

Definición de una ontología para sistemas de información de administración de laboratorios (OLIMS)

Definition of an ontology for laboratory information management systems (OLIMS)

Grimaldos, Nelson¹; Besembel, Isabel^{1*}; Rivero, Dulce¹; Narciso, Flor¹; Díaz, Gilberto¹; Hernández, Domingo¹; Hidrobo, Francisco² y Ramírez, Vicente³

¹Departamento de Computación. Escuela de Sistemas. Facultad de Ingeniería. ULA

²Facultad de Ciencias. ULA

³Departamento de Investigación de Operaciones. Escuela de Ingeniería de Sistemas. ULA

Mérida, 5101-Venezuela

*ibc@ula.ve

Recibido: 28-10-2008

Revisado: 15-09-2009

Resumen

La World Wide Web (WWW) ha tenido un éxito tremendo al lograr que un rango increíble de información y servicios sean accesibles a billones de usuarios a nivel mundial, sin embargo, se ha hecho cada vez más difícil administrar el constante incremento de la información disponible y la necesidad de utilizar tal información para realizar tareas más complejas que las búsquedas basadas solo en palabras claves y recuperación de datos. El problema se expone aún más con la falta de estructura y carácter descentralizado en la naturaleza de la Web. Con el ideal de lograr que los recursos en la Web sean mucho más útiles, se ha propuesto que la información sea dada con un significado mejor definido, que forme una red de información lógica, esta idea se ha convertido en lo que se conoce como Web Semántica. Las ontologías son teorías de contenido acerca de los tipos de objetos, propiedades y relaciones que existen entre ellos y que son posibles en un dominio específico del conocimiento. Proveen vocabularios de términos especiales con significados formalmente definidos y accesibles por máquinas (a través del software diseñado con este propósito) para describir el entendimiento sobre un ámbito particular que pueden ser usados en la Web Semántica. El principal objetivo de este trabajo es definir formalmente una ontología en el lenguaje OWL (Ontology Web Language), que modele el dominio de un Sistema de Administración de Laboratorios denominada OLIMS (Ontology for Laboratory Information Management Systems). Se utilizó la metodología Methontology. OLIMS fue generada automáticamente desde el diagrama de clases UML (Unified Modeling Language) y está embebida en el sistema SIGELAB, es de fácil extensión y mantenimiento, y por ello hace uso de recursos Web que cumplen con los estándares aprobados por el World Wide Web Consortium (W3C).

Palabras clave: Ontologías, Web semántica, sistemas de administración de laboratorios, sistemas de información.

Abstract

The World Wide Web (WWW) has obtained a great success allowing accessibility of an incredible range of information and services to billions of users at world level. However, nowadays it is becoming more difficult to manage the constant increase of the available information and the need of using such information to do more complex tasks than searches based on key words and data retrieval. The problem is more complex due to the lack of structure and decentralized nature of the Web. Looking for a better utilization of the Web resources it has been proposed that the information in the Web should be provided with a better defined meaning, forming a net of logic information. This idea has been defined as the Semantic Web. Ontologies are theories of the content about types of objects, properties and relationships that exist in a specific domain of the knowledge. Ontologies provide vocabularies of special terms with formal defined meanings accessible to machines throughout specifically designed softwares, to describe the knowledge of a particular domain that can be used in the semantic web. The main objective of this work is to formally define an ontology called OLIMS (Ontology for Laboratory Information Management Systems), in the OWL (Ontology Web Language) language for modeling the domain of the Laboratory Infor-

mation Management Systems (LIMS). We used the methodology called Methontology. OLIMS was automatically generated from the UML (Unified Modeling Language) class diagram, it is embedded in the SIGELAB system and it is of easy extensibility and maintainable. OLIMS uses web resources that are approved standards of the World Wide Web Consortium (W3C).

Key words: Ontologies, semantic Web, laboratory management systems, information systems.

1 Introducción

La World Wide Web (WWW) ha tenido un éxito tremendo logrando que un rango increíble de información y servicios sean accesibles a billones de usuarios a nivel mundial. Sin embargo, se ha hecho cada vez más difícil administrar el constante incremento de la información disponible y la necesidad de utilizar tal información para realizar tareas más complejas, que sólo búsquedas basadas en palabras claves y recuperación de datos (Patel, 2006). El problema se expone aún más con la falta de estructura y carácter descentralizado de la naturaleza de la web y en el enfoque dado por el HTML (Hypertext Markup Language) (HTML, 1990) donde la presentación está por encima del contenido. Con el fin de lograr que los recursos en la web sean más útiles, se ha propuesto que la información sea colocada junto con su significado, para formar una red de información lógica, conocida como Web Semántica (Berners-Lee, 2001; Davies, 2003; Davies, 2006).

Las investigaciones más recientes sobre la web semántica se han enfocado en el desarrollo de lenguajes para aumentar el contenido con marcado semántico y el establecimiento de ontologías, las cuales proveen términos para describir el conocimiento sobre un dominio específico. Según indica Chandrasekaran en (Chandrasekaran, 1999), el análisis ontológico clarifica la estructura del conocimiento. Dado un dominio, su ontología forma el corazón de cualquier sistema de representación de sabiduría para ese dominio. Sin las ontologías o la conceptualización que inspira el entendimiento, no puede existir un vocabulario para representarlo.

Nirenburg y Raskin (Nirenburg, 2004) señalan que las semánticas ontológicas son una teoría de significación en el lenguaje natural y una aproximación al procesamiento del lenguaje natural, el cual usa un modelo construido del mundo, u ontología, como recurso central para extraer y representar el significado de los textos en lenguaje natural, el razonamiento acerca del conocimiento derivado de dichos textos, así como la generación de textos en lenguaje natural basados en las representaciones de sus significados.

Una ontología es una especificación explícita y formal de una conceptualización de un dominio de interés (Gruber, 1995). Esta definición marca dos puntos claves: que el formalismo de los conceptos produce una descripción estricta y por tanto permite el razonamiento del computador; y que una ontología práctica está diseñada para algún dominio de interés particular, en este caso, el domi-

nio definido por los sistemas de gestión administrativa de laboratorios, conocidos como LIMS, por sus siglas en inglés (Laboratory Information Management Systems).

Según María Teresa Abad en (Abad, 2004) las tareas identificadas para el desarrollo de ontologías se clasifican en 3 categorías:

- Actividades ligadas al manejo del proyecto: planificación, control de seguimiento, y garantizar la calidad del producto.
- Actividades orientadas al desarrollo de la ontología: especificación, conceptualización, formalización, implementación y mantenimiento.
- Actividades integrales (necesarias para un buen desarrollo de la ontología): adquisición de conocimiento, integración con otras ontologías, evaluación y documentación.

En particular, la etapa de conceptualización se detalla en la Fig. 1.

En general, las metodologías proporcionan un conjunto de directrices que indican cómo llevar a cabo las actividades en cada diligencia y qué produce cada una de ellas. No existe una metodología estándar para la creación de ontologías, pero pueden distinguirse 4 pasos básicos en el proceso de creación.

- Identificación del propósito y del alcance: Se trata de especificar el contexto de la aplicación y el modelado del punto de vista que se desea describir. El contexto de la aplicación describe el dominio, los objetos de interés y las tareas que van a realizarse. El modelado del punto de vista describe el tipo de modelo, por. Ejem, dinámico-estático, funcional-causal, etc.
- Construcción de la ontología: se pueden distinguir las siguientes etapas: Captura, Codificación (representación explícita de la conceptualización en un lenguaje formal) e Integración de las ontologías existentes (determinar si se va a reutilizar y cómo hacerlo con alguna de las ontologías existentes).
- Evaluación del diseño definitivo: se tendrán en cuenta aspectos como la posible reutilización de la ontología construida.
- Documentación y reutilización: como es habitual, la documentación debe desarrollarse paralela a la realización de las etapas anteriores. Debe incluirse la justificación de las decisiones tomadas, la evaluación realizada, el conocimiento adicional para usarla, etc. También ha de ser indexada y colocada con las ontologías existentes para su posible reutilización.

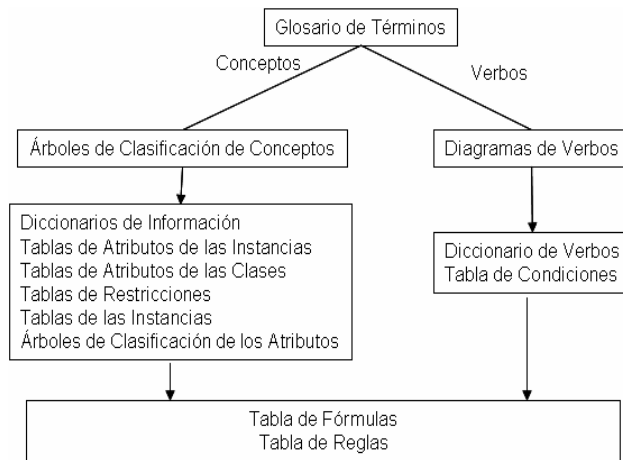


Fig. 1. Etapa de conceptualización (Abad, 2004)

Para la realización del componente OLIMS se utilizará la metodología Methontology (Gómez y col., 2004). Para mejorar la aplicabilidad de Methontology, ésta adoptó algunas ideas de la disciplina de la Ingeniería del Software, en específico, el proceso de desarrollo de ontologías está basado en las actividades identificadas en el estándar del Institute of Electrical and Electronics Engineers (IEEE) para el desarrollo de software (Schultz, 1997).

El proceso de desarrollo de la ontología mostrado en la Fig. 2, se refiere a qué actividades deben realizarse cuando se construyen ontologías. Identifica tres categorías de labores.

- Actividades de administración de la ontología: incluyen programación, control y garantía de calidad. La actividad de planificación identifican las tareas a realizarse, sus arreglos, y el tiempo y recursos necesitados. Las actividades de control garantizan la finalización de las tareas pretendidas. Al asegurar la eficacia se chequea la calidad de cada producto final de la metodología (ontología, software y documentación).
- Actividades orientadas al desarrollo de la ontología: agrupadas en actividades de pre-desarrollo, desarrollo y pos-desarrollo.
- Actividades de Mantenimiento de la Ontología: son realizadas al mismo tiempo que las actividades orientadas al desarrollo. Durante el soporte toman lugar las siguientes tareas. La adquisición del conocimiento, cuyo objetivo es conseguir "sabiduría" de los expertos o por aprendizaje de ontologías automáticas. Tareas de Evaluación que juzgan la ontología desarrollada, software y documentación contra un marco de referencia. Procesos de integración en caso que existan otras ontologías de carácter reutilizable y que necesiten ser combinadas. La mezcla produce una nueva ontología de la composición mientras que la formación establece la correspondencia preservada en la ontología original. La documentación detalla cada fase completada y mantiene registros de las ontologías, pro-

ducto, administración de la configuración, software y versiones de documentación con la finalidad de mantener un control de cambios.

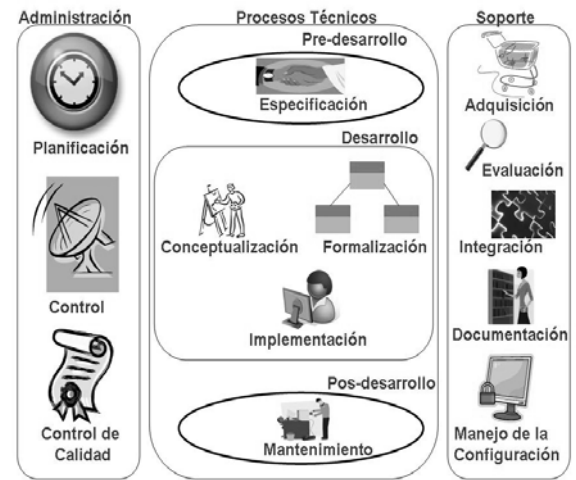


Fig. 2. Metodología

El objetivo de este trabajo es el logro de una definición formal de un Sistema de Administración de Laboratorios en la que los componentes se enlacen de manera lógica y cuyas definiciones sean únicas e inequívocas, y en aquellos casos en que sea necesario, se logre establecer las diferencias jerárquicas tanto de procesos como de administración del laboratorio para el claro entendimiento de los actores del sistema. Esta definición formal está expresada con dos lenguajes diferentes, el Lenguaje de Modelado Unificado UML (OMG, 2004) y en particular, el diagrama de clases que se definió en un proyecto anterior (Rivero y col., 2008) que representa el modelo de dominio, y el Lenguaje de Marcado para Ontologías OWL (OWL, 2004) que es el lenguaje estándar para la utilización de ontologías en la Web según el consorcio W3C.

El resto de este trabajo está organizado como sigue: la sección 2 está dedicada a la descripción de la gestión de laboratorios, la sección 3 presenta el modelado semántico de los conceptos involucrados en dicha gestión, la sección 4 muestra la ontología creada y generada automáticamente a partir del diagrama de clases de la sección anterior, denominada OLIMS y finalmente, la sección 5 presenta las conclusiones y trabajos futuros de esta investigación.

2 Gestión de laboratorio

Entre las actividades de pre-desarrollo de una ontología se encuentra la especificación que establece el propósito y el dominio de la misma. En esta sección se presenta la especificación utilizada para OLIMS.

El dominio de aplicación son los LIMS que dan soporte al sistema de negocios de cualquier laboratorio. El principal objetivo de estos sistemas administrar y gestionar los

procesos inherentes de los laboratorios que prestan servicios de análisis de muestras. Para cumplir con este objetivo principal se definen un conjunto de objetivos específicos que se listan a continuación:

- Evitar el desabastecimiento de material necesario en los análisis.
- Suministrar información oportuna a los clientes sobre el estado actual de los análisis solicitados.
- Emitir los resultados de los análisis bajo la figura de certificación.
- Planificar la asignación del personal de los laboratorios a fin de garantizar la continuidad en las actividades de análisis de muestras.
- Cuantificar en términos económicos la producción del laboratorio y mantener indicadores de gestión para los laboratorios.
- Velar por que los equipos se mantengan en sus condiciones óptimas a fin de garantizar la calidad de los análisis.

Un proceso es un conjunto estructurado de actividades diseñado para alcanzar un objetivo establecido. Un proceso de software puede ser definido o diseñado como una jerarquía de procesos a diferentes niveles de abstracción (Montilva y Barrios, 2004). La Fig.3 muestra los procesos primarios que corresponden al nivel 0 de la cadena de valor.

Para el soporte del proceso de gestión de análisis es necesario el manejo de todos los recursos que están involucrados en el mismo, ellos son: manejo del personal, equipos y materiales, los cuales no son únicamente responsabilidad del LIMS, sino que implican la interconexión con otros sistemas.

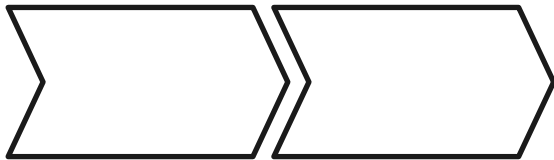


Fig. 3. Cadena de valor de nivel 0

La cadena de valor de nivel 0 se descompone para describir los subprocesos del proceso 1, obteniendo la cadena de valor de nivel 1 mostrada en la Fig. 4.

Esta cadena de valor de nivel 1 lleva implícitos procesos de soporte como son el manejo de: inventario, personal, mantenimiento, calibraciones de equipos y clientes.

Cada uno de los subprocesos de la cadena de valor de nivel 1 se descompone en subprocesos. Un ejemplo de esta descomposición se muestra en la Fig. 5, donde aparece la descomposición del proceso 1.1 que incluye el manejo de los clientes del LIMS en el subproceso 1.1.1, el manejo de todas las ensayos que se realizan en los laboratorios a efecto de seleccionar los que convengan (subproceso 1.1.2), el manejo de las muestras necesarias para cada uno de los ensayos seleccionados (subproceso 1.1.3) y por último, la emisión de las etiquetas correspondientes para cada muestra según el ensayo.



Fig. 4. Cadena de valor de nivel 1



Fig. 5. Cadena de valor de nivel 2

Cada proceso se describe indicando las reglas, actores, objetivos, objetos e informaciones relacionadas directamente con dicho proceso, como se muestra en la Fig. 6.

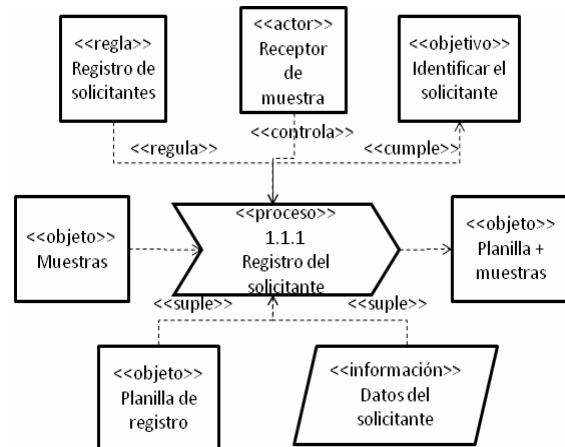


Fig. 6. Modelado del proceso de registro del solicitante

Con el modelado del proceso de negocio, el cual no se incluye aquí por falta de espacio, se obtiene toda la información requerida para modelar la vista estructural del LIMS, que es la base de la ontología y que incluye el modelado conceptual expresado en este caso, a través de diagramas de clases en el lenguaje unificado de modelado UML (Eriksson y col., 2004; Muller, 1997; Rumbaugh y col, 2004; Pilone y Pitman, 2005; OMG, 2004).

Dada la complejidad y falta de descripciones específicas de los componentes de un LIMS, se hace necesario entonces tener una representación por medio de un modelo estandarizado de la definición del dominio, para que los participantes activos o pasivos puedan entenderse entre sí y a dicho sistema.

3 Modelado semántico

El objetivo de este modelado es organizar y estructurar el conocimiento adquirido usando representaciones externas que son independientes de los paradigmas de representación e implementación donde la ontología será posteriormente formalizada. Una visión informal del dominio es convertida en un modelo semi-formal usando representaciones intermedias (conceptos, atributos, relaciones, axiomas y reglas) que son importantes ya que pueden ser entendidos por expertos en el dominio y desarrolladores de ontologías. Por ello, funcionan como puente entre la percepción de las personas y el lenguaje de implementación.

Los términos identificados en el dominio son clasificados en uno o más conceptos taxonómicos, donde un concepto es una abstracción para uno o más términos. Las relaciones entre los conceptos de origen y destino se determinan y se nombran. Se describe cada concepto, relación y atributo.

Para expresar esta descripción se presenta en la Fig. 7 una sección del diagrama de clases en UML para el soporte de las muestras. No se incluye el diagrama de clases UML completo por falta de espacio y por ser el resultado de un trabajo anterior incluido en (Rivero y col., 2008). En ella se observan las clases *Solicitud*, *Muestra* y *Porción*. En particular la clase *Solicitud* soporta el registro de las solicitudes hechas por los clientes, de los cuales también se lleva un registro, pero ésta no aparece en la sección de la figura 7. Estas solicitudes se realizan indicando qué análisis se solicitan y se indican también sobre cuales muestras se deben realizar.

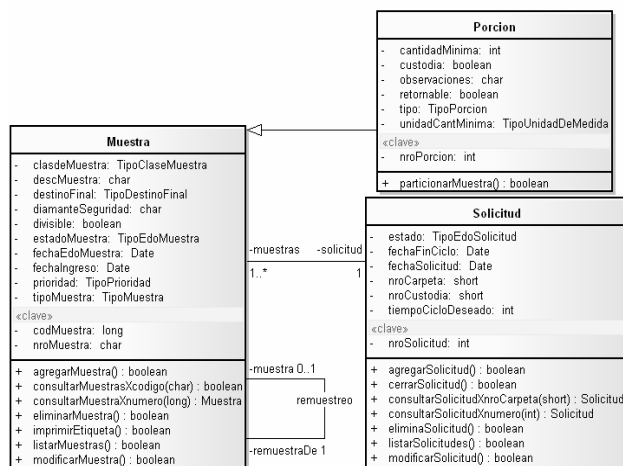


Fig. 7. Sección del diagrama de clases UML

Las muestras recibidas se identifican y si es necesario, se separan en porciones según el número de análisis solicitados. Cada análisis tiene un conjunto de pruebas definidas y cada prueba tiene uno o varios métodos o procedimientos de realización. La selección de la conjunción de método, prueba y análisis se identifica como un ensayo. Los ensayos se realizan para obtener un resultado, lo cual no aparece en

la sección de la figura. Si el resultado obtenido en un ensayo no está aprobado por el cliente, entonces se produce un reclamo, donde cada reclamo se registra en la clase *Reclamo* y si éste es aprobado, es decir el laboratorio lo avala, entonces se registra un re-análisis para el ensayo ya realizado a través de una repetición del mismo.

También se puede dar el caso que una muestra no sea la adecuada y haya necesidad de realizar un re-muestreo. Cada re-muestreo se registra asociado a la muestra original registrada. Las muestras tienen diferentes prioridades, permitiendo así la escogencia del manejo de prioridades en el sistema LIMS. Las porciones tienen una cantidad mínima definida para tomar de la muestra la cantidad necesaria para la realización de cada ensayo, y una unidad de medida para dicha cantidad mínima, que depende del tipo de muestra.

Las solicitudes de análisis de laboratorio se registran con su fecha de llegada y se mantiene un seguimiento del estado de cada una junto con la fecha en que ocurrió el cambio de estado. Así mismo, tanto las muestras como los ensayos y sus reclamos también llevan registro de sus estados asociados a las fechas de sus cambios de estado.

4 OLIMS

En esta sección se describen las actividades realizadas en el desarrollo de OLIMS, lo cual incluye las actividades de especificación que declaran por qué la ontología está siendo construida, el uso previsto y el o los usuarios finales que se prevén. Esta especificación puede ser informal en lenguaje natural o formal. En este sentido OLIMS se construye para compartir el conocimiento en el dominio de los LIMS y favorecer el intercambio de información y significados entre los actores del dominio. Los actores principales son: Analistas, Receptores de muestras, Gerentes, Clientes y sistemas de software que puedan conectarse con los LIMS. Los usuarios finales de OLIMS serán todas aquellas personas que necesiten utilizarla además de los actores ya definidos.

El siguiente grupo de actividades está dedicado a la conceptualización para estructurar el conocimiento del dominio como modelos significativos en el nivel de entendimiento. Normalmente, se construye primero un glosario de los términos que deben ser incluidos en la ontología, su definición en lenguaje natural y sus sinónimos y acrónimos, identificando el núcleo de la terminología. Luego, estos términos se clasifican en uno o más conceptos taxonómicos con sus inter-relaciones determinadas por sus nombres y los conceptos de origen y destino.

A este respecto, los términos utilizados en los LIMS fueron identificados y plasmados en el diagrama de clases UML o modelo conceptual de OLIMS. Con este, se definió el diccionario de conceptos con las relaciones dentro del dominio, sus instancias y sus atributos. Asimismo, se encontraron las expresiones lógicas o restricciones para cada atributo, relación y concepto. Como ejemplo de estas restricciones se tienen:

- Se debe conocer la ubicación y el estado de cada muestra a lo largo de su vida útil en el LIMS.
- Antes de ejecutar un ensayo, se debe verificar la existencia completa del material necesario y la disponibilidad adecuada de equipos e instrumentos, o
- La realización de un ensayo se hace con equipos o instrumentos que no tenga planificado mantenimiento en el lapso del ensayo.

Como parte de la etapa de formalización de Methontology, se transforma el modelo conceptual en un modelo formal. Para cubrir esta actividad, se realizó una búsqueda de las herramientas, extensiones (*plugins*), plataformas y ambientes de software necesarios para la representación de OLIMS en OWL, encontrándose el ambiente y plataforma Eclipse, que permite tener el diagrama de clases en UML 2.0 (OMG, 2004) y su modelo equivalente en XML. Para luego, a través de UML2OWL, hacer la transformación automática del diagrama de clases en UML 2 al lenguaje OWL, o más precisamente, de la representación XML del diagrama de clases UML2 al lenguaje OWL, que también es un lenguaje de marcado. La Fig. 8 muestra parte del contenido del diagrama de clases UML en el ambiente Eclipse.

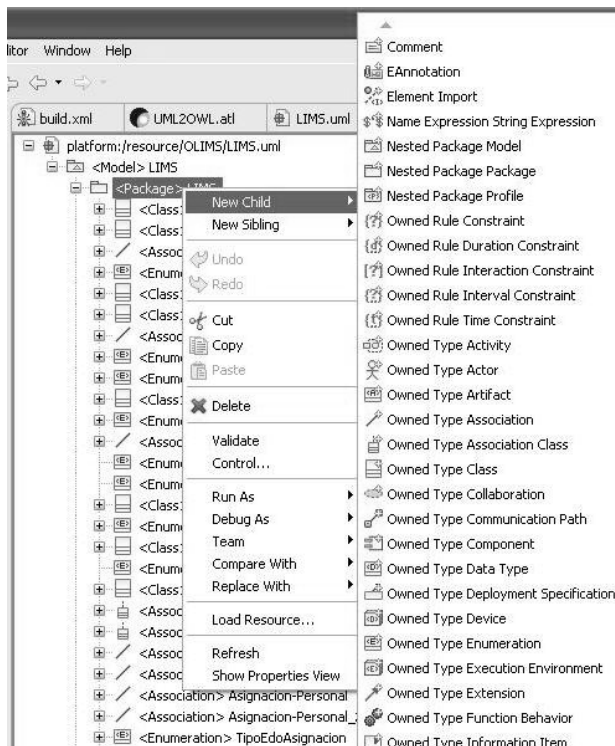


Fig. 8. Modelo conceptual de OLIMS en UML 2.0 bajo eclipse

La descripción equivalente del modelo conceptual en XML para OLIMS se observa en la Fig. 9.

Finalmente, el modelo computable, que puede ser comprendido directamente por la computadora, se construye en la implementación de OLIMS. La implementación del modelo conceptual formalizado puede ser un proceso me-

cánico si la estandarización puede ser traducida automáticamente a un lenguaje de implementación de ontologías.

Entre los lenguajes de implementación para ontologías se encuentra el lenguaje OWL (Ontology Web Language) (OWL, 2004) que ofrece las capacidades de: ser distribuida a través de varios sistemas, es escalable a las necesidades de la Web, es abierto, extensible y es compatible con los estándares Web de accesibilidad e internacionalización.

OWL se divide en tres sub-lenguajes OWL-Lite, OWL-DL y OWL-Full, cada uno de los cuales proporciona un dominio definido de trabajo, siendo el más sencillo OWL-Lite y el más completo OWL-Full.

En el caso OLIMS al tener el modelo conceptual bajo el ambiente Eclipse con su representación en UML y XML, facilitó la transformación en OWL, ya que se utilizó la herramienta UML2OWL (Hillairet, 2004) para la generación automática de OLIMS en OWL-Full.

La Fig. 10 presenta parte del resultado obtenido al utilizar UML2OWL bajo Eclipse. El modelo de datos obtenido se transforma a través del uso de la herramienta para creación de ontologías desde UML denominada UML2OWL, la cual está escrita en el lenguaje ATL (ATL, 2004) y parte del resultado obtenido se presenta en la Fig. 10.

ATL es un lenguaje para la transformación de modelos y por ello, soporta los medios para transformar un modelo fuente en un modelo resultante. ATL también se considera una herramienta para el soporte de dichas transformaciones, bajo la plataforma y ambiente Eclipse.

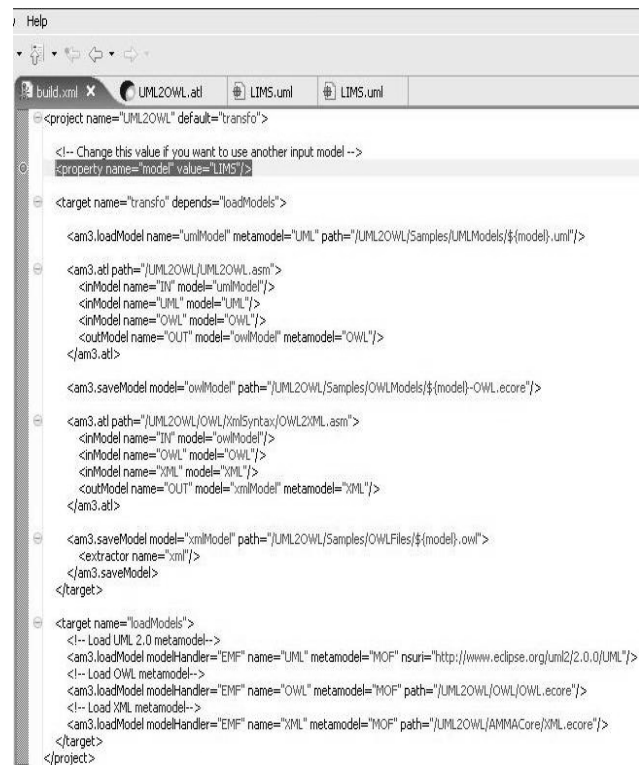


Fig. 9. Sección del modelo conceptual de OLIMS en XML

La Fig. 11 presenta una sección de OLIMS correspondiente a la clase Proyecto, la cual está asociada con la clase Cliente y Solicitante. Esta figura ilustra mejor parte del árbol de visualización de las clases mencionadas, ya que el formato XML es menos ilustrativo.

Para la creación de OLIMS se probaron dos herramientas como son: Protégé y SemanticWorks, siendo esta última la seleccionada, por la posibilidad de recibir el código generado por UML2OWL.

La Fig. 12 presenta una sección de la OLIMS en OWL, la cual utiliza el lenguaje RDF (RDF, 2004) que es el formato universal para la información en la Web semántica.

OLIMS es de fácil extensión y mantenimiento, y por ello hace uso de recursos Web que cumplen con los estándares aprobados por el World Wide Web Consortium (W3C).

Para el desarrollo de OLIMS se utilizó la metodología Methontology, logrando seguir sus etapas hasta alcanzar la implementación de la ontología propuesta.

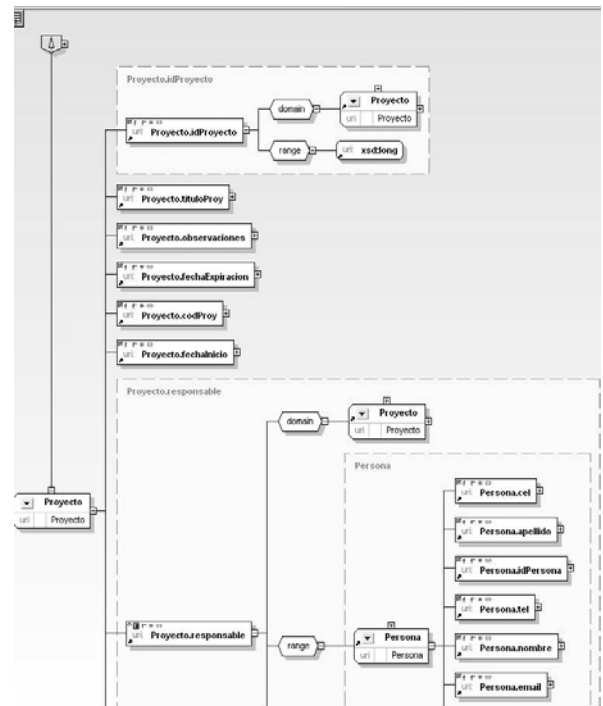


Fig. 11. Sección de OLIMS para la clase proyecto

```

build.xml UML2OWL.atl LIMS.uml LIMS.uml
--@atlcompiler atl2006
module UML2OWL; -- Module Template
create OUT : OWL from IN : UML;

-----
-- UML2OWL
-- Version 1.0
-- Release date : 20/02/2007
-- Author : Guillaume Hillairet (g.hillairet@gmail.com)
-----

helper def: primitiveTypeMap : Map(String, String) =
Map {
('String', 'http://www.w3.org/2001/XMLSchema#string'),
('Integer', 'http://www.w3.org/2001/XMLSchema#integer'),
('Boolean', 'http://www.w3.org/2001/XMLSchema#boolean'),
('UnlimitedNatural', 'http://www.w3.org/2001/XMLSchema#integer'),
('Date', 'http://www.w3.org/2001/XMLSchema#date'),
('Long', 'http://www.w3.org/2001/XMLSchema#long'),
('Float', 'http://www.w3.org/2001/XMLSchema#float'),
('Time', 'http://www.w3.org/2001/XMLSchema#time'),
('UnsignedInt', 'http://www.w3.org/2001/XMLSchema#unsignedInt'),
('Money', 'http://www.w3.org/2001/XMLSchema#money')
};

helper def: addNamespace( s : String ) : String = 'http://sigelab.org/' + s ;

helper context UMLuml::Association def: isBinary() : Boolean =
self.ownedEnd->asSequence().size() = 2;

helper context UMLuml::Property def: isSymmetric() : Boolean =
self.type =
self.association.ownedEnd->asSequence()->any( e |
e.ocIsTypeOf(UMLuml::Property) and not (e = self)).type;

helper context UMLuml::Property def: isInverseFunctional() : Boolean =
self.association.ownedEnd->asSequence()->any( e |
e.ocIsTypeOf(UMLuml::Property) and not (e = self)).upper = 1
and
self.association.ownedEnd->asSequence()->any( e |
e.ocIsTypeOf(UMLuml::Property) and not (e = self)).lower = 1;

helper def: sequenceOfUnionClass : Sequence( UMLuml::Class ) =
let s : Sequence( UMLuml::Class ) =
UMLuml::GeneralizationSet.allInstances()->select( e |
e.ocIsTypeOf(UMLuml::GeneralizationSet) and e.isCovering )
->iterate( acc; res : Sequence(UMLuml::Class) = Sequence{} |
res -> including( acc.powerType )
)
in s;

helper def: getIntegerURI() : OclAny =
thisModule.resolveTemp(
UMLuml::PrimitiveType.allInstances()->any( e |
e.ocIsTypeOf(UMLuml::PrimitiveType) and e.name = 'Integer' ),

```

Fig. 10. Sección de OLIMS resultado de UML2OWL

```

674 <owl:Class rdf:ID="Proyecto">
675 <rdf:label>Proyecto</rdf:label>
676 <rdf:subClassOf>
677 <owl:Restriction>
678 <owl:onProperty rdf:resource="#Proyecto.tituloProy"/>
679 <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</owl:cardinality>
680 </owl:Restriction>
681 </rdf:subClassOf>
682 <rdf:subClassOf>
683 <owl:Restriction>
684 <owl:onProperty rdf:resource="#Proyecto.observaciones"/>
685 <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</owl:cardinality>
686 </owl:Restriction>
687 </rdf:subClassOf>
688 <rdf:subClassOf>
689 <owl:Restriction>
690 <owl:onProperty rdf:resource="#Proyecto.fechaExpiracion"/>
691 <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</owl:cardinality>
692 </owl:Restriction>
693 </rdf:subClassOf>
694 <rdf:subClassOf>
695 <owl:Restriction>
696 <owl:onProperty rdf:resource="#Proyecto.idProyecto"/>
697 <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</owl:cardinality>
698 </owl:Restriction>
699 </rdf:subClassOf>
700 <rdf:subClassOf>
701 <owl:Restriction>
702 <owl:onProperty rdf:resource="#Proyecto.codProy"/>
703 <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</owl:cardinality>
704 </owl:Restriction>
705 </rdf:subClassOf>
706 <rdf:subClassOf>
707 <owl:Restriction>
708 <owl:onProperty rdf:resource="#Proyecto.fechaInicio"/>
709 <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</owl:cardinality>
710 </owl:Restriction>
711 </rdf:subClassOf>
712 <rdf:subClassOf>
713 <owl:Restriction>
714 <owl:onProperty rdf:resource="#Proyecto.responsable"/>
715 <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</owl:cardinality>
716 </owl:Restriction>
717 </rdf:subClassOf>
718 <rdf:subClassOf>
719 <owl:Restriction>
720 <owl:onProperty rdf:resource="#Proyecto.contactos"/>
721 <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">0</owl:minCardinality>
722 </owl:Restriction>
723 </rdf:subClassOf>
724 </owl:Class>
725 <owl:Class rdf:ID="ClienteContacto">
726 <rdf:label>ClienteContacto</rdf:label>
727 <rdf:subClassOf rdf:resource="#Persona"/>
728 </rdf:subClassOf>
729 <owl:Restriction>
730 <owl:onProperty rdf:resource="#ClienteContacto.fechaEdoCIP"/>
731 <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</owl:cardinality>
732 </owl:Restriction>
733 </rdf:subClassOf>

```

Fig. 12. Sección de OLIMS en OWL

Se probó la generación automática de ontologías mediante las transformaciones de modelos y a través de las

herramientas, plataformas y ambientes de software libre encontradas en Web, haciendo uso de los estándares actuales para la implementación de ontologías.

5 Conclusiones

En este trabajo se muestra el proceso de desarrollo de la ontología OLIMS que está embebida en el sistema SIGELAB (Rivero y col., 2008), el cual es un sistema de información para laboratorios de análisis químicos en la industria venezolana.

Agradecimiento

El presente trabajo ha sido realizado en el marco del proyecto Nro. I-1060-07-02-F del Consejo de Desarrollo Científico, Humanístico y Tecnológico de la Universidad de Los Andes (CDCHT-ULA).

Referencias

- Abad M, 2004, Ontologías, Inteligencia Artificial. FBI-Universidad Politécnica de Cataluña, España, ATL, 2004, ATLAS Transformation Language Project. Eclipse Foundation. <http://www.eclipse.org>.
- Berners-Lee T y Col, 2001, The Semantic Web. Scientific American, Vol. 5.
- Chandrasekaran B y Josephson J, 1999, What are ontologies, and why do we need them, IEEE Intelligent Systems, Ene-Feb, pp. 20-26.
- Davies J, Studer R y Warren P, 2006, Semantic Web Technologies: Trends and Research in Ontology Based Systems. Chichester: John Wiley & Sons Ltd.
- Davies J, Fensel D y Van Harmelen F, 2003, Towards the Semantic Web: Ontology Driven Knowledge Management. Chichester: John Wiley & Sons Ltd.
- Eriksson HE, Penker M, Lyons R y Fado D, 2004, UML 2 Toolkit. Wiley Publishing, Indiana.
- Gómez A, Fernández M y Corcho O, 2004, Ontological Engineering, Springer-Verlag.
- Hillairet G, 2004, UML2OWL Programa de software libre.
- HTML, 1990, Hypertext Markup Language, WWW Consortium. <http://www.w3.org/TR/html>.
- Montilva J y Barrios J, 2004, The WATCH Method for Developing Web Applications. En Montilva J.
- Besembel I, Pérez M & Losavio F (Eds.). Sistemas de Información e Ingeniería de Software: Temas Selectos. (pp. 328-343). Mérida, Venezuela: Centro de Estudios en Informática.
- Muller JC, 1997, Modelado de objetos con UML. Eyrolles-Gestión 2000, Barcelona, España.
- Nirenburg S y Raskin V, 2004, Ontological Semantics. The MIT Press.
- Patel P, 2006, A comparison of two modeling paradigms in the semantic web, International World Wide Web Conference (IW3C2).
- RDF, 2004, Resource Description Framework. W3C, Hayes, P. (ed). <http://www.w3.org/TR/rdf-nt>
- Rivero D, Besembel I, Hidrobo F, Díaz G, Narciso F y Hernández D, 2008, Análisis, Diseño e Implantación de Aplicación LIMS para la Automatización de la Gestión de los Laboratorios de INTEVEP. Universidad de Los Andes, Escuela de Ingeniería de Sistemas.
- Rumbaugh J, Jacobson I y Booch G, 2004, Unified Modeling Language Reference Manual. Second Edition. Addison Wesley.
- OMG, 2004, Object Management Group, UML 2.0 Superstructure Specification. <http://www.omg.org>
- OWL, 2004, Web Ontology Language, W3C, McGuinness D y Van Harmelen F, (eds). <http://www.w3.org/TR/owl-features>.
- Schultz, DJ, y col., 1997, IEEE Standard for Developing Software Life Cycle Processes. The Institute of Electrical and Electronics Engineers, New York, USA, IEEE Std, No. 1074-1997.
- Pilone y Pitman, 2005, UML 2.0 in a Nutshell. O'Reilly.