Una Arquitectura de Integración de Datos como Base de un Lenguaje Conceptual de Recuperación de Datos Médicos

María Constanza Pabón[1] and Martha Millán[2]

[1] Departamento de Ciencias e Ingeniería de la Computación
Pontificia Universidad Javeriana
Cali, Colombia
mcpabon@javerianacali.edu.co
http://www.javerianacali.edu.co/
[2] Escuela de Ingeniería de Sistemas y Computación
Universidad del Valle
Cali, Colombia
martha.millan@correounivalle.edu.co
http://www.univalle.edu.co/

Abstract. En el campo médico, la información digital se ha incrementado significativamente. Sin embargo, su integración, para apoyar procesos de diagnóstico, de investigación o de enseñanza, entre otros, implica acceder tanto a fuentes como a tipos de estructuras heterogéneas (e.g. imágenes médicas, datos clínicos del paciente, datos demográficos). Una forma de lograr esta integración es usar una arquitectura basada en mediación, en la cual, un modelo de datos global representa la información de las diferentes fuentes de datos. En este artículo se describe una arquitectura de integración cuyo mediador usa un modelo de datos conceptual. La arquitectura subyace a un lenguaje conceptual de consulta y recuperación de datos diversos y heterogéneos en el dominio médico. El lenguaje es de alto nivel, cercano al usuario, e independiente de los detalles de almacenamiento de los datos.

Digital information in medical domain has increased significantly during last years. However, its integration, in order to support processes of diagnosis, research or learning, involves access to heterogenous repositories and data structures (e.g. medical images, clinical and demographic data). One approach to achieve this integration is the use of a mediation-based architecture, with a global data model that represents the data avalaible on diverse repositories. This article describes an integration architecture, based on a mediator which uses a conceptual data model, in order to propose a conceptual query language to retrieve diverse and heterogeneous data on medical domain. The query lenguage is a high level language, close to the user, and independent of the details of the structures with which data is stored.

Keywords: Conceptual models, data integration, query languages, data retrieval in medical domain

1 Introducción

Una de las fuentes de generación de importante información médica son las imágenes. Sin embargo, recuperarlas y analizarlas, de manera aislada, con base únicamente en sus características de bajo nivel o en sus características semánticas, generalmente, no le ofrece a un médico respuestas satisfactorias. Se requiere entonces, relacionar los datos y la información proveniente de la imagen con datos de otras fuentes, tales como la historia clínica, los reportes de diagnóstico, los datos demográficos del paciente, los metadatos de la imagen, y las ontologías del dominio médico. Estos datos son de diversa naturaleza y pueden residir en más de un repositorio. Se hace entonces necesario integrar los datos, de manera que los usuarios puedan recuperar información relacionada, sin preocuparse de su forma, estructura o del lenguaje de consulta propio de cada fuente de almacenamiento.

De otra parte, los médicos requieren realizar consultas *ad hoc* sobre los datos integrados. Por esta razón, se hace necesario proveer herramientas que faciliten formularlas, independientemente de su complejidad. En este artículo se propone una arquitectura de integración bajo el enfoque de traducción de consultas, basada en un mediador, en cuyo modelo global subyace un modelo conceptual. Bajo esta arquitectura propuesta es posible ofrecerle al usuario un lenguaje de consulta de alto nivel, fácil de usar, con una sintaxis simple, y con un poder expresivo que le permita formular consultas útiles en el campo médico.

El resto de este artículo está organizado en tres secciones. En la sección 2, se presentan los conceptos y los trabajos relacionados con integración de fuentes de datos heterogéneas y proyectos de integración de datos en el dominio médico, modelos de datos conceptuales, y los lenguajes de recuperación de datos que se proponen con base en dichos modelos. En la sección 3, se describe la arquitectura propuesta y, finalmente, en la sección 4, se presentan algunas conclusiones y se proponen algunas líneas de trabajo futuro.

2 Trabajos Relacionados

En esta sección se presentan los conceptos relacionados con integración de datos, modelos conceptuales, y lenguajes de recuperación de datos basados en modelos conceptuales, como áreas principales sobre las cuales se apoya la arquitectura propuesta. Se incluyen también algunos de los sistemas desarrollados, particularmente en el dominio médico.

2.1 Integración de Datos de Fuentes Heterogéneas

La integración de datos [24] busca proveer a los usuarios de una vista unificada, integrada y global de datos que se encuentran en fuentes diversas, heterogéneas y autónomas. De esta manera, los usuarios pueden formular consultas que el sistema ejecuta de manera transparente. La vista puede ser materializada o virtual [5], materializada cuando se replican los datos de las diversas fuentes

en un repositorio central, y virtual cuando los datos se acceden en sus fuentes locales. Diferentes modelos y arquitecturas se han propuesto para resolver el problema de integración de datos [35, 42, 36, 45, 4, 13, 22]. Desde el punto de vista de cómo se implementa la integración, los enfoques utilizados son [42]: *Data Translation*, *Query Translation* e *Information Linkage*. En *Data Translation* [8], la vista es materializada y los datos de las diferentes fuentes se copian a un repositorio central adecuándolos a un esquema unificado. *Query Translation*, usa una vista virtual y provee mecanismos que permiten acceder a los datos en sus respectivos repositorios. Finalmente, en *Information Linkage*, se crean referencias entre los datos de las diferentes fuentes que corresponden a un mismo objeto del mundo real.

En particular, en la propuesta de integración de datos que aquí se presenta, el enfoque *Query Translation* parece ser el más adecuado, si se tiene en cuenta que bajo un enfoque *Data Translation* se requeriría gran capacidad de almacenamiento, particularmente para las imágenes médicas. Por otro lado, *Information Linkage* no requiere de un modelo global que represente la información de las fuentes, por lo cual, no sería útil como base para la implementación de un lenguaje conceptual para recuperación de datos e imágenes médicas.

En el enfoque *Query Translation* (traducción de consultas), las consultas del usuario se procesan en cada una de las fuentes de datos. Comunmente, en este enfoque, se usa una arquitectura basada en mediación [44], que incluye la interfaz de usuario, un mediador, *wrappers*, y las fuentes de datos. La interfaz de usuario captura la consulta y la entrega al mediador. El mediador procesa globalmente la consulta, con base en un modelo de datos global y en los mapeos entre éste y los esquemas de datos locales y genera subconsultas que envía a las fuentes de datos pertinentes. El lenguaje o la representación de las subconsultas en el mediador puede ser diferente del lenguaje o de la representación usada para capturar la consulta del usuario y de los lenguajes de consulta de las fuentes de datos. Cuando el lenguaje de las subconsultas es diferente al lenguaje de consulta de la fuente de datos, se necesita un *wrapper* que haga la traducción entre los dos lenguajes. Finalmente, después de que las subconsultas se ejecutan en cada fuente de datos, el mediador recibe los resultados, los integra y los entrega al usuario. La Figura 1, muestra una arquitectura genérica de un integrador basado en mediación, adaptada de [43].

Entre los enfoques basados en mediación, los modelos de integración semántica de datos [30] se caracterizan por construir el modelo global considerando, únicamente, aquellos elementos que son necesarios para representar la semántica del dominio, sin incluir detalles de implementación. Varios métodos de modelado de conocimiento y de datos cumplen esta característica, entre ellos las ontologías y los modelos de datos conceptuales. A su vez, los modelos de datos conceptuales permiten ofrecer lenguajes de consulta de alto nivel, cercanos al usuario y que no incluyen detalles de la estructura de los datos o de las características propias del DBMS (*DataBase Management System*) usado para implementar el repositorio de datos.

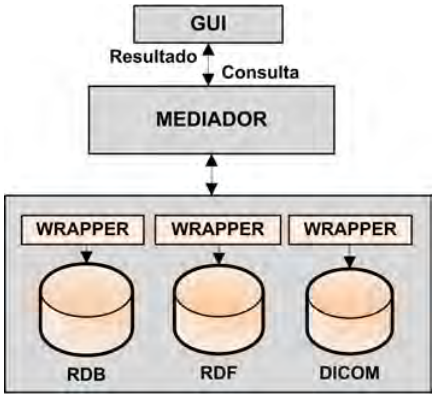

Fig. 1. Arquitectura de Integración Basada en Mediación

Existen dos aproximaciones para hacer el mapeado entre los esquemas de datos locales de las fuentes y el modelo global [5]: punto de vista global (*Global-As-View* - GAV) o punto de vista local (*Local-As-View* - LAV). En el primero, el modelo global se define a partir de los esquemas locales, y cada uno de sus elementos se mapea a una vista que combina datos de los esquemas locales. En el segundo, LAV, el modelo global es independiente de los esquemas locales, y cada elemento de los esquemas locales se mapea a una vista del modelo global. Este último, facilita la extensibilidad del sistema, haciendo posible agregar nuevas fuentes locales sin alterar el modelo global. También se han propuesto enfoques híbridos, como GLAV [10].

2.2 Integración de Datos en el Dominio Médico.

En el dominio médico, se han desarrollado soluciones que integran semánticamente fuentes de datos heterogéneas, algunas de las cuales implementan el enfoque *Query Translation*, entre ellas, MEDICO [39], NIF [12] (*Neuroscience Information Framework*), y ACGT [42] (*Advancing Clinico-genomic Trials on Cancer*).

MEDICO [39, 27], hace parte del proyecto THESEUS, cuyo propósito es construir un motor de recuperación de imágenes médicas. MEDICO permite consultar imágenes en diferentes formatos, datos de pacientes y metadatos, a través de imágenes de ejemplo y conceptos (vocabulario de las ontologías). Para ello, incluye una jerarquía de ontologías (de representación, superiores, médicas, clínicas y de anotación) cuyos conceptos se usan para anotar las imágenes, dotándolas así de semántica. La arquitectura de MEDICO consta de cuatro capas: de aplicación, de consultas/inferencia/análisis, de acceso a los datos, y de datos externos. La capa de aplicación recibe las consultas y la divide según el tipo de los datos a los que hace referencia (imagen, texto, metadatos). La capa de consultas/inferencia/análisis, ejecuta las subconsultas en los diferentes motores

de búsqueda. Se ofrecen mecanismos para recuperación de imágenes basada en sus características de bajo nivel, en texto y en metadatos semánticos. La capa de acceso a los datos, provee un acceso unificado a las diferentes fuentes de datos. Finalmente, en la capa de datos externos se encuentran las fuentes de datos. Sonntag et al. [40] proponen, para MEDICO, un sistema de recuperación basado en un diálogo, que incluye, entre otros, reconocimiento del habla y procesamiento de lenguaje natural.

El proyecto NIF [12] tiene como objetivo brindar un acceso integrado a diversos recursos sobre neurociencia disponibles en la web. En NIF, un usuario formula consultas basadas en conceptos de una ontología de dominio (*NIFSTD*: *NIF Standard*), para recuperar datos de sitios web, bases de datos relacionales, documentos XML y documentos en texto. La arquitectura de NIF consta de cinco capas: de aplicación, de búsqueda, de estructura de datos, de computación y consultas, y de datos. La capa de aplicación, permite a los usuarios administrar un catálogo de recursos y realizar consultas, expande las consultas con términos relacionados de la ontología, y da al usuario la posibilidad de hacer refinamiento sobre esos términos. La capa de búsqueda procesa las consultas e incluye *wrappers* que las transforman para enviarlas a las fuentes de datos en diversos formatos (SQL, HTTP calls, XML request). En la capa de estructuras de datos se encuentran los índices, que permiten identificar las fuentes que tienen datos relacionados con una consulta, y los catálogos de recursos, que incluyen los esquemas de las bases de datos relacionales. En la capa de computación y consultas están los módulos para gestionar la ontología NIFSTD, el integrador de datos estructurados, que permite acceder a bases de datos relacionales a través de consultas SQL, operaciones GET y POST de HTTP, invocación de servicios, o usando un mediador BIRN-M [14] y el prostprocesador que hace un ranking de los resultados que provienen de sitios Web. Finalmente, la capa de datos incluye el subsistema Textpresso (para búsqueda en publicaciones en texto, indexadas), las fuentes de datos, y la ontología NIFSTD.

Por su parte, ACGT [25, 26] busca proveer acceso integrado a bases de datos clínicos, genéticos e imágenes, a través de una infraestructura *grid*, con el fin de dar soporte a estudios sobre cáncer. La plataforma ACGT incluye una ontología (ACGT-MO), una capa de mediación semántica (ACGT-SM) y un servicio de acceso a datos (ACGT-DAS). ACGT-MO representa el dominio de la investigación médica en cáncer y actúa como un esquema global para la integración. Las consultas se expresan con SPARQL haciendo referencia a los conceptos y relaciones representados en ACGT-MO. ACGT-SM trata de resolver la heterogeneidad semántica e implementa un enfoque LAV de traducción de consultas. Los esquemas de las fuentes locales se definen en RDF usando la terminología y relaciones de la MO. Los mapeos entre ACGT-MO y los esquemas RDF de las fuentes de datos se basan en *path mapping*. ACGT-SM divide la consulta en subconsultas, expresadas también con SPARQL, que se procesan en diferentes fuentes de datos. ACGT-DAS provee servicios para acceder a los diferentes tipos de fuentes de datos (heterogeneidad sintáctica) y retorna, en XML, los resultados. ACGT-DAS también exporta los modelos de datos de

las fuentes, para que los clientes los puedan usar en la construcción de las consultas, y permite la conexión a diferentes tipos de fuentes de datos (bases de datos relacionales, imágenes médicas (PACS), bases de datos públicas (Web), y archivos en diversos formatos). Hasta el momento los autores han reportado la implementación de los servicios de acceso a bases de datos relacionales y a PACS.

Parece ser una tendencia, en las soluciones existentes, el realizar las búsquedas con conceptos de la ontología. Sin embargo, esta forma de hacer las búsquedas limita la expresividad del usuario, a la hora de formular una consulta, porque no es posible ni especificar relaciones entre estos conceptos ni imponer condiciones sobre los mismos. De otra parte, lenguajes de consulta, como SPARQL o SQL, proveen mayor expresividad, pero es poco usual que ésta la pueda aprovechar el usuario de la aplicación, porque para formular la consulta, require conocer y entender la estructura utilizada para describir los datos (e.g. tripletas, en el caso de SPARQL). Otra tendencia de las soluciones actuales es usar una ontología de dominio como modelo global para la integración, con la cual se describe, de forma más general, el dominio (en comparación con un modelo de datos), y permite solucionar la heterogeneidad semántica (heterogeneidad en el contenido de la información). Por otro lado, estos sistemas integran diversos tipos de datos, aunque solo MEDICO soporta la búsqueda de imágenes con base en sus características de bajo nivel. Adicionalmente y a diferencia de ACGT, NIF y MEDICO no hacen mapeos entre un modelo global y los modelos de datos locales, sino que eligen las fuentes apropiadas para la consulta por la información de sus índices.

2.3 Modelos de Datos Conceptuales

Un modelo de datos [38] es un conjunto de herramientas que permiten representar las entidades del mundo real y sus relaciones, estructurar y describir los datos, mantenerlos, y recuperarlos o consultarlos [1]. Los modelos de datos se diferencian por los detalles semánticos que pueden expresar, identificándose cuatro niveles [3, 9]: físico, lógico, conceptual, y externo (de programación/presentación). El modelo físico, describe la representación de los datos en los medios físicos (discos o memoria). En el nivel lógico, los datos se especifican siguiendo un modelo que impone el DBMS (*Database Management System*) en el cual se hará la implementación. El modelo conceptual [29, 15] representa la percepción del usuario sobre sus datos, usa conceptos específicos del dominio, y no incluye detalles de la implementación. Finalmente, el modelo externo, de programación/presentación, describe las vistas de datos que se presentan a cada usuario de acuerdo con la actividad que éste realiza.

Los modelos de nivel conceptual se crearon con el fin de facilitar la comunicación entre los usuarios y los diseñadores de sistemas de información [32, 28]. Se caracterizan por permitir crear representaciones simplificadas de objetos, fenómenos o situaciones reales [3, 20], ser precisos, completos, expresivos y fáciles de usar [11], posibilitando describir, a alto nivel, aspectos relevantes de un dominio en un contexto particular [17, 7], usando un vocabulario propio

del dominio, con términos y conceptos familiares al usuario [17, 7]. Algunos de los modelos conceptuales más conocidos son el Modelo Entidad Relación [6] y ORM [16]. Los modelos del nivel conceptual (ej. modelo entidad-relación) son mapeados en un esquema lógico particular (ej. esquema relacional), que se representa en un DBMS específico [23].

Los modelos conceptuales subyacen a la definición de lenguajes de recuperación de datos, que se denominan Lenguajes de Consulta Conceptuales [15]. Los lenguajes de consulta conceptuales se han propuesto con dos objetivos [31]. El primero, formalizar las restricciones sobre los conceptos del modelo, formulándolas como una combinación de consulta y aserción sobre los resultados. El segundo, proveer un lenguaje de alto nivel y fácil de usar, para que los usuarios puedan, de forma sencilla, formular consultas complejas, de forma similar a como lo hace en su dominio, con términos específicos del mismo [23], y obviando las desventajas que se presentan por la formulación de las consultas en el nivel externo o con base en los modelos lógico o físico. En el nivel externo, las interfaces, generalmente, restringen la expresividad de la consulta y no involucran operaciones complejas. Por su parte, los lenguajes de consulta en el nivel lógico son más expresivos, puesto que permiten definir consultas complejas. Sin embargo, para usuarios no expertos en sistemas de información, son difíciles de usar, debido a que la formulación de una consulta implica conocer la estructura de almacenamiento de los datos. En este nivel, el lenguaje de consulta más usado es SQL.

Entre los lenguajes de consulta conceptuales propuestos en los últimos años están CUDL (*Conceptual Universal Database Language*) [21], *Constellation Query Language* [19], NeuroQL [41], SCQL (*Semantically Complete Query Language*) [34], y ConQuer (*Conceptual Query*) [15].

3 La Arquitectura

En esta sección se describe la arquitectura de integración de datos propuesta, que utiliza un enfoque de traducción de consultas y ofrece soporte para implementar un lenguaje conceptual de recuperación de diversos tipos y fuentes de datos en el dominio médico. Con el lenguaje se busca permitir al usuario especificar sus consultas de manera simple, sin tener en cuenta la estructura y los detalles de implementación de las fuentes de datos y sacando ventaja del conocimiento del usuario experto en el tema de la consulta. Tanto el lenguaje como la arquitectura de integración se están desarrollando de forma paralela.

La arquitectura, Fig. 2, está integrada por las componentes comunmente utilizadas: interfaz de usuario, mediador, *wrappers* y fuentes de datos. Sin embargo, se presentan algunas diferencias que se derivan del hecho de que, para esta propuesta particular, en el diseño de los componentes se han incluido características que permiten soportar mejor el lenguaje de consulta. Un componente diferenciador incluye un conjunto de ontologías de dominio de la aplicación, separadas del modelo de datos global, y un conjunto de metadatos asociados a los *wrappers* y a las fuentes de datos. La separación entre el modelo

de datos global y las ontologías de dominio obedece a la necesidad de definir la estructura del lenguaje de consulta. Las consultas se expresan combinando estructura y contenido de los datos, y la sintaxis del lenguaje hace uso de la estructura de los datos. Si bien el modelo de datos global se puede definir usando una ontología de aplicación, ésta se debe diferenciar de las ontologías de dominio en el objetivo que persigue: un modelo de datos especifica la estructura y las reglas de integridad de los datos en una aplicación particular, mientras que una ontología describe la conceptualización de un dominio [33].

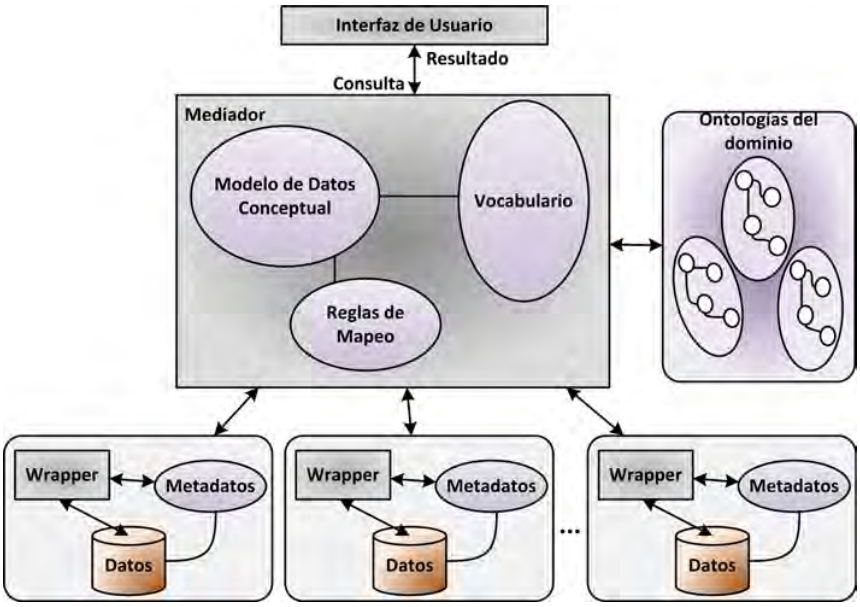

Fig. 2. Arquitectura de Integración Basada en Mediación

Por su parte, las ontologías de dominio son usadas para expandir la consulta, aplicando algunas de las técnicas que se proponen en [18, 2, 37]. La incorporación de los metadatos responde a una característica que se presenta con frecuencia en aplicaciones en el dominio médico, en las cuales es común encontrar datos no estructurados. Un ejemplo de ello, son las imágenes médicas y los campos de texto usados en reportes e historias clínicas. En estos últimos, el médico hace una descripción libre de la información relevante. Este caso se tiende a presentar cuando se hace referencia a antecedentes quirúrgicos, patológicos o familiares, o en la descripción del estado del paciente en el momento de la consulta. Para manejar la ambiguedad o la falta de semántica que se presenta en los datos no estructurados, la arquitectura provee la posibilidad de asociar metadatos, específicamente anotaciones a los datos. Para las imágenes médicas,

estos pueden ser descriptores de características de bajo nivel (color, forma, textura) o descriptores semánticos (conceptos de las ontologías) y para los campos de texto, son los conceptos de la o las ontologías apropiadas. Además, los metadatos permiten limitar el vocabulario de las consultas, ya que éste se restringe a los términos de las ontologías cuando se hace referencia a los datos de campos que tienen asociados metadatos.

Para realizar el procesamiento global de las consultas, el mediador cuenta con un modelo conceptual de datos que representa la información disponible en las diversas fuentes, un vocabulario asociado a este modelo, y la definición de los mapeos entre los elementos del modelo conceptual y los de las fuentes de datos. El vocabulario asociado al modelo conceptual permite que el usuario emplee términos equivalentes o sinónimos de aquellos usados en el modelo. Se propone que este conjunto de términos esté en un vocabulario anexo para evitar sobrecargar el modelo conceptual, ya que cuando el usuario formule su consulta podrá acceder a una vista simplificada del modelo, que le permitirá identificar las entidades que puede involucrar en la consulta.

Los componentes de esta arquitectura se detallan en las secciones 3.1, 3.2, y 3.3.

3.1 La interfaz de usuario

La interfaz de usuario recibe las consultas, que se especifican usando un lenguaje conceptual, cercano al usuario y que permite realizar diferentes tipos de consultas sobre diversos tipos de datos. Dicho lenguaje está basado en un modelo conceptual, que es independiente de la estructura de los datos y de los detalles de implementación.

El lenguaje de consulta ofrece al usuario un vocabulario y una sintaxis que le permiten expresar, a alto nivel, cuáles datos desea recuperar y cuáles características (o condiciones) deben satisfacer esos datos. Las consultas del usuario pueden incluir primitivas del lenguaje, términos de las ontologías de dominio y del vocabulario asociado al modelo conceptual, y constantes (e.g. el nombre o la edad de un paciente).

3.2 El mediador

El mediador es el componente encargado del procesamiento global de la consulta integrado por las siguientes etapas: análisis de la consulta (léxico, sintáctico y semántico), identificación de fuentes relevantes para responder a la consulta, generación de subconsultas, generación del plan de ejecución global y del plan de agregación de resultados, ejecución y agregación de resultados. La Fig. 3 muestra las etapas del mediador.

En la etapa de análisis, se identifican las primitivas del lenguaje, el vocabulario asociado al modelo conceptual, los términos de las ontologías, y las constantes numéricas o textuales que el usuario usó en la especificación de la consulta. Las primitivas definen el conjunto de operadores del lenguaje usados en la consulta, los términos del vocabulario asociado al modelo conceptual hacen

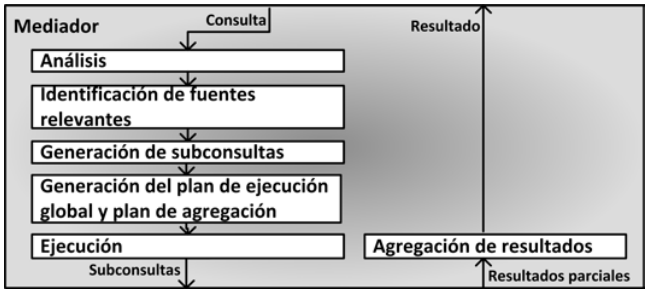

Fig. 3. Etapas del Mediador

referencia a la estructura de los datos, los términos de las ontologías hacen referencia a metadatos, y las constantes se refieren a datos. Teniendo en cuenta que es posible que en algunas consultas aparezcan términos que pueden tener más de una interpretación en el modelo conceptual o en las ontologías, será necesaria la intervención del usuario para precisar el término. Por ejemplo, la palabra "nombre" puede hacer referencia al nombre de un paciente, de un médico, o de un procedimiento. Para finalizar la etapa de análisis, el mediador genera un árbol de sintaxis compuesto por los operadores del lenguaje y los términos usados en la consulta. Esta representación de la consulta es más cercana al nivel lógico, e incluye detalles que el lenguaje conceptual abstrae, pero que se pueden agregar con base en la información del modelo de datos conceptual.

A partir del árbol de sintaxis, tres tareas se llevan a cabo: la identificación de las fuentes que tienen los datos que involucra la consulta, la generación de las subconsultas y la generación del plan de ejecución global y del plan de agregación de los resultados. Para realizar estas tareas, el mediador implementa un enfoque GAV (*Global As a View*) que hace uso del modelo de datos conceptual y de un conjunto de reglas de mapeo que definen cómo se obtienen las instancias de cada elemento del modelo conceptual a partir de la información de las diferentes fuentes de datos. El plan de ejecución global, en el caso más simple, consiste en enviar las sub-consultas a todas las fuentes para que se ejecuten en paralelo. Sin embargo, un plan de ejecución optimizado podría utilizar el resultado de una subconsulta como parámetro de otra, produciendose entonces ejecuciones secuenciales. En consonancia con el plan de ejecución global, se genera un plan de agregación de los resultados producidos por las fuentes de datos, que permite integrar la respuesta que se dá al usuario.

3.3 *Wrapper*-Metadatos-Fuente de Datos

El *wrapper* traduce la consulta enviada por el mediador al lenguaje que la fuente de datos es capaz de procesar. Para ello, usa los metadatos, el modelo de datos lógico de la fuente y un conjunto de reglas de traducción que permiten reescribir la consulta en términos del lenguaje que la fuente de datos reconoce. Además

el *wrapper* distingue los datos que tienen metadatos asociados y los que no los tienen, para procesarlos de forma diferente. En el primer caso, si los términos de la consulta hacen referencia a los metadatos, entonces el *wrapper* hace uso de éstos para identificar los datos que debe recuperar. En el segundo caso, el *wrapper* accede directamente a los datos en el repositorio.

4 Conclusiones y trabajo futuro

La arquitectura planteada en este artículo se construye como base para el desarrollo de un lenguaje conceptual de recuperación de datos e imágenes médicas. Este lenguaje busca proveer al usuario con una herramienta para especificar consultas complejas, con un lenguaje simple, y usando un vocabulario que le sea familiar. El lenguaje debe permitir recuperar datos de diversos tipos y de fuentes heterogéneas, de manera independiente de los detalles de las estructuras con las cuales se almacenan los datos o de las características particulares de cada fuente de datos. Por lo tanto, el lenguaje debe ser cercano al usuario, de alto nivel, con una sintaxis simple y un poder expresivo que permita formular consultas útiles en el campo médico. Para lograrlo, se pretende sacar ventaja del conocimiento del usuario, experto en el tema de la consulta.

Con este objetivo, se propone una arquitectura de integración basada en mediación, en la cual, la vista global de los datos se representa con un modelo conceptual, de tal forma que las consultas del usuario se formulen con base en esta vista. De esta manera, la especificación de las consultas no incluyen detalles de la estructura de los datos ni de las fuentes de datos. Asociado al modelo conceptual se cuenta con un vocabulario, con el que se pretende conservar el modelo de datos tan simple como sea posible, sin restringir el uso de un vocabulario amplio para referirse a los elementos del mismo. Además, tanto la ambiguedad como la falta de semántica se gestionan a través de un conjunto de metadatos con los que se anotan los campos no estructurados, tales como textos con descripciones libres o imágenes.

Esta arquitectura podrá ser mejorada con trabajos futuros enfocados, entre otros objetivos, a:

- Flexibilizar el mediador para permitir la actualización de los esquemas de las fuentes de datos, a través del mecanismo que permita definir los mapeos de forma que un usuario administrador del sistema, experto en tecnologías de información, pueda agregar o modificar reglas de mapeo, para incluir nuevas fuentes o actualizar las existentes de acuerdo con modificaciones en sus esquemas de datos locales.
- Incluir técnicas para optimizar el plan de ejecución global de la consulta.
- Automatizar la creación de metadatos a partir de los datos no estructurados y las ontologías.

References

1. Angles, R., Gutierrez, C.: Survey of Graph Database Models. ACM Computing Surveys 40(1), 1–39 (2008)

2. Bhogal, J., Macfarlane, A., Smith, P.: A Review of Ontology Based Query Expansion. Information Processing and Management 43(4), 866–886 (2007)
3. Blakeley, J.A., Campbell, D., Muralidhar, S., Nori, A.: The ADO.NET Entity Framework: making the conceptual level real. ACM SIGMOD Record 35(4), 32 (2006)
4. Calvanese, D., De Giacomo, G.: Data Integration: a Logic-based Perspective. AI Magazine 26, 59–70 (2005)
5. Calvanese, D., De Giacomo, G., Lenzerini, M.: Description Logics for Information Integration. In: Kakas, A., Sadri, F. (eds.) Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part II. LNCS, vol. 2408, pp. 41–60. Springer, Heidelberg (2002)
6. Chen, P.: The entity-relationship Model: Toward a Unified View of Data. ACM Transactions on Database Systems 1(1), 9-36 (1976)
7. El-Ghalayini, H., Odeh, M., McClatchey, R.: Developing Ontology-driven Conceptual Data Models. In: Proc. of the 1st Intl. Conf. on Intelligent Semantic Web-Services and Applications ISWSA'10, pp. 1–6. ACM, New York (2010)
8. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data Exchange: Semantics and Query Answering. Theoretical Computer Science 336(1), 89–124 (2005)
9. Fankam, C., Jean, S., Pierra, G., Bellatreche, L., Ameur, Y.: Towards Connecting Database Applications to Ontologies. In: 1st Intl. Conf. on Advances in Databases, Knowledge, and Data Applications DBKDA'09, pp. 131–137. IEEE Computer Society, Washington (2009)
10. Friedman, M., Levy, A., Millstein, T.: Navigational Plans for Data Integration. In: Proc. of the 16th National conf. on Artificial Intelligence and the 11th Conf. Innovative Applications of Artificial Intelligence AAAI'99/IAAI'99, pp. 67–73. American Association for Artificial Intelligence (1999)
11. Gea, M., Gutiérrez, F., Garrido, J., Cañas, J.: Teorías y Modelos Conceptuales para un Diseño Basado en Grupos. In: IV Congreso Intl. de Interacción Persona-Ordenador. España (2003)
12. Gupta, A., Bug, W., Marenco, L., Qian, X., Condit, C., Rangarajan, A., Muller, H.M., Miller, P.L., Sanders, B., Grethe, J.S., Astakhov, V., Shepherd, G., Sternberg, P.W., Martone, M.E.: Federated Access to Heterogeneous Information Resources in the Neuroscience Information Framework (NIF). Neuroinformatics 6(3), 205–217 (2008)
13. Gupta, A., Condit, C., Qian, X.: BioDB: an Ontology-enhanced Information System for Heterogeneous Biological Information. Data & Knowledge Engineering 69(11), 1084–1102 (2010)
14. Gupta, A., Ludascher, B., Martone, M.E.: BIRN-M: a Semantic Mediator for Solving Real-world Neuroscience Problems. In: Proc. of the 2003 ACM SIGMOD Intl. Conf. on Management of Data - SIGMOD'03, p. 678. ACM, New York (2003)
15. Halpin, T.: Conceptual Queries. Database Newsletter 26(3) (1998)
16. Halpin, T.: Object-Role Modeling (ORM/NIAM). In: Bernus, P., Mertins, K., Schmidt, G. (eds) Handbook on Architectures of Information Systems, pp. 81–103, Springer-Verlag, Heidelberg (1998)
17. Halpin, T., Bloesch, A.C.: Data modeling in UML and ORM: a Comparison. Journal of Database Management JDM 10(4), 4–13 (1999)
18. Harman, D.: Towards Interactive Query Expansion. In: Proc. of the 11th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval SIGIR'88, pp. 321–331. ACM, New York (1988)
19. Heath, C.: The constellation Query Language. In: Meersman, R., Herrero, P., Dillon, T. (eds) OTM 2009 Workshops. LNCS, vol. 5872, pp. 682–691. Springer, Heidelberg (2009)

20. Hevner, A., March, S., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
21. Karanikolas, N.N.: Conceptual Universal Database Language. In: Proc. of the 2009 Euro American Conf. on Telematics and Information Systems New Opportunities to increase Digital Citizenship EATIS'09, pp. 1–5. ACM, New York (2009)
22. Langegger, A.: Virtual Data Integration on the Web: Novel Methods for Accessing Heterogeneous and Distributed Data with Rich Semantics. In: Proc. of the 10th Intl. Conf. on Information Integration and Web-based Applications & Services, pp. 559–562. ACM (2008)
23. Lawley, M., Topor, R.: A Query Language for EER Schemas. In: ADC 94 Proc. of the 5th Australian Database Conf., pp. 292—304. Global Publications Service (1994)
24. Lenzerini, M.: Data integration: a Theoretical Perspective. In: Proc. of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems PODS'02, pp. 233–246. ACM (2002)
25. Martin, L., Anguita, A., Maojo, V., Bonsma, E., Bucur, A.: Ontology Based Integration of Distributed and Heterogeneous Data Sources in ACGT. In: Proc. of the First Intl. Conf. on Health Informatics HEALTHINF, pp. 301–306. Springer-Verlag (2008)
26. Martin, L., Bonsma, E., Anguita, A., Vrijnsen, J., Garcia-Remesal, M., Crespo, J., Tsiknakis, M., Maojo, V.: Data Access and Management in ACGT: tools to Solve Syntactic and Semantic Heterogeneities between Clinical and Image Databases. In: Hainaut, J., Rundensteiner, E.A., Kirchberg, Bertolotto, M. (eds) Advances in Conceptual Modeling Foundations and Applications. LNCS, vol.4802, pp. 24–33. Springer, Heidelberg (2007)
27. Moller, M., Sintek, M.: A Generic Framework for Semantic Medical Image Retrieval. In: KAMC, CEUR Workshop Proc (2007)
28. Mylopoulos, J.: Conceptual Modelling and Telos. In: Loucopoulos, P., Zicari, R. (eds) Conceptual Modelling, Databases and CASE: An Integrated View of Information Systems Development, pp. 49–68. John Wiley & Sons Inc., New York (1992)
29. Navathe, S.,B.: Evolution of Data Modeling for Databases. Communications of the ACM 35, 112–123 (1992)
30. Ovchinnikov, V.V.: Architecture of a Semantic Data Integration System Based on a Semantically Complete Model and a Semantically Complete Query Language. Programming and Computer Software 32(4), 228–242 (2006)
31. Owei, V.: Development of a Conceptual Query Language: Adopting the User-Centered Methodology. The Computer Journal 46(6), 602—624 (2003)
32. Roussopoulos, N., Karagiannis, D.: Conceptual Modeling: Past, Present and the Continuum of the Future. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds) Conceptual Modeling: Foundations and Applications. LNCS, vol. 5600, pp. 139–152. Springer, Heidelberg (2009)
33. Ruiz, F.J., Hilera, R.: Using Ontologies in Software Engineering and Technology. In: Calero, C., Ruiz, F., Piattini, M. (eds) Ontologies for Software Engineering and Software Technology, pp. 49–102. Springer, Heidelberg (2006)
34. Savinov, A.: Concept-Oriented Model and Query Language. Submitted to ACM Transactions on Database Systems TODS (2010)
35. Seng, J., Kong, I.: A Schema and Ontology-aided Intelligent Information Integration. Expert Systems with Applications 36, 10538–10550 (2009)
36. Sheth, A.P., Larson, J.A.: Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases. ACM Computing Surveys 22, 183–236 (1990)

37. Shiri, A., Revie, C.: Query Expansion Behavior within a Thesaurus-enhanced Search Environment: A User-centered Evaluation. Journal of the American Society for Information Science and Technology 57, 462–478 (2006)
38. Silberschatz, A., Korth, H.F., Sudarshan, S.: Data models. ACM Computing Surveys 28, 105–108 (1996)
39. Sonntag, D., Moller, M.: Unifying Semantic Annotation and Querying in Biomedical Images Repositories. In: Proc. of the Intl. Conf. on Knowledge Management and Information Sharing KMIS IC3K, 89–94 (2009)
40. Sonntag, D., Moller, M.: A Multimodal Dialogue Mashup for Medical Image Semantics. In: Proc. of the 15th intl. conf. on Intelligent User Interfaces IUI'10, pp. 381–384. ACM (2010)
41. Tian, H., Sunderraman, R., Calin-Jageman, R., Yang, H., Ying, Z., Katz, P.: NeuroQL: a Domain-specific Query Language for Neuroscience Data. In: Current Trends in Database Technology - EDBT. LNCS vol.4254, pp. 613 – 624. Springer, Heidelberg (2006)
42. Tsinakis, M., Doerr, M., Kondylakis, H., Martín, L., Anguita, A., Maojo, V., Bonsma, E., Bucur, A., Vrijnsen, J., Brochhausen, M., Cocos, C., Stenzhorn, H.: Ontology Based Integration of Distributed and Heterogeneous Data Sources in ACGT. In: Proc. of the 1st Intl. Conf. on Health Informatics HEALTHINF, pp. 301–306. Springer-Verlag (2008)
43. Ullman, J.D.: Information Integration Using Logical Views. In: Proc. of the 6th Int. Conf. on Database Theory ICDT'97, pp. 19–40. Springer-Verlag, London (1997)
44. Wiederhold, G.: Mediators in the Architecture of Future Information Systems. Computer 25, 38–49 (1992)
45. Wilkinson, M., Schoof, H., Ernst, R., Haase, D.: BioMOBY Successfully Integrates Distributed Heterogeneous Bioinformatics Web Services. The Planet Exemplar Case. Plant Physiology 138, 5–17 (2005)