attribute"/>
                <rdfs:comment>One or more metrics can quantify an attribute.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="AutomatedBy">
                <rdfs:label xml:lang="en">AutomatedBy</rdfs:label>
                <rdfs:domain rdf:resource="#Method"/>
                <rdfs:range rdf:resource="#SoftwareTool"/>
                <rdfs:comment>One or more methods can be automated by none or several software
                tools.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="Includes">
                <rdfs:label xml:lang="en">Includes</rdfs:label>
                <rdfs:domain rdf:resource="#Indicator"/>
                <rdfs:range rdf:resource="#CalculationMethod"/>
                <rdfs:comment>An indicator includes a specific calculation method.</rdfs:comment>
        </rdf:Property>

        <rdf:Property rdf:ID="Includes">
                <rdfs:label xml:lang="en">Includes</rdfs:label>
```

```
        <rdfs:domain rdf:resource="#Metric"/>
        <rdfs:range rdf:resource="#Method"/>
        <rdfs:comment>A metric includes a specific measurement and/or calculation
        method.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Contains">
        <rdfs:label xml:lang="en">Contains</rdfs:label>
        <rdfs:domain rdf:resource="#Metric"/>
        <rdfs:range rdf:resource="#Scale"/>
        <rdfs:comment>A metric contain a specific scale.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Contains">
        <rdfs:label xml:lang="en">Contains</rdfs:label>
        <rdfs:domain rdf:resource="#Indicator"/>
        <rdfs:range rdf:resource="#Scale"/>
        <rdfs:comment>An indicator contain a specific scale.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Evaluates/Estimates">
        <rdfs:label xml:lang="en">Evaluates/Estimates</rdfs:label>
        <rdfs:domain rdf:resource="#Indicator"/>
        <rdfs:range rdf:resource="#CalculableConcept"/>
        <rdfs:comment>An indicator evaluates/estimates a calculable concept.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Produces">
        <rdfs:label xml:lang="en">Produces</rdfs:label>
        <rdfs:domain rdf:resource="#Measurement"/>
        <rdfs:range rdf:resource="#Measure"/>
        <rdfs:comment>A measurement activity produces a specific measure
        value.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Produces">
        <rdfs:label xml:lang="en">Produces</rdfs:label>
        <rdfs:domain rdf:resource="#Calculation"/>
        <rdfs:range rdf:resource="#IndicatorValue"/>
        <rdfs:comment>A calculation activity produces a specific indicator value.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="RelatedMetric">
        <rdfs:label xml:lang="en">RelatedMetric</rdfs:label>
        <rdfs:domain rdf:resource="#IndirectMetric"/>
        <rdfs:range rdf:resource="#Metric"/>
        <rdfs:comment>An indirect metric can be structured on the basis of two or more related
        metrics.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="SpecifiedBy">
        <rdfs:label xml:lang="en">SpecifiedBy</rdfs:label>
        <rdfs:domain rdf:resource="#IndirectMetric"/>
        <rdfs:range rdf:resource="#Function"/>
        <rdfs:comment>An indirect metric is specified by a given function (or formula).
        </rdfs:comment>
```

```
</rdf:Property>

<rdf:Property rdf:ID="Has">
        <rdfs:label xml:lang="en">Has</rdfs:label>
        <rdfs:domain rdf:resource="#ElementaryModel"/>
        <rdfs:range rdf:resource="#DecisionCriteria"/>
        <rdfs:comment>An indicator model has one or more decision criteria.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Has">
        <rdfs:label xml:lang="en">Has</rdfs:label>
        <rdfs:domain rdf:resource="#GlobalModel"/>
        <rdfs:range rdf:resource="#DecisionCriteria"/>
        <rdfs:comment>An indicator model has one or more decision criteria.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="ExpressedIn">
        <rdfs:label xml:lang="en">ExpressedIn</rdfs:label>
        <rdfs:domain rdf:resource="#NumericalScale"/>
        <rdfs:range rdf:resource="#Unit"/>
        <rdfs:comment>A numerical scale must be expressed in a specific unit. (In a strict sense,
        there is no idea of unit for categorical scales)</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Associated_with">
        <rdfs:label xml:lang="en">Associated_with</rdfs:label>
        <rdfs:domain rdf:resource="#Attribute"/>
        <rdfs:range rdf:resource="#Entity"/>
        <rdfs:comment>One or more measurable attributes are associated with one or more
        entities.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Combines">
        <rdfs:label xml:lang="en">Combines</rdfs:label>
        <rdfs:domain rdf:resource="#CalculableConcept"/>
        <rdfs:range rdf:resource="#Attribute"/>
        <rdfs:comment>A calculable concept combines (associates) one or more measurable
        attributes. </rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Describes">
        <rdfs:label xml:lang="en">Describes</rdfs:label>
        <rdfs:domain rdf:resource="#CalculableConcept"/>
        <rdfs:range rdf:resource="#InformationNeed"/>
        <rdfs:comment>One or more calculable concepts are defined in order to satisfy a concrete
        information need. So, a calculable concept describes a concrete information
        need.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Interprets">
        <rdfs:label xml:lang="en">Interprets</rdfs:label>
        <rdfs:domain rdf:resource="#ElementaryIndicator"/>
        <rdfs:range rdf:resource="#Metric"/>
        <rdfs:comment>An elementary indicator may interpret none or one specific
        metric.</rdfs:comment>
</rdf:Property>
```

```
<rdf:Property rdf:ID="Modeled_by">
        <rdfs:label xml:lang="en">Modeled_by</rdfs:label>
        <rdfs:domain rdf:resource="#ElementaryIndicator"/>
        <rdfs:range rdf:resource="#ElementaryModel"/>
        <rdfs:comment>An elementary indicator is modeled by one elementary
        model.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Modeled_by">
        <rdfs:label xml:lang="en">Modeled_by</rdfs:label>
        <rdfs:domain rdf:resource="#GlobalIndicator"/>
        <rdfs:range rdf:resource="#GlobalModel"/>
        <rdfs:comment>An global indicator is modeled by one global model.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Refers-to">
        <rdfs:label xml:lang="en">Refers-to</rdfs:label>
        <rdfs:domain rdf:resource="#Measurement"/>
        <rdfs:range rdf:resource="#Metric"/>
        <rdfs:comment>A measurement activity is related to a metric (description). None or several
        measurements can be made on the same metric.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Related_indicators">
        <rdfs:label xml:lang="en">Related_indicators</rdfs:label>
        <rdfs:domain rdf:resource="#GlobalIndicator"/>
        <rdfs:range rdf:resource="#Indicator"/>
        <rdfs:comment>A global indicator can be structured (aggregated) on the basis of two or
        more related indicators.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Related-to">
        <rdfs:label xml:lang="en">Related-to</rdfs:label>
        <rdfs:domain rdf:resource="#Calculation"/>
        <rdfs:range rdf:resource="#Indicator"/>
        <rdfs:comment>A calculation activity is related to an indicator (description). None or
        several calculations can be made on the same indicator.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="Represented_by">
        <rdfs:label xml:lang="en">Represented_by</rdfs:label>
        <rdfs:domain rdf:resource="#CalculableConcept"/>
        <rdfs:range rdf:resource="#ConceptModel"/>
        <rdfs:comment>A calculable concept can be represented by none or several concept
        models. </rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="SubConcept">
        <rdfs:label xml:lang="en">SubConcept</rdfs:label>
        <rdfs:domain rdf:resource="#CalculableConcept"/>
        <rdfs:range rdf:resource="#CalculableConcept"/>
        <rdfs:comment>A calculable concept may be composed of none or several sub-concepts,
        which are in turn calculable concepts. </rdfs:comment>
</rdf:Property>
```