Ciencias de la Información Vol. 46, No. 1, enero-abril, pp. 19 - 25, 2015

ViewOnto: modelo conceptual para la generación automática de vistas de datos

Félix Fernández Peña
Rolando Acosta Sánchez
Yudit Ponce Toste

En este trabajo se propone un modelo de datos conceptual basado en ontologías para representar vistas de datos. El objetivo esencial es evitar la recodificación de los sistemas de información en caso de que surjan cambios en la interpretación del negocio o que el cliente introduzca nuevos requisitos en la gestión de la información. Para la experimentación fue creada una herramienta prototipo capaz de generar la interfaz de usuario a partir de la ontología propuesta. Los resultados obtenidos evidencian la factibilidad de la propuesta para el caso de bases de datos relacionales. La propuesta es un primer resultado de un trabajo de mayor envergadura donde se pretende describir fuentes de datos abiertas.

Palabras clave: sistemas de información, vista de dato, ontología, bases de datos relacionales.

RESUMEN

ABSTRACT

This paper introduces a conceptual data model for representing data views using ontologies. The main objective is to avoid the recodification of information systems when business logic changes appear. The experiments carried out are supported by a prototype of data retrieval tool capable of the dynamic generation of user interfaces based on the proposed ontology. As result of the experimentation, the practicability of the proposal has been proved. Results shown in here are a first step in a semantic-approach for the integration of open data sources.

Keywords: information systems, data view, ontology, relational databases.

Introducción

El entorno corporativo es cada día más dependiente de la precisión y la velocidad con que se efectúa la recuperación de la información (Lin, 2014). Información fidedigna, actualizada y de fácil compresión, con el consiguiente conocimiento por parte de los expertos para analizarla, son probablemente hoy en día los recursos más importantes en estos escenarios (Lin, 2014) (Niemi, 2007).

Las operaciones de recuperación de información tienen un papel escencial en los sistemas informáticos corporativos al actuar como la interface entre la información y los datos almacenados. Un sistema de recuperación de información debe ser flexible, siendo capaz de adaptarse a posibles cambios en la lógica de funcionamiento de la organización y en general al ciclo de vida de la información (Niemi, 2007). Nuevos datos producto de las operaciones de las empresas, nuevos análisis, y cambios en la forma de entender el funcionamiento de las organizaciones son tan comunes y naturalmente aceptados como lo es la fiabilidad del conocimiento (Lin, 2014). Sin embargo, adaptar los sistemas informáticos a ambientes cambiantes generalmente requiere mucho esfuerzo manual (Niemi, 2007). Casi siempre se requiere una recodificación de los sistemas de información tradicionales cuando cambia la interpretación de los datos (Niemi, 2007).

La presente investigación propone reducir el costo de mantenimiento e incrementar la flexibilidad de los sistemas de información

haciendo explícita la semántica de la Base de Datos Relacional (RDB por sus siglas en inglés).El Lenguaje de Descripción de Recursos (RDF por sus siglas en inglés) hace posible definir el significado (semántica) de los datos en un formato que puede ser interpretado por las máquinas (Motik, 2009). Tomando en cuenta este planteamiento, es posible emplear las tecnologías de la Web Semántica para el manejo de la dimensión semántica de las RDBs e incrementar así la flexibilidad y escalabilidad de los sistemas de recuperación de información.

Numerosas investigaciones y propuestas han sido realizadas en aras del enriquecimiento semántico de los datos almacenados en una RDB (Chebotko, 2010)(Sun, 2009). La evolución de RDF en un Lenguaje para la Descripción de Ontologías (OWL por sus siglas en inglés) permite una descripción semántica mayor, basada en Lógica Descriptiva (DL por sus siglas en inglés) (Horrocks, 2003). OWL es un lenguaje formal para la representación de ontologías en la Web Semántica (Horrocks, 2003). Este lenguaje ha sido empleado en diversos escenarios para la construcción de modelos de datos semánticos de mayor flexibilidad (Agus, 2011) (Èeráns, 2010) (Munir, 2012).

El interés por relacionar datos de fuentes relacionales a documentos RDF se ha incrementado con el propósito de publicar datos relacionados (Sudairy, 2011). SPARQL (acrónimo recursivo del inglés SPARQL Protocol and RDF Query Language) es el lenguaje de consulta para la recuperación de información a partir de documentos RDF recomendado por el World Wide Web Consortium (W3C, 2014). En el entorno de la semántica, SPARQL es considerado el equivalente teórico al Lenguaje de Consulta Estándar para Bases de Datos (SQL por sus siglas en inglés). Sin embargo, de forma similar a (Hert, 2010), los autores consideran que convertir datos relacionales en RDF es una tarea con frecuencia no factible. Por otra parte, varios estudios han demostrado que ejecutar la consulta SPARQL sobre el archivo RDF correspondiente a una RDB resulta en una respuesta del sistema más tardía que la ejecución directa de la consulta SQL correspondiente (Antoniou, 2012).

Los autores proponen no transformar las bases de datos en RDF sino en su lugar definir un mecanismo de recuperación de datos guiado por ontologías. La idea esencial es describir las fuentes de datos usando una ontología Web.En el diseño propuesto se formaliza la estructura de Vistas de Datos (***V***) de RDB genéricas en base a la descripción de la relación de estas con el conjunto de columnas de datos (***S***). El objetivo es definir un modelo conceptual, no en la dimensión sintáctica sino en la semántica.

De esta forma si cambiase el dominio de discurso de la aplicación, no sería necesario recodificar los métodos de recuperación de la información sino que en su lugar se transformaría únicamente la representación semántica de los datos a recuperar. Como beneficios: es posible hacer una generación automática de las vistas de datos (***V***), y no menos valioso: los datos se muestran al usuario final de acuerdo con la relevancia de la información en un momento determinado y acorde a la semántica definida. Para evaluar las capacidades de presentación y navegación utilizando el método propuesto se construyó una aplicación prototipo. Un estudio comparativo con las implementaciones tradicionales sobre Sistemas de Administración de Bases de Datos Relacionales (RDBMS por sus siglas en inglés) muestra que el método puede ser beneficioso en la recuperación de información cundo se ha hecho una adecuada descripción semántica de la RDB.

Materiales y métodos

La formalización de la propuesta está sustentada en el basamento matemático de la lógica descriptiva, por lo que se introduce brevemente en el epígrafe siguiente. A partir de esta lógica se formalizó el modelo propuesto y se diseñó un experimento para su evaluación en el ámbito de los sistemas de información que utilizan bases de datos relacionales. El caso de estudio utilizado para la experimentación ha sido referenciado en la bibliografía en estudios de sistemas de información.

Lógica descriptiva

La Lógica Descriptiva (DL por sus siglas en inglés) es definida como una familia de formalismos para la representación del conocimiento. Conceptos relevantes para un dominio específico son definidos mediante la declaración de los objetos del dominio, sus propiedades y sus instancias (Baader, 2010).

Las estructuras básicas en un lenguaje conceptual son los conceptos y los roles. Un concepto es la descripción de las propiedades que son comunes a un grupo de individuos y desde el punto de vista de DL representa un predicado unario. Las relaciones entre esos individuos son representadas mediante roles (articulados como relaciones binarias). Generalizando esta formalización, podemos decir que una base de conocimientos queda definida como un conjunto finito de axiomas sobre los conceptos y roles (nombrado TBox) y un conjunto de axiomas sobre sus instancias (conocido como ABox).

Visto desde el punto de vista funcional, un sistema basado en DL está caracterizado por cuatro aspectos fundamentalmente (Baader, 2010): un conjunto de estructuras que constituyen el lenguaje empleado para describir los conceptos y roles, las declaraciones contenidas en el TBox, las declaraciones contenidas en el ABox y un mecanismo de inferencia para el razonamiento a partir de las bases de conocimiento.

OWL fue concebida de forma que es capaz de codificar esquemas de bases de datos y expresarlos en Modelos de Datos Semánticos (Horrocks, 2003)(Song, 2013). OWL es empleado como lenguaje de representación básico de la terminología que se propone en el presente artículo.

Modelo de Datos Propuesto

ViewOnto está descrito en tres niveles de abstracción: En su nivel más abstracto está definido como la terminología que permite describir vistas de bases de datos relacionales genéricas (en lo que constituye el TBox de la ontología de dominio). Este nivel describe conceptos relevantes para la recuperación de información desde cualquier RDB. En la capa intermedia se instancia el esquema abstracto para dominios de aplicación específicos (llegando a construir el ABox de la ontología relativo a una única RDB). Finalmente, en la capa de abstracción más baja, los datos son recuperados mediante consultas SQL especificadas en el ABox que describe cada RDB que se utiliza.

El procedimiento que se propone para la

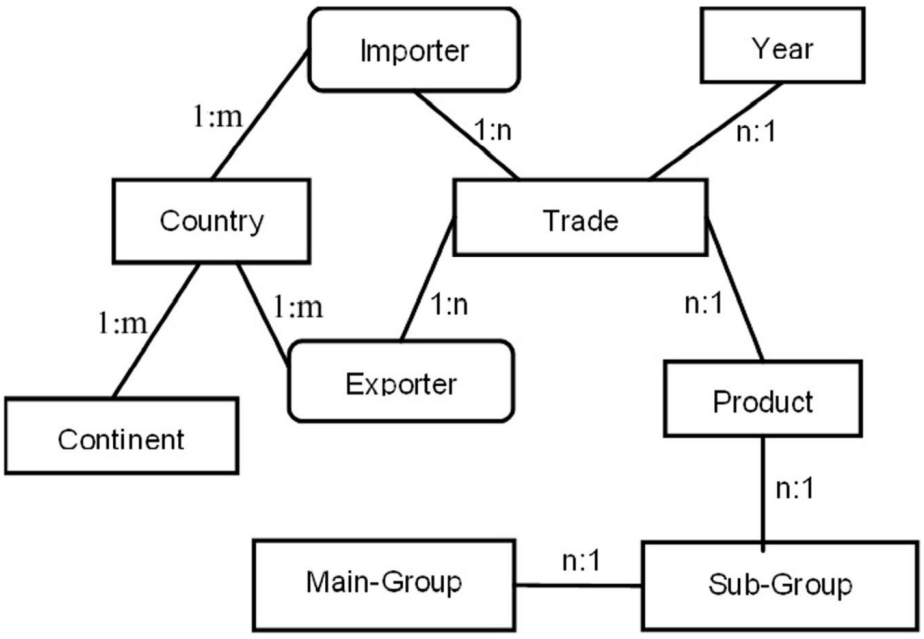

Figura 1. ***Modelo Relacional sintáctico empleado como caso de estudio.***

recuperación de información debe seguir los siguientes pasos: Se construye una definición conceptual de cada Vista de Datos instanciando los conceptos de *ViewOnto TBox* para el dominio de aplicación específico. Como resultado, se obtiene el *ViewOnto ABox* del dominio de aplicación dado. La semántica de las vistas de datos queda definida formalmente en el *ABox*. Posteriormente se relacionan los datos recuperables mediante consultas SQL al RDBMS con los conceptos del *ABox*. Finalmente, una aplicación genérica construye la interfaz de usuario apropiada al usuario final en correspondencia con la semántica definida y las capacidades de recuperación del lenguaje SQL utilizado.

Para ilustrar el método se emplean datos de un caso de estudio que versa sobre comercio internacional. Para una mejor comprensión por parte del lector se expone en la figura 1 el modelo relacional (sintáctico) del caso de estudio. La figura 2 muestra la interacción de los tres niveles de abstracción definidos para el modelo *ViewOnto* en el dominio específico (simplificado) del comercio de productos ilustrado en la figura 1.

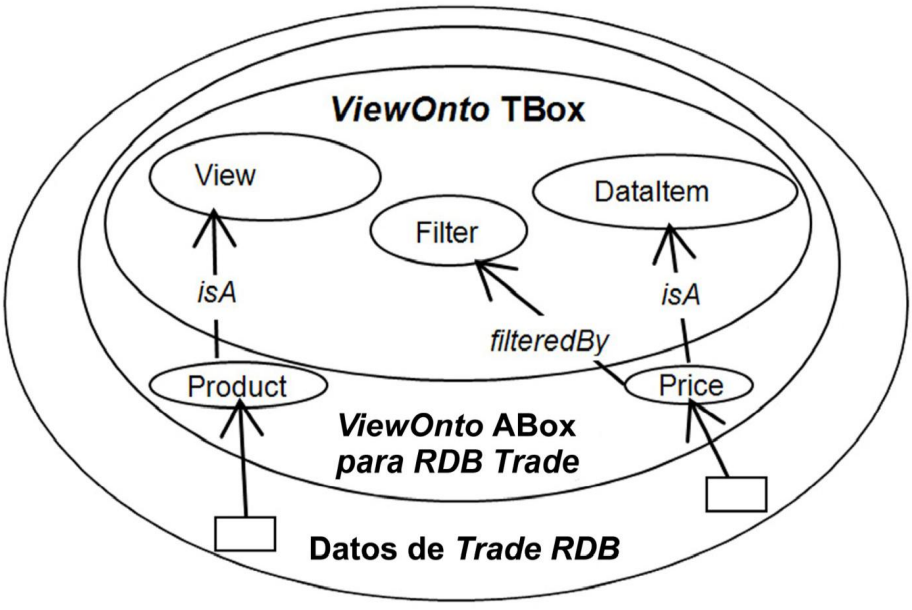

Figura 2. ViewOnto *vista en sus tres niveles de abstracción para el caso del comercio de productos.*

Codificación de la semántica de una vista de datos de una BDR

En la propuesta se emplea OWL para la codificación de la semántica. A continuación se define la forma en que una vista de una RDB se especifica en la lógica de OWL para el modelo propuesto.

ViewOnto TBox

Los conceptos fundamentales de la terminología propuesta son: Vista (View) V, Identificador (Identifier) I, Descriptor (Descriptor) D, Elemento de Dato (DataItem) S, Relación (Relationship) R y Filtro (Filter) F. En la definición se utilizan los términos en idioma inglés para ganar en generalidad y simplicidad de representación en la propuesta.

Una *Vista* se define como la vista tabular de los elementos de información que describen determinado concepto de relevancia en el dominio del conocimiento que se modela. El concepto extiende el concepto *Vista* del modelo relacional con la representación de un conjunto de propiedades que lo describen en lenguaje natural y un conjunto de relaciones semánticas que se establecen con los elementos de información que lo conforman (identificadores y descriptores) y otras vistas del modelo (vistas relacionadas). Otros conceptos relacionados son:

Identificador:definidocomo cada uno de los elementos de datos que identifican a una vista.

Descriptor: Cada uno de los elementos de datos que no identifican a una vista pero sí forman parte de los elementos de datos de esta constituye un *descriptor*.

Elemento de Dato: Homólogo al campo de dato en el modelo relacional, cada *elemento de dato* está asociado a un campo o función de agrupación del modelo relacional. Se han definido cinco *elementos de datos* específicos; estos son: *Elemento de Dato Booleano (BooleanDataItem), Elemento de Dato Fecha (DateDataItem), Elemento de Dato Imagen (ImageDataItem), Elemento de Dato Numérico (NumberDataItem)* y *Elemento de Dato Cadena de Caracteres (StringDataItem).*

Relación: Vínculo que existe entre dos

vistas del modelo o entre dos elementos de dato de una vista. Los elementos de dato relacionados pueden ser, descriptores o identificadores, indistintamente. En dependencia del tipo de relación esta es nombrada: Relación entre Elementos de Datos(*InterDataItemRelationship*) o Relación entre Vistas (*InterView Relationship*).

Filtro: Operación producto de la cual se obtiene un subconjunto del conjunto de tuplas asociadas a una vista dada. Existe un conjunto de filtros aplicable en dependencia del tipo de dato que se maneja asociado a cada elemento de dato. Así, se han definido filtros como *lessThan*, *greaterThan* y *equalTo*, válidos para Elementos de Datos Numéricos mientras que *startsWith*, *contains y endsWith* son filtros válidos para Elementos de Datos Cadena de Caracteres**.**

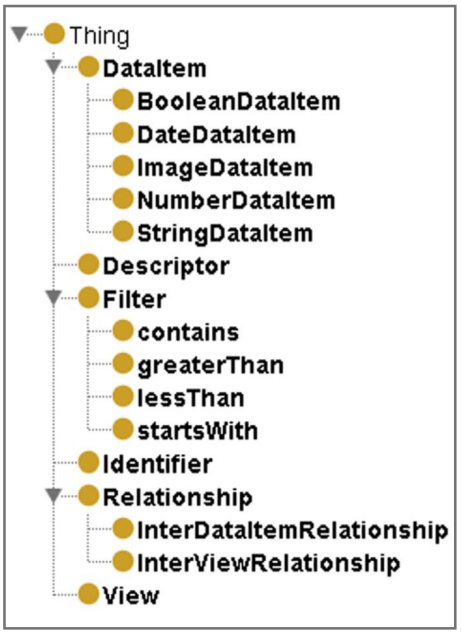

Figura 3. Jerarquía de Conceptos definidos en ViewOnto.

La declaración explícita de filtros permite personalizar la interfaz del usuario (que limita el conjunto de filtros aplicable a un subconjunto prediseñado teniendo en cuenta las características propias de la vista en cuestión). Si el conjunto de filtros a aplicar no se declara explícitamente entonces se tiene en cuenta el conjunto de filtros aplicable según el tipo de dato del elemento de dato de que se trate.La figura 3 muestra una vista gráfica de estos conceptos teniendo en cuenta la relación jerárquica existente.

A continuación se definen las propiedades que caracterizan a los conceptos definidos. Su aplicabilidad a los conceptos definidos se muestra en la tabla 1.

title: constituye una expresión en lenguaje natural que define el concepto al que se aplica.

mappedTo: sentencia en lenguaje SQL que permite la recuperación de los datos asociados a determinada vista de datos.

*isAlternative:*define cuándo determinado identificador o descriptor es considerado fundamental en la interpretación de una vista. Toma valores verdadero o falso y permite dotar de versatilidad a la interfaz de aplicación.

*fieldName:*relaciona cada elemento de dato con determinado nombre de campo o función de agregación.

sqlCond: es condición programática para filtrar las tuplas asociadas a una vista dada por un criterio dado.

*clientFiltering:*especifica si determinado filtro es aplicado en la propia interfaz de usuario o si se requiere de una operación en el servidor de fuente de datos; toma valores verdadero o falso.

Además de estas propiedades, se definen otras que hacen explícitas las relaciones semánticas existentes entre los conceptos definidos. Estas se describen a continuación y en la figura 4 se representa la relación jerárquica existente entre ellas.

isIdentifiedBy (View, Identifier): Una *Vista*es *identificada por (isIdentifiedBy)* un conjunto de identificadores.

isDescribedBy (View, Descriptor): Una *Vista* es *descrita por (isDescribedBy)* un conjunto de descriptores.

definedBy (Identifier, DataItem): Un *Identificador* es definido por un *Elemento de Dato*.

describedBy (Descriptor, DataItem): Un *Descriptor* es descrito por un *Elemento de Dato*.

isFilteredBy (DataItem, Filter): Un Elemento de Dato es filtrado por un Filtro.

dataRelatedBy ({Descriptor, Identifier} ,InterDataItemRelationship): Un Descriptor o Identificador está relacionado a través de una Relación entre Elementos de Dato.

viewRelatedBy (View, InterViewRelationship): Una Vista está relacionada a través de una Relación entre Vistas.

dataRelatedTo (InterDataItemRelationship, {Descriptor, Identifier}): Una relación entre elementos de datos está relacionada con un descriptor o identificador del modelo.

viewRelatedTo (InterViewRelationship, View):Una relación entre vistas está relacionada con una vista del modelo.

Tabla 1: Propiedades aplicadas a cada uno de los conceptos definidos en *ViewOnto*

Concepto	title	mappedTo	isAlternative	fieldName	sqlCond	clientFiltering
View	X	X				
Identifier	X		X			
Descriptor	X		X			
DataItem	X			X		
Relationshi	X					
Filter	X				X	X

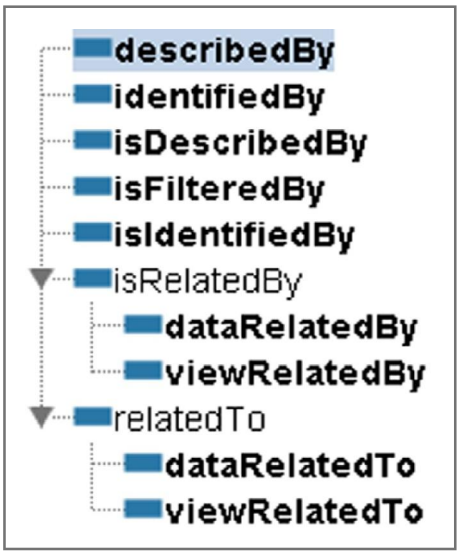

Figura 4. Jerarquía de Propiedades en ViewOnto

ViewOnto ABox para el caso de estudio Comercio de Productos

Hasta este momento se ha mostrado una formalización abstracta del modelo conceptual de datos. La terminología semántica de una vista de datos de una RDB específica se describe instanciando *ViewOnto*para dicho escenario. Para el caso de estudio en cuestión implicó la definición de:

- las vistas TRADE y EXPORTER,
- identificadores como TRADEID y
- descriptores como TRADEVALUE, TRADECOUNT, IMPORTERNAME y EXPORTERNAME,

además de :

- la relación entre vistas T_E y
- la relación entre elementos de datos TRADEVALUE y TRADECOUNT con identificador TV_TC.

Haciendo uso de esta instanciación en el *ABox* es posible que un proceso automático de inferencia determine, por ejemplo, que la vista TRADE es identificada por el *Identificador* TRADEID, TRADE es descrita por los *Descriptores* IMPORTERNAME, EXPOERTERNAME, TRADECOUNT y TRADEVALUE. Del mismo modo, TRADEVALUE es considerada escencial dado que se asocie al rol *isAlternative* con valor falso. Al analizar la vista de datos TRADE, la vista de datos EXPORTER se resalta como posible contenedora de información relevante puesto que existe una relación T_E que las vincula. En este mismo sentido la correlación entre los *Descriptores* TRADEVALUE y TRADECOUNT es sobreentendida como significante para la toma de decisiones dada la relación TV_TC.

De forma general, una *Vista* puede estar relacionada con una o más tablas en la RDB y viceversa. Por ejemplo: el nombre de los exportadores es considerado un *Descriptor*en la vista TRADE (varias tablas relacionadas a una vista). Por otra parte la tabla *exporter* está relacionada con la vista EXPORTER también (una tabla relacionada con varias vistas). Cada vista de datos se transforma en una instancia única de *ViewOnto*. Sin embargo, esta instancia es determinada en la capa sintáctica combinando campos, funciones de agregación y campos calculados provenientes de una o varias tablas mediante sentencias SQL. Por ejemplo: los *Descriptores*TRADEVALUEDy TRADECOUNTpueden ser calculados empleando sentencias SQL *sum()* y *count()* respectivamente.

Si la información requerida por el usuario final cambia. La nueva definición puede ser fácilmente hecha cambiando el fichero donde está definida la OWL. No son necesarios cambios en el código de la aplicación sino cambios declarativos en el modelo de datos (*OntoView ABox).*

Herramienta prototipo

Para evaluar la factibilidad del modelo propuesto se construyó una aplicación prototipo para el caso de estudio que se ha venido presentando. La implementación básica sigue los siguientes pasos:

1. Un *ABox* que especifica la ontología *ViewOnto* para el caso de estudio se carga en la aplicación prototipo.

2. El valor del rol *title* para cada instancia *View* en el *ABox* es mostrado como un punto de acceso a la información en la interface de usuario final.

3. El role *mappedTo* se emplea en el proceso de recuperación de información para la obtención de los datos a partir de la RDB. La interfaz de usuario se reajusta y expone la información.

4. El usuario emplea las vistas relacionadas (role *viewRelatedTo*), datos relacionados (role *dataRelatedTo*) y los filtros definidos (role *isFilteredBy*) para enriquecer el espectro de búsqueda del usuario.

El proceso de recuperación de información se hace mediante una arquitectura en tres capas. El usuario final interactúa con las «Vistas de Datos» ubicadas en la «Capa Interfaz de Usuario». Estas vistas son conformadas mediante el proceso de recuperación que se encuentra en la capa denominada «Capa de Recuperación de Datos». El modelo semántico específico para el dominio de la aplicación, así como la base de datos relacional están en la capa de persistencia «Capa de Persistencia de Datos».

La figura 5 muestra la arquitectura de la herramienta propuesta para la recuperación de información mediante el empleo del modelo propuesto.La figura 6 muestra como la vista descrita en la ontología se muestra al usuario final mediante la interfaz de aplicación de la herramienta prototipo desarrollado.

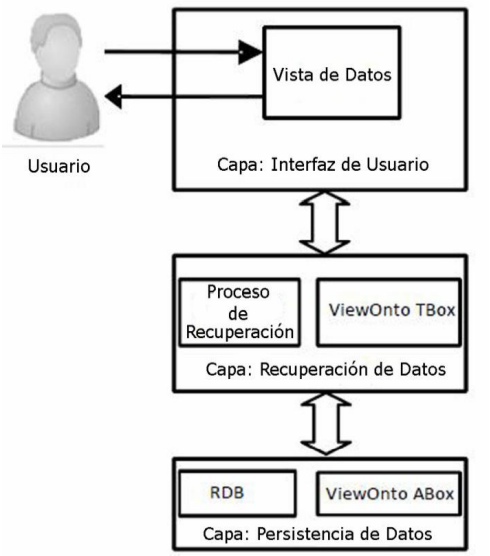

Figura 5. ***Arquitectura en capas del proceso de recuperación de información mediante el modelo propuesto.***

OFFICIAL TRADE			1
2 *Imported by*		*Exported by*	*Amount*
Copextel	3	Intel	10400
Desoft		Haier	62000

Figura 6. Vista de usuario del concepto «TRADE»

En la figura 6 «Official Trade» corresponde al rol *title* de la *Vista «*TRADE». Los encabezados de la tabla («Imported by», «Exported by» y «Amount») se corresponden

con el rol *tilte* de cada *Descriptor (D)* donde el rol *isAlternative* toma valor «false». Cada fila en la tabla se corresponde con los valores de cada *DataItem (S)* en la *Vista (V)* «TRADE».

La recuperación de información es intuitiva en esta herramienta. Cuando un usuario está sobre el valor de un DataItem (S) y abre el menú de contexto aparecen las opciones que se muestran en la Figura 7. El usuario podrá entonces solicitar datos relacionados y filtrarlos. La Figura 7 muestra los pasos para navegar entre los datos (de la Vista TRADE a EXPORTER y de esta a PRODUCT). Se puede observar cómo toda la navegación está definida de forma declarativa en el ABoxinstanciado para el caso de estudio.La figura 8 muestra cómo el usuario puede aplicar los filtros definidos en la ontología.

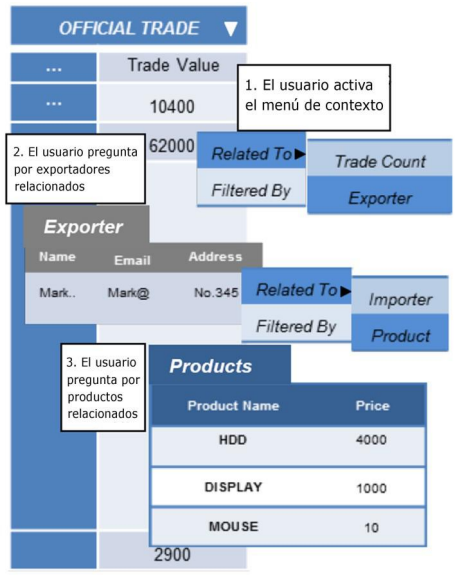

Figura 7. Muestra de navegación semántica en la recuperación de datos para el caso de estudio.

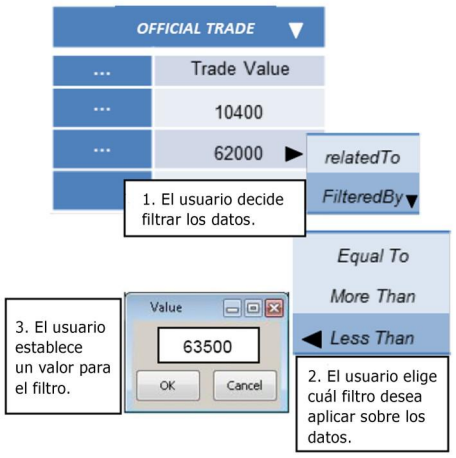

Figura 8. Muestra de la aplicación de filtros sobre los datos.

Resultados y discusión

Fue diseñado un experimento en aras de comparar la recuperación de información a través del método propuesto con los métodos tradicionales de los RDBMS. Un grupo de diez estudiantes de Ingeniería Informática participó en la evaluación de la propuesta. Cada estudiante recibió diez operaciones de recuperación de información que debían realizar (las mismas diez para cada persona) con complejidad creciente en la sentencia SQL requerida. Las últimas cuatro tareas demandaban cambios en la interpretación semántica de los datos.

Todos los estudiantes tenían experiencia en el diseño e implementación de RDBMS en *Java*. Se asignó a cada estudiante un ordenador con un procesador i3, 2Gb de memoria RAM y suficiente espacio en disco duro. Cada estudiante recibió el código de un RDBMS implementado en *Java* y la herramienta prototipo basada en la ontología *ViewOnto* (ambos trabajando con la RDB *Trade*). A cinco estudiantes se les pidió completar tareas que requerían hacer modificaciones en el RDBMS y luego se les pidió realizar las mismas tareas empleando la herramienta prototipo propuesta. A los otros cinco estudiantes se les pidió lo mismo pero en orden inverso (primero el prototipo y luego el RDBMS). El tiempo se registró para cada tarea por cada estudiante.

Fue requisito para los estudiantes que las vistas de datos resultantes al aplicar tanto un método como el otro recuperaran los mismos datos. Este experimento preliminar mostró que la propuesta incrementa la flexibilidad y decrementa el costo de mantenimiento del sistema de información cuando está definida una semántica que describe de forma correcta la RDB. En el 92% de los casos los estudiantes obtuvieron la solución de forma más rápida con el método propuesto. La tabla 2 muestra el valor medio del tiempo (en minutos) medido para cada tarea. Por otra parte, el conocimiento requerido para adaptar el prototipo a datos cambiantes mediante el prototipo, resulta irrelevante en comparación con el necesario para hacerlo en *Java*.

Tabla 2: Resultados experimentales

Tarea	ViewOnto(*)	RDBMS(*)
T1	1.52	3.83
T2	1.41	3.82
T3	1.35	3.79
T4	1.42	4.02
T5	1.36	3.09
T6	1.35	3.88
T7	1.67	3.72
T8	1.65	3.69
T9	1.62	6.08
T10	1.42	3.82

*** Los valores están medidos en minutos.**

Conclusiones

Como resultado del trabajo realizado se obtuvo un modelo conceptual para la generación automática de vistas de datos, al cual se le denominó ViewOnto. Su construcción, unida a la experimentación llevada a cabo ha permitido corroborar que la descripción semántica de una fuente de datos mediante ontologías se puede emplear para evitar la recodificación de la aplicación al cambiar los requisitos de información de este. Del mismo modo, la descripción semántica de los datos provee una experiencia de usuario final más rica durante la recuperación de información. En sentido general, los experimentos realizados evidenciaron la efectividad del modelo.

Actualmente se trabaja en la extensión del modelo para que permita no solo la recuperación de información de fuentes de datos relacionales sino que además la recuperación pueda hacerse a través de servicios Web. Esto deberá permitir la recuperación de información de fuentes de datos con características diferentes a las relacionales.

Agradecimientos

Los autores desean agradecer en especial al Dr. Cornelio Yañez, del Centro de Investigación en Computación del Instituto Politécnico Nacional (CIC-IPN) Ciudad de México y al Dr. Jyrki Nummenmaa, de la universidad de Tampere, Finlandia. La colaboración de estas dos personas ha resultado vital para el desarrollo de la investigación aquí presentada. Del mismo modo, se agradece a todos los que, de una forma u otra, han colaborado y, por motivos de espacio no serán mencionados, entre ellos, profesores, investigadores y estudiantes dela Cujae que, del alguna forma, han estado vinculados al desarrollo

de éste proyecto. Reciban todos, nuestro reconocimiento y agradecimiento.

Bibliografía

Agus-Santoso, H., Cheng-Haw, S., Abdul-Mehdi, Z. (2011). Ontology extraction from relational database: Concept hierarchy as background knowledge. Knowledge-Based Systems, 24(3), 457 - 464.

Antoniou, G., Corcho, O., Aberer, K. (2012). Semantic Data Management. Trabajo presentado en Dagstuhl Seminar.

Baader, F., Calvanese, D., McGuinness, D. (2010). The Description Logic Handbook. Theory, Implementation and Application. 2da Edición. Cambridge University Press.

Èeráns, K., Búmans, G. (2010). RDB2OWL: a RDB-to-RDF/ OWL Mapping Specification Language. Trabajo presentado en Conference on Databases and Information Systems.

Chebotko, A., Lub S., Fei, X., Fotouhi, F. (2010). RDFProv: A relational RDF store for querying and managing scientific workflow provenance. Data & Knowledge Engineering, 69(8), 836 - 865.

Hert, M., Reif, G., Gall, H. (2010). Updating relational data via SPARQL/update. Trabajo presentado en EDBT/ICDT Workshops.

Horrocks, I., Patel-Scheneider, P. F., van Harmelen, F. (2003). From SHIQ and RDF to OWL: The Making of a Web Ontology Language. Web Semantics, 1(1), 7 - 26.

Lin, Y., Cole, Ch., Dalkir, K. (2014). The relationship between perceived value and information source use during KM strategic decision-making: A study of 17 Chinese business managers. Information Processing & Management, 50(3), 156 - 174.

Motik, B., Horrocks, I., Sattler, U. (2009). Bridging the gap between OWL and relational databases. Web Semantics, 2(2), 74 - 89.

Munir, K., Odeh, M., McClatchey, R. (2012). Ontology-driven relational query formulation using the semantic and assertional capabilities of OWL-DL. Knowledge-Based Systems, 35, 144 - 159.

Niemi, T., Toivonen, S., Ninimäki, M. Nummenmaa, J. (2007). Ontologies with Semantic Web/grid in data integration for OLAP. International Journal on Semantic Web and Information Systems, Special Issue on Semantic Web and Data Warehousing, 3(4).

Sun, H., Fan, Y. (2009). Semantic Extraction for Multi-Enterprise Business Collaboration. Tsinghua Science & Technology, 14(2), 196 - 205.

Sudairy, M., Vasista, T. (2011). Semantic data integration approaches for e-governances. Web & Semantic Technology, 2(1).

W3C. (2014). SPARQL. Consultado 23 de marzo de 2014,disponible http://www.w3.org/TR/ sparql11-overview.

Song, F., Zacharewicz, G., Chen, D. (2013). An ontology-driven framework towards building enterprise semantic information layer. Advanced Engineering Informatics, 27(1), 38 - 50.

Recibido: 21 de agosto de 2014.
Aprobado en su forma definitiva:
2 de febrero de 2015

Félix Fernández Peña
Instituto Superior Politécnico José Antonio Echeverría (CUJAE), La Habana, Cuba.
Correo-e.: felix@ceis.cujae.edu.cu

Rolando Acosta Sánchez
Instituto Superior Politécnico José Antonio Echeverría (CUJAE), La Habana, Cuba.
Correo-e.: racostas@ceis.cujae.edu.cu

Yudit Ponce Toste
Instituto Superior Politécnico José Antonio Echeverría (CUJAE), La Habana, Cuba.
Correo-e.:yudit@ceis.cujae.edu.cu