Polibits

ISSN: 1870-9044

polibits@nlp.cic.ipn.mx

Instituto Politécnico Nacional

México

Velázquez Ordoñez, Benina; Olivares Ceja, Jesús Manuel; Patiño Ortíz, Miguel; Patiño Ortíz, Julián; Guzmán Arenas, Adolfo

Integración de fuentes heterogéneas de datos textuales

Polibits, vol. 51, enero-junio, 2015, pp. 19-25

Instituto Politécnico Nacional

Distrito Federal, México

Disponible en: http://www.redalyc.org/articulo.oa?id=402641203004

Cómo citar el artículo

Número completo

Más información del artículo

Página de la revista en redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

ISSN 2395-8618

Integración de fuentes heterogéneas de datos textuales

Benina Velázquez Ordoñez, Jesús Manuel Olivares Ceja, Miguel Patiño Ortíz,
Julián Patiño Ortíz, Adolfo Guzmán Arenas

***Resumen*—Se ha detectado que en algunas aplicaciones de integración de información de fuentes de datos, en algunos casos pueden ocurrir inconsistencias y en otros, se carece de una entidad para almacenar los datos. Algunas inconsistencias se deben a que los datos se expresan en diferente idioma al utilizado en el repositorio o por el uso de diferentes unidades de medida. En este artículo, la propuesta utiliza reglas en la integración de datos tratando de preservar la consistencia y en otros casos implican modificaciones al esquema. Se seleccionó el modelo orientado a objetos por sus características que facilitan la reutilización de clases. La base de datos de ejemplo utiliza datos obtenidos de fuentes heterogéneas de la Web pertenecientes al dominio de equipos de computación. En la integración, intervienen entidades, atributos, valores y unidades de medida. Esta propuesta se enfoca en el contenido que es una alternativa a la integración de esquemas de datos.**

***Palabras clave*—Integración de datos, información compartida, intercambio de información, bases de datos orientadas a objetos.**

Integration of Heterogeneous Textual Data Sources

***Abstract*—This paper proposes an alternative to data integration from heterogeneous sources or databases. In some cases, inconsistencies may occur, and in others, the schema lacks of any attribute or entity to store the data. Some inconsistencies are consequence of using a language different with the one employed in the schema definition; others are due to the use of distinct units of measure. The object-oriented model provides characteristics that facilitate the class reuse and extension. The samples are obtained from heterogeneous Web sources belonging to the domain of computer equipment. Integration involves entities, attributes, values, and units of measurement.**

Manuscrito recibido el 19 de junio de 2014, aceptado para su publicación el 10 de julio de 2014, publicado el 15 de junio 2015.

Benina Velázquez Ordoñez (autor correspondiente) estudia en el Instituto Politécnico Nacional (IPN), en la Escuela Superior de Ingeniería Mecánica y Eléctrica (ESIME), DF, México (correo: bvelazquez@ipn.mx).

Jesús Manuel Olivares Ceja y Adolfo Guzmán Arenas trabajan en el IPN, en el Centro de Investigación en Computación (CIC), México, DF (correo: jesus@cic.ipn.mx, a.guzman@ieee.org)

Miguel Patiño Ortíz y Julián Patiño Ortíz trabajan en el IPN–ESIME, DF, México (correo: {mpatino2002, jpatinoo}@ipn.mx).

***Index Terms*—Data integration, information sharing, information exchange, object oriented databases**.

I. Introducción

Actualmente, como consecuencia de la globalización varias organizaciones requieren integrar datos de fuentes heterogéneas en forma eficiente. El problema de la integración se considera en dos vertientes: integración del esquema y del contenido [9, 22]. La mayoría de los trabajos publicados se enfocan en la integración de esquemas [1, 2, 7, 8, 12, 14, 16, 18, 19], sin embargo [9, 22, 25], también tratan la integración de contenido. El principal problema que se menciona en estos trabajos es la preservación de la *consistencia* de datos y de las relaciones entre ellos al hacer la integración.

Este artículo se enfoca en la integración de documentos heterogéneos obtenidos de la Web en forma de texto plano hacia un modelo de base de datos orientado a objetos; este modelo se seleccionó porque actualmente representa un intermediario entre el modelo semántico más cercano a las abstracciones manejadas por el usuario y aquellas propias del modelo relacional que utiliza (tablas) con la ventaja de ser más eficiente al ejecutarse y validar las restricciones de integridad.

La integración se hace en forma incremental conforme se leen los datos, algunos casos de integración implican modificar el esquema.

En este documento se trata el dominio de equipos de computación obtenidos de diferentes sitios de la Web.

Históricamente en el modelo entidad-relación extendido [20], además de las entidades y relaciones, se incorpora la generalización y especialización; estableciendo las bases para el modelo orientado a objetos utilizado recientemente y cada vez con más frecuencia. Los objetos se asemejan a los marcos de Minsky [26] que se propusieron para representar entidades de conocimiento estereotipadas desde diferentes puntos de vista, con elementos que se dividen en descriptivos y conductuales; sirven para facilitar la inferencia en forma más amplia que los términos y expresiones ofrecidos por la lógica.

El propósito fundamental de los objetos es el manejo de la complejidad, de acuerdo con [23] los principios para el manejo de la complejidad son: la abstracción, la encapsulación, la herencia, la asociación, la comunicación de

ISSN 2395-8618

mensajes mediante métodos, la escala y la clasificación. Por otra parte [24] indica que los objetos están compuestos por cuatro principales elementos: la abstracción, encapsulación, modularidad y jerarquía.

Actualmente, en las aplicaciones, generalmente, se utilizan objetos complejos formados por conjuntos de atributos. Cada atributo, a su vez puede ser simple o complejo. Un atributo simple es un número entero, cadena o booleano; mientras que un atributo complejo es una combinación de atributos simples y complejos. Con base en los atributos puede generarse una jerarquía de objetos complejos.

El objetivo de esta propuesta es la transformación de textos, redactados con diferente vocabulario e idioma, hacia nombres de clases, propiedades, valores y unidad de medida; mismos que empleando conjuntos de reglas permiten hacer la integración hacia un modelo orientado a objetos, preservando la integridad en los datos. Es posible que en algunos casos en que la ambigüedad o los errores léxicos no permitan asegurar la integridad de los datos, se presenten las partes de texto con problemas para que sean revisados por parte del usuario.

El resto del artículo se organiza como sigue: La sección II comenta algunos de los trabajos previos que se han desarrollado hasta ahora. La sección III describe las características de las fuentes textuales heterogéneas y la base de datos que se utiliza para la integración de datos. La sección IV los casos de integración considerados. La sección V describe el método de integración propuesto y la descripción de las fuentes de datos de ejemplo y finalmente en la sección VI se indican las conclusiones y trabajo futuro.

II. Trabajos previos

En [1] se introducen metadatos para resolver semánticamente la heterogeneidad en bases de datos federadas. En [2, 16] se hace la integración de un esquema XML hacia otro esquema basado en el modelo relacional. En [9] se hace el mapeo de dos esquemas de base de datos orientadas a objetos mediante la alineación de un esquema local hacia otro esquema global. En [22] se hace la integración entre dos modelos entidad-relación (ER) resolviendo diferentes relaciones semánticas. En [12] se describe la integración de esquemas orientados a objetos generando un esquema común utilizando tesauros. En [7] se utiliza una técnica basada en clasificadores y redes neuronales para hacer la integración de fuentes de XML a XML. En [14] también se hace un mapeo de XML a un esquema minimal también en XML. En [15] se describe un integrador de fuentes de datos en formato XML del dominio de datos bibliográficos. En [8] se describe la integración de esquemas de base de datos intermediado con una ontología.

En [25] se describe la arquitectura Tisimmis que integra datos de diferentes fuentes utilizadas para hacer consultas con un lenguaje común. En [18] se emplea una arquitectura similar a la de [25] complementada con el uso de una ontología para hacer la integración de fuentes XML. En [19] se describe la unificación de esquemas orientados a objetos realizando una conversión al formato XML y luego utilizando una ontología para producir un esquema común. En [22] se desarrolla la similitud semántica entre entidades del modelo relacional usando diferentes relaciones de similitud semántica.

III. Fuentes y base de datos de ejemplo

En este documento se presenta una alternativa a la integración de datos de fuentes textuales hacia una base de datos orientada a objetos para el dominio de equipos de computación. La base de datos integrada tiene el propósito de ser útil para emitir sugerencias de equipo de cómputo a diferentes usuarios con base en el equipo que mejor satisfaga sus requerimientos. En este caso es muy importante que la base de datos contenga los datos de equipos más actualizados sin importar el proveedor, esto dificulta que puedan establecerse acuerdos para solicitar los datos que se incorporan en la base de datos; como consecuencia, se considera más apropiado integrar los datos de las fuentes textuales hacia la base de datos marcando el proveedor, que en este trabajo se indica como un dato anónimo por cuestiones de derechos de autor de sus marcas registradas.

La propuesta de este trabajo es un módulo integrador de datos que toma las fuentes textuales heterogéneas que deben cumplir con una estructura de encabezado, donde se identifica el nombre de la clase principal y un cuerpo de documento donde se esperan las propiedades, valores y unidades de medida. Algunas de las propiedades pueden ser objetos que componen a la clase principal. Se realiza un análisis léxico de los elementos de las fuentes textuales para identificar los elementos que coinciden con la base de datos (figura 1). El integrador utiliza un diccionario y un conjunto de reglas para encontrar los mapeos entre las palabras de un documento de texto con los elementos de la base de datos.

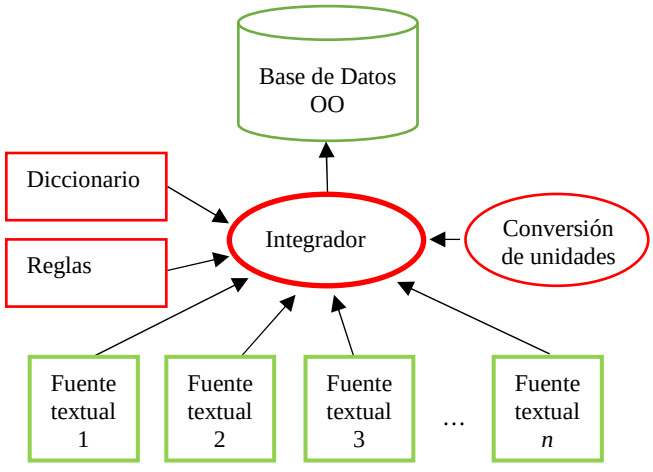

Fig. 1. Integración de fuentes textuales hacia una base de datos

 http://dx.doi.org/10.17562/PB-51-3

ISSN 2395-8618

El modelo semántico de datos [5, 6] tiene la ventaja de utilizar abstracciones cercanas al usuario facilitando la identificación de las principales entidades (clases) con un triángulo y a partir de las que se derivan las subclases identificadas con un círculo. Las clases formadas con la agrupación de propiedades (variables) se indica con un círculo con una cruz, los atributos se indican con un ovalo y cuando son atributos multivaluados se preceden de un circulo con doble cruz. En la figura 2 se utiliza un modelo semántico para mostrar algunas de las entidades principales de la base de datos de ejemplo. A partir del modelo semántico se obtiene el modelo orientado a objetos que como diferencia del anterior las clases y superclases utilizan el mismo símbolo con lo cual se destacan las relaciones de herencia, composición, agregación y asociación. En la figura 3 se indican las clases que destacan las relaciones de la clase computadora con sus componentes y sus relaciones de herencia.

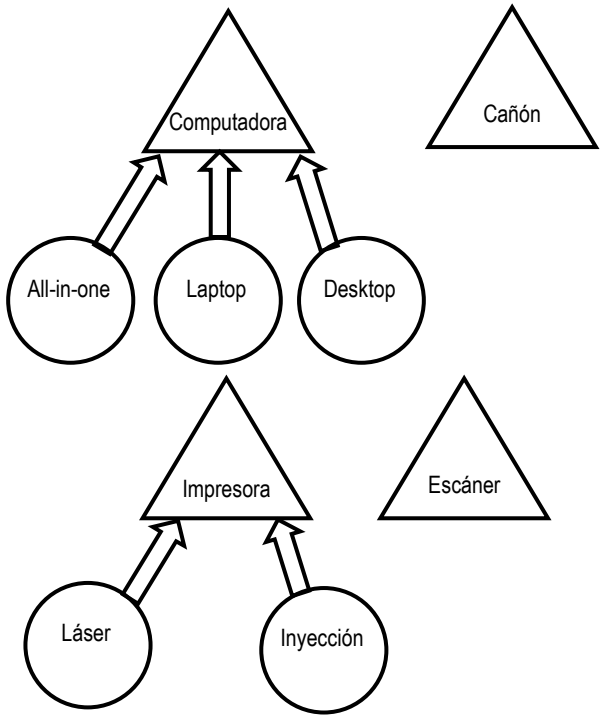

Fig. 2. Modelo semántico de la base de datos de ejemplo

La integración de datos hacia el esquema orientado a objetos utiliza fuentes textuales de equipo de cómputo (figura 4). Se utiliza la convención que cada fuente textual tiene un encabezado que se mapea hacia el nombre de una clase. El resto del documento está formado por una lista de datos que se procesan con un analizador léxico para separar las propiedades, valores y unidades de medida de cada uno. Algunas propiedades pueden ser multivaluadas u objetos que componen al objeto principal.

En los textos se aplican algunas adecuaciones de los símbolos previo al análisis léxico, entre estas está el cambio de los signos ® y ™ a la notación (R) y (TM) respectivamente. Los exponentes se cambian al signo ^ y el valor, de esta forma m2 se cambia a m^2, algunas unidades como los dpi de impresión se escriben como 9.600 dpi por lo que deben cambiarse a 9600 dpi, en general a los números se les eliminan las comas separadoras o los puntos cuando se trata de enteros. Algunas veces las unidades como 2GB se escriben sin espacio por lo que se separan 2 GB.

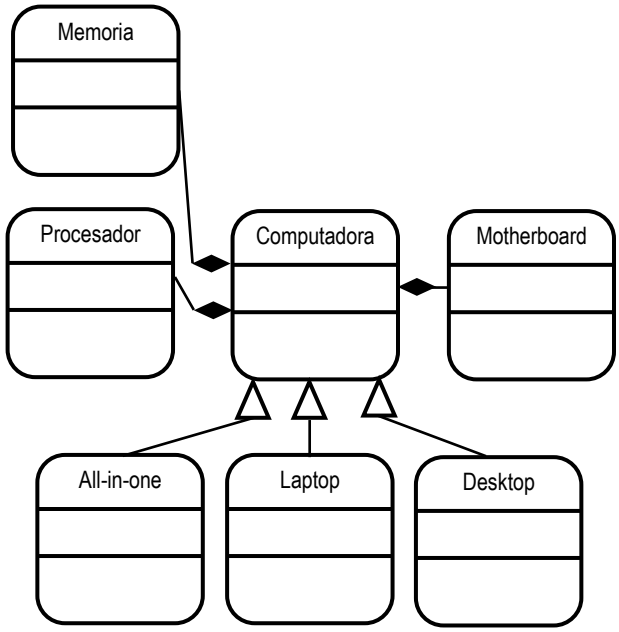

Fig. 3. Algunas clases del modelo orientado a objetos

Las palabras se utilizan respetando las mayúsculas y minúsculas. Después de las adecuaciones de valores y unidades se procesan las unidades léxicas [10] para corregir los errores de ortografía.

El mapeo de palabras a los elementos de la base de datos utiliza un diccionario dividido en secciones para cada uno de los elementos de la base de datos considerados: nombres de clases, identificadores de clases, nombres de propiedad, valores y unidades de medida.

En la figura 5 se muestran las estructuras que componen al diccionario. Las estructuras se llenan en tres etapas, en la etapa inicial se llena la columna de Clase, Identificador, Propiedad, Valor, Unidad con el contenido del esquema de la base de datos orientada a objetos. En la segunda etapa se utilizan recursos léxicos disponibles en la Web para asignar palabras y sinónimos que representan a los elementos de las tablas, registrándolos en la columna Palabras del elemento al que corresponde. La tercera etapa es durante el proceso de integración, cuando se encuentran palabras o símbolos que no se encuentran en el diccionario y que, a criterio del administrador, se añaden. Las palabras e identificadores que no se encuentren en el diccionario se marcarán como desconocidos y no se utilizarán en el mapeo.

ISSN 2395-8618

a) Descripción de una computadora en idioma Inglés

XPS 8700 Special Edition (Computer)
Processor<TAB>4th Generation Intel® Core™ i7
SO<TAB>Windows 8.1
Memory<TAB>16GB
Hard Drive <TAB>2TB
Monitor<TAB>Includes a 23" Dell monitor to maximize your media experience (a $219 value).

b) Descripción de una computadora en idioma Español

XPS 8700 (Computadora)
Procesador<TAB> Cuarta generación del Intel® Core™ i7-4770 (8MB Caché, hasta 3.90 GHz)
SO<TAB>Windows 8.1 Single Language (64Bit) Spanish
Memoria<TAB>16 GB[1] Dos canales SDRAM DDR3 a 1600 MHz
Disco Duro<TAB> SATA de 2TB 7200 RPM (6.0 Gb/s) + Disco Duro de Estado Solido (SSD) de 32GB con Tecnología de Respuesta Inteligente (SRT)
Tarjeta de video<TAB>AMD Radeon™ HD R9 270 2GB GDDR5
garantía Estándar<TAB> 1 Año de, con Servicio en el sitio al siguiente día laborable
Unidad combo<TAB>DVD-RW o Blu-ray Disc
Chipset<TAB>Chipset Intel® Z87 Express

Fig. 4. Ejemplo de fuentes textuales obtenidas de la Web

IV. Integración de fuentes heterogéneas de datos

La integración de las fuentes heterogéneas en este documento se divide en integración hacia el esquema y hacia el contenido.

A. Casos de mapeo del esquema y contenido

La identificación de las clases o subclases se hace utilizando los atributos, la clase se asigna con base en la mayor similitud entre propiedades.

Si se detecta un documento que tiene un nombre de clase y atributos que no mapean con alguna clase existente, se presenta al usuario con sus propiedades para que se considere su integración como una clase o subclase nueva en el modelo de datos.

La integración de datos se hace mediante el mapeo de palabras hacia los elementos del esquema de la base de datos. El encabezado de los documentos textuales se utiliza para encontrar el nombre de la clase y opcionalmente el tipo de equipo. El resto de los elementos se utilizan para encontrar los nombres de las propiedades, los valores y cuando se encuentran las unidades de medida. La integración se hace primero hacia el esquema y luego hacia el contenido. El mapeo del esquema se desarrolla en cuatro casos caracterizados mediante la notación siguiente: a) cuando se puede encontrar un mapeo hacia una clase usando las palabras del texto se indica "$\exists$ clase", en caso contrario se indica "$\neg\exists$ clase"; b) Cuando se encuentra un mapeo hacia las propiedades de alguna clase, no necesariamente de la clase de a), se indica "$\exists$ propiedades", en caso contrario se indica "$\neg\exists$ ropiedades", esto también se aplica cuando al menos una propiedad es diferente.

En los casos se tiene un antecedente que se debe cumplir para que se considere aplicable y un consecuente que indica las acciones que deben aplicarse como resultado de cada caso. Los casos aplicables a la integración del esquema se indican en la tabla I. En forma similar como se aplica en el área de sistemas expertos, se deja al experto humano la decisión de modificar el esquema y el contenido de la base de datos por lo que las reglas se utilizan para proponer cambios en el esquema.

El caso A se presenta cuando no se encuentra una clase en el esquema indicada por las palabras del encabezado de la fuente y tampoco existe una clase que tenga las propiedades encontradas con los mapeos de las palabras del contenido del texto. En este caso se hace el mapeo de valores y unidades de medida para completar una clase con su contenido y se le propone al usuario para darla de alta en el esquema y en la base de datos.

TABLA I
CASOS DE INTEGRACIÓN DEL ESQUEMA

CASO	ANTECEDENTE	CONSECUENTE
A	Si $\neg\exists$ clase $\wedge$ $\neg\exists$ propiedades	Crear una clase nueva con las propiedades detectadas
B	Si $\exists$ clase $\wedge$ $\neg\exists$ propiedades	Crear una clase heredando de una superclase común
C	Si $\neg\exists$ clase $\wedge$ $\neg\exists$ propiedades	Verificar si la clase es sinónimo de una existente o crear otra
D	Si $\exists$ clase $\wedge$ $\exists$ propiedades	Realizar el mapeo de valores y unidades de medida a la clase

En el caso B se encuentra coincidencia en el nombre de la clase pero no de las propiedades de una clase, por lo que se procede a mapear los valores y unidades de medida y se le indica al usuario que revise la clase y en dado caso se tendrá que crear una superclase que agrupe a la clase o clases que tienen propiedades similares con la que se está mapeando.

El caso C se presenta cuando existe coincidencia completa de las propiedades de una clase pero el nombre es diferente. Esta situación se puede presentar cuando algunos dispositivos similares como el caso de las laptop con algunas tabletas pero deben registrarse en clases diferentes. En otros casos similares puede tratarse de un sinónimo como desktop y PC-escritorio.

El caso D se presenta cuando existe coincidencia en el nombre de la clase y sus propiedades, por lo que se requiere proceder a verificar si los valores y unidades de medida existen como algún objeto entre los existentes. Una de las

ISSN 2395-8618

contribuciones en este documento es utilizar las unidades de medida para utilizar funciones de equivalencia y que permiten detectar que dos objetos son iguales después de aplicar las funciones de equivalencia, por ejemplo si dos equipos con propiedades-valor similares tiene un costo de USD$1,000 y otro con costo de MXP$ 13,000 al aplicar una conversión de unidades se encuentra que son iguales. Este caso se divide en cuatro sub-casos como se indica en la tabla II, estos se aplican a una clase con sus propiedades y los objetos de la misma representados por sus valores y unidades de medida.

TABLA II
SUB-CASOS DE INTEGRACIÓN DEL CONTENIDO

SUBCASO	ANTECEDENTE	CONSECUENTE
D.1	Si valores = ∧ unidades =	Objeto existente, se omite
D.2	Si valores = ∧ unidades ≠	Se igualan las unidades y se añade como un objeto nuevo
D.3	Si valores ≠ ∧ unidades =	Se añade como un objeto nuevo
D.4	Si valores ≠ ∧ unidades ≠	Se igualan las unidades y se prueba si es D.1 o D.3

El subcaso D.1 ocurre cuando los valores mapeados de una fuente textual coinciden con los de un objeto de la clase detectada en todas sus propiedades y lo mismo ocurre con las unidades de medida, por lo tanto se le notifica al usuario que se tiene un objeto duplicado y que debe omitirse.

El subcaso D.2 se presenta cuando los valores mapeados coinciden con un objeto existente pero hay diferencias en sus unidades de medida, esto hace necesario que se apliquen funciones de equivalencia que generalmente producen cambios en los valores por lo tanto se vuelve a verificar si es un objeto diferente y se añade a los existentes, en caso que con la aplicación de las funciones de equivalencia se obtenga un objeto existente se procede como en D.1 reportando que es un objeto duplicado.

El subcaso D.3 añade un objeto como consecuencia que se detectaron valores diferentes con las mismas unidades de medida y ningún objeto es igual.

El subcaso D.4 en que los valores y unidades de medida son diferentes se deben aplicar primero las funciones de equivalencia para igualar las unidades de medida transformando los datos al sub-caso D.1 o el D.3.

B. Mapeo de palabras a elementos de base de datos

En el proceso de integración, primero se hacen las adecuaciones a las unidades léxicas y luego se resuelve la identificación de la clase y su identificador junto con las propiedades para determinar el caso que se tiene; si se encuentra el caso D entonces se determina el subcaso de acuerdo con los datos de la fuente textual.

El contenido del diccionario se utiliza para encontrar los elementos del esquema con los que se hace el mapeo del contenido de cada fuente textual.

C. Nombres de clase, propiedades y valores omitidos

En la integración de textos en ocasiones se omiten algunos elementos por lo que se utiliza el contenido del diccionario para inferir la clase o las propiedades que permiten hacer la integración en caso que se omita algún valor.

Clase

nombreClase	palabrasClase
Laptop	Computadora portátil, laptop
Impresora	Printer, printer, impresora

Identificadores

nombreIdent	palabrasIdent
XPS8700	XPS 8700
L-800	L-800

Propiedad

nombreProp	palabrasProp
Procesador	Procesador, CPU, Processor
Memoria	RAM, Memory, Memoria

Valor

nombreVal	palabrasVal
2	2
600x600	600 x 600, 600x600

Unidad

nombreUnid	palabrasUnid
Gb	Gb, GB, gigas, Gigabytes
Tb	Terabytes, teras, TB, Tb

Fig. 5. Estructuras del diccionario para encontrar mapeos

V. PRUEBAS Y RESULTADOS

En esta sección se muestran algunas pruebas y resultados del método de integración propuesto, mostrando la fuente textual, a continuación los elementos mapeados obtenidos y el caso que se aplica.

Sea el texto original:

XPS 8700 Special Edition (Computer)
Processor<TAB>4th Generation Intel® Core™ i7 SO<TAB>Windows 8.1 Memory<TAB>16GB Hard Drive <TAB>2TB Monitor<TAB>Includes a 23" Dell monitor to maximize your media experience (a $219 value).

ISSN 2395-8618

Al efectuar la adecuación se obtiene:

XPS 8700 Special Edition (Computer)
Processor<TAB>4th Generation Intel(R) Core(TM) i7
SO<TAB>Windows 8.1
Memory<TAB>16 GB
Hard Drive <TAB>2 TB
Monitor<TAB>Includes a 23 inches Dell monitor to maximize your media experience (a USD$ 219 value).

Después del mapeo de unidades léxicas se obtienen varias listas de datos que se utilizan para integrarse en la base de datos. En este ejemplo se observa que hace falta el valor de la propiedad velocidad para el caso del procesador y también faltan las propiedades de la clase Motherboard que es una clase que compone a desktop. Se detecta por lo tanto el caso B. Se requiere que previo a su integración, el usuario debe completar los datos faltantes. Las unidades son opcionales porque solamente algunas están presentes con los valores de sus propiedades (figura 6).

Clase	**Instancia**
Desktop	XPS8700

Propiedad	**Valor**	**Unidad**
discoDuro	2	Tb
monitor	tamaño	23 pulgadas

Clase compuesta	**Instancia**
Procesador	core i7

Propiedad	**Valor**	**Unidad**
marca	Intel	
tipo	Core i7	
generacion	4	
velocidad	(INDEFINIDO)	Ghz

Clase compuesta	**Instancia**
Memoria	RAM

Propiedad	**Valor**	**Unidad**
tamaño	16	Gb

Clase compuesta	**Instancia**
Motherboard	(INDEFINIDO)

Propiedad	**Valor**	**Unidad**
fabricante	(INDEFINIDO)	
modelo	(INDEFINIDO)	

Fig. 6. Elementos del ejemplo que se integran en la base de datos

VI. Conclusiones

Se ha presentado una alternativa de integración de datos textuales de fuentes heterogéneas mediante la extracción de los nombres de clases, identificadores, propiedades, valores y unidades de medida. El método se divide en integración del esquema y de contenido. En los casos del esquema una opción es proponer la creación de clases nuevas o adicionar propiedades en alguna clase existente. En la integración de contenido con el apoyo de las funciones de conversión es posible detectar objetos repetidos.

Este método agiliza el proceso de integración cuando se carece de un esquema de datos en los datos fuente, por ejemplo, cuando se aprovechan datos existentes en la Web en forma pública y que se requieren almacenar en un repositorio de datos estructurados.

En la propuesta aún se requiere de la intervención del usuario en la decisión final de creación o actualización del esquema.

El trabajo futuro es aprovechar las estructuras propuestas para dar sugerencias a usuarios que proporcionan información incompleta o con errores.

Agradecimientos

Benina Velázquez Ordoñez y Jesús Manuel Olivares Ceja agradecen el apoyo y comentarios de la Dra. Blanca Lidia Miranda Valencia.

References

[1] G. Aslan and D. McLeod, "Semantic heterogeneity resolution in federated databases by metadata implantation and stepwise evolution," *The VLDB Journal*, vol. 8, no. 2, pp. 120–132, Oct. 1999.

[2] M. Atay, et al., "Efficient schema-based XML-to-Relational data mapping," *Information Systems*, vol. 32, no. 3, pp. 458–476, May 2007.

[3] G. Davies and L. Ekenberg, "Model correspondence as a basis for schema domination," *Knowledge-Based Systems*, vol. 23, no. 7, pp. 693–703, Oct. 2010.

[4] R. C. Goldstein and V. C. Store, "Data abstractions: Why and how?," *Data & Knowledge Engineering*, vol. 29, no. 3, pp. 293–311, Mar. 1999.

[5] R. Hull and R. King, "Semantic database modeling: survey, applications, and research issues," *ACM Computing Surveys*, vol. 19, no. 3, pp. 201–260, Sept. 1987.

[6] R. Hull, "Managing Semantic Heterogeneity in Databases: A Theoretical Perspective," *Proc. ACM Symposium on Principles of Database Systems (PODS'97)*, pp. 51–61, 1997.

[7] B. Jeong, D. Lee, H. Cho and J. Lee, "A novel method for measuring semantic similarity for XML schema matching," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1651–1658, Apr. 2008.

[8] J. Kohler, et al., "Logical and Semantic Database Integration," *Proc. 1st IEEE International Symposium on Bioinformatics and Biomedical Engineering (BIBE '00)*, pp. 77–80, 2000.

[9] E.-P. Lim and R. H. L. Chiang, "Accommodating instance heterogeneities in database integration," *Decision Support Systems*, vol. 38, no. 2, pp. 213–231, Nov. 2004.

[10] C. D. Manning, P. Raghavan and H. Schütze, *An Introduction to Information Retrieval*, Cambridge, MA: Cambridge University Press, 2009.

[11] S. Madria, K. Passi and S. Bhowmick, "An XML Schema integration and query mechanism system," *Data & Knowledge Engineering*, vol. 65, no. 2, pp. 266–303, May 2008.

[12] I. Mirbel, "Semantic integration of conceptual schemas," *Data & Knowledge Engineering*, vol. 21, no. 2, pp. 183–195, Jan. 1997.

ISSN 2395-8618

[13] M. L. Nguyen and A. Shimazu, "A semi supervised learning model for mapping sentences to logical forms with ambiguous supervision," *Data & Knowledge Engineering*, vol. 90, no. 1, pp. 1–12, Mar. 2014.

[14] H.-Q. Nguyen, et al., "Double-layered schema integration of heterogeneous XML sources," *The Journal of Systems and Software*, vol. 84, no. 1, pp. 63–76, Jan. 2011.

[15] H. Nottelmann and U. Straccia, "Information retrieval and machine learning for probabilistic schema matching," *Information Processing and Management*, vol. 43, no. 3, pp. 552–576, May 2007.

[16] G. Della Penna, et al., "Interoperability mapping from XML schemas to ER diagrams," *Data & Knowledge Engineering*, vol. 59, no. 1, pp. 166–188, Oct. 2006.

[17] G. Pirró, "A semantic similarity metric combining features and intrinsic information content," *Data & Knowledge Engineering.* Vol. 68, no. 11, pp. 1289–1308, Nov. 2009.

[18] J.-L. Seng and I.L. Kong, "A schema and ontology-aided intelligent information integration," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10538–10550, Sept. 2009.

[19] R. dos Santos Mello, S. Castano and C. A. Heuser, "A method for the unification of XML schemata," *Information and Software Technology*, vol. 44, no. 4, pp. 241–249, Mar. 2002.

[20] J. M. Smith and D. C. P. Smith, "Database Abstractions: Aggregation and Generalization," *ACM Transactions on Database Systems*, vol. 2, no. 2, pp. 105–133, June 1977.

[21] Victor Vianu. "A Web Odyssey: from Codd to XML," *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems (PODS '01)*, 1–15, 2001.

[22] William Wei Song, Paul Johannesson, Janis A. Bubenko Jr. "Semantic similarity relations and computation in schema integration," *Data & Knowledge Engineering*, Vol. 19, no. 1, pp. 65–97, May 1996.

[23] P. Coad and E. Yourdon, *Object-Oriented Design*, Yourdon Press, New Jersey, 1991.

[24] G. Booch. *Object Oriented Design with Applications*, New York: Benjamin/Cummings, 1994.

[25] H. Garcia-Molina, et al. "The TSIMMIS project: integration of heterogeneous information sources," *Journal of Intelligent Information Systems*, Vol. 8 no. 2, pp. 117–132, 1997.

[26] M. Minsky, "A Framework for Representing Knowledge," MIT-AI Laboratory Memo 306, June, 1974