Sistema Cooperativo para la Integración de Fuentes Heterogéneas de Información y Almacenes de Datos

José Samos[a] Alberto Abelló[b] Marta Oliva[c] Elena Rodríguez[b] Fèlix Saltor[b]
Jaume Sistac[b] Francisco Araque[a] Cecilia Delgado[a] Eladio Garví[a] Emilia Ruiz[a]

jsamos@ugr.es, aabello@lsi.upc.es, oliva@eup.udl.es, {malena, saltor}@lsi.upc.es, sistac@lsi.upc.es, {faraque, cdelgado, egarvi, eruiz}@ugr.es

[a] Dept. de Lenguajes y Sistemas Informáticos, Universidad de Granada
[b] Dept. de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya
[c] Dept. d'Informàtica i Enginyeria Industrial, Universitat de Lleida

Resumen

En este trabajo se presenta nuestra propuesta de creación de un prototipo de sistema cooperativo para la integración de fuentes heterogéneas de información y almacenes de datos en el cual se centran actualmente nuestras investigaciones. El objetivo general es proporcionar una capa de software que permita la cooperación entre diversas fuentes de información que están interconectadas mediante una red de líneas de comunicación. Cada fuente posee sus propios servicios de respuesta a preguntas que sobre sus datos realizan sus usuarios y, adicionalmente, se desea ofrecer a determinados usuarios la capacidad de acceder al conjunto de datos de una forma uniforme (acceso integrado), ya sea en tiempo real, ya sea a través de almacenes de datos.

Palabras clave: Sistema cooperativo, almacén de datos, heterogeneidad, autonomía, intercomunicación, seguridad, acceso integrado, clases derivadas, evolución de esquemas.

1 Introducción

Tanto la administración pública como el sector empresarial se encuentran, cada vez con más frecuencia, con múltiples sistemas de bases de datos a los que convendría acceder de un modo integrado, es decir, como si se tratase de una sola base de datos. Si el acceso no es integrado, hace falta acceder separadamente a cada uno de los sistemas de bases de datos (para lo cual hay que conocer qué datos contiene, en qué forma, y uno de sus lenguajes de acceso), y luego combinar las respectivas respuestas (pasándolas a un formato común y eliminando redundancias).

Además, esta creciente necesidad del acceso integrado no sólo se presenta para datos almacenados en sistemas de bases de datos, sino también para otras *fuentes de información*: hojas de cálculo, páginas de la World Wide Web, servidores públicos de información, y otros sistemas de información "cerrados", es decir, capaces de responder a ciertas preguntas pero que "esconden" el formato de sus datos y no permiten su acceso directamente.

Este acceso integrado puede hacerse a) *en tiempo real*, es decir, que la consulta se procesa accediendo directamente a las bases de datos preexistentes, que interoperan formando un *sistema cooperativo*, o bien b) consultando un *almacén de datos (Data Warehouse)*, en el que periódicamente se vuelcan y consolidan los datos de esas bases de datos. Nos proponemos investigar sobre estos temas en un proyecto conjunto de la Universitat Politècnica de Catalunya y la Universidad de Granada. Una característica **distintiva** e **innovadora** de nuestro enfoque es trabajar con una arquitectura que integra tanto el acceso en tiempo real como los almacenes de datos, y que pone las bases para acceder no sólo a bases de datos, sino también a otras fuentes de información. En otras palabras, un **sistema cooperativo para la integración de fuentes heterogéneas de información y almacenes de datos.**

En el apartado 2 se presenta la arquitectura general del sistema cooperativo, distinguiendo dos partes: la arquitectura de construcción y la arquitectura de ejecución. En el apartado 3 se amplía la arquitectura anterior para incorporar el concepto de almacén de datos. El apartado 4 corresponde a la descripción de varios puntos en los que estamos trabajando para implementar la arquitectura propuesta. Por último, las conclusiones son presentadas en el apartado 5.

2 Arquitectura del sistema

Para conseguir un acceso integrado en tiempo real hace falta superponer una capa de "software" a los respectivos sistemas de gestión de bases de datos (SGBDs) y demás fuentes, para constituir un sistema conjunto, que llamamos **sistema cooperativo** (o federado) **de fuentes de información**. Es entonces cuando podemos efectuar una sola consulta y obtener una respuesta única, porque es la *arquitectura* del *sistema cooperativo* quién toma a su cargo los aspectos derivados de que no se trata de una sola base de datos, sino de una red de bases de datos (BDs) y otras fuentes interconectadas. Ello tiene lugar en la *fase de operación*, que requiere previamente una *fase de construcción* para interconectar las fuentes de información mediante una red de líneas de comunicaciones, y para resolver los problemas de autonomía y heterogeneidad que aparezcan, y que se comentan a continuación.

2.1 Arquitectura de construcción

En un sistema cooperativo, las fuentes de información que lo componen habrán sido diseñadas independientemente, por lo que, en general, serán heterogéneas. Habrá **heterogeneidad** de "hardware" (ordenador central, discos, ordenadores de comunicaciones, etc.) y de "software" (sistema operativo, subsistema de comunicaciones, subsistema de gestión de bases de datos -SGBD-, etc.). Respecto a los datos, habrá heterogeneidad de contenido, de modelo de datos (relacional, orientado a objetos, etc.), de diseño, de control, etc. Más importante es la *heterogeneidad semántica* entre los datos de las distintas bases: qué clases de entidades (objetos) contienen, con qué estructuras, atributos –incluyendo codificación o unidad de medida, dominio–, operaciones (métodos), etc., así como la *heterogeneidad de los sistemas de seguridad* (discrecionales, obligatorios, basados en los "roles", etc.).

La superación de la heterogeneidad entre modelos de datos se consigue mediante un *Modelo Canónico de Datos* (MCD), que actúa de pivote al cual y desde el cual se traducen los demás modelos, y que debe tener construcciones que faciliten la resolución de la heterogeneidad semántica. En investigaciones previas ya concluimos que los *modelos orientados a objetos* son los más adecuados como modelos canónicos de un sistema cooperativo [7], y desarrollamos el modelo *BLOOM*, que utilizamos como modelo canónico de datos. De ahí nuestro interés en investigar sobre *clases derivadas* y *esquemas externos* en *Bases de Datos Orientadas a Objetos* (BDOO) (ver apartado 4.3).

Para resolver la heterogeneidad de los sistemas de seguridad se utiliza un modelo de seguridad propio del sistema cooperativo donde, además de tener en cuenta los criterios de seguridad locales de las distintas fuentes de información, se deben considerar los criterios de seguridad globales del sistema cooperativo. Este nuevo sistema de seguridad basa su mecanismo en la técnica de seguridad multinivel (sistema obligatorio), donde se clasifican los datos y se autorizan a los usuarios teniendo en cuenta un rango de niveles de seguridad (de la federación), de forma que un usuario podrá acceder a un dato determinado si ese dato está clasificado en un nivel igual o inferior al nivel autorizado del usuario, y lo podrá actualizar si el nivel de clasificación del dato coincide con el nivel de autorización del usuario.

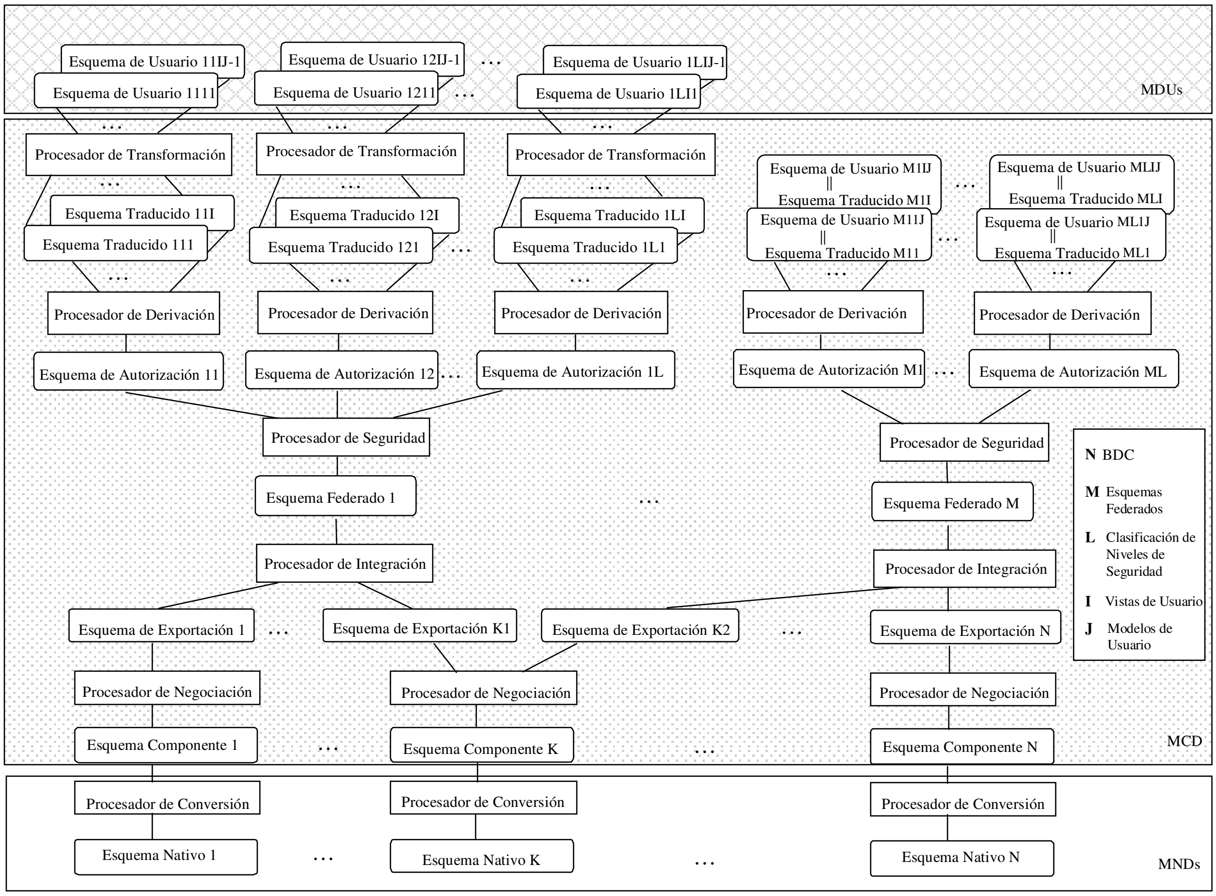

Figura 1. Arquitectura de construcción.

En nuestra arquitectura, además de los cinco niveles de esquema propuestos en el marco de referencia de [11], aparecen otros niveles: *Esquema de Autorización* para dar soporte a la protección de los datos, y *Esquema Traducido* para soportar modelos de datos de usuario (MDU) diferentes del canónico. Estos niveles se intercalan entre los del marco de referencia, dando lugar a siete niveles (ver fig. 1). Los diferentes esquemas son:

1. *Esquema Nativo*: es el esquema conceptual de una base de datos componente expresado en su modelo de datos nativo.
2. *Esquema Componente*: es la conversión de un Esquema Nativo en BLOOM (nuestro MCD).
3. *Esquema de Exportación*: representa la parte de un Esquema Componente que está disponible a un conjunto de usuarios federados.
4. *Esquema Federado*: es la integración de múltiples Esquemas de Exportación, cada Esquema Federado soporta una semántica.

5. *Esquema de Autorización*: representa el subconjunto de un Esquema Federado accesible por un usuario o grupo de usuarios con autorización en un nivel específico del rango de niveles de seguridad de la federación.
6. *Esquema Traducido*: define un esquema para un usuario y/o aplicación o un conjunto de usuarios/aplicaciones, está definido mediante el MCD.
7. *Esquema de Usuario*: es la conversión de un Esquema Traducido en el modelo de datos nativo del usuario.

Las *correspondencias* entre estos niveles de esquema se crean en la fase de construcción y se guardan en un *directorio* del sistema cooperativo, para su uso en la fase de operación (ver fig. 2). Dichas correspondencias se obtienen mediante un proceso ascendente llevado a cabo por los siguientes procesadores: *Procesador de Conversión*, *Procesador de Negociación*, *Procesador de Integración*, *Procesador de Seguridad*, *Procesador de Derivación* y *Procesador de Transformación* (ver fig. 1).

En primer lugar, cuando se forma una federación, se necesita usar un Procesador de Conversión para convertir Esquemas Nativos en Esquemas Componentes expresados en el MCD de la federación. Esta conversión incluye el enriquecimiento semántico de los Esquemas Nativos y la traducción de estructuras de datos. En este punto mantenemos la clasificación de los datos correspondiente a los rangos de niveles de clasificación pertenecientes a los sistemas de seguridad nativos (suponiendo que se basan en un sistema de seguridad multinivel), a la vez que se hace necesaria la clasificación de los elementos originados durante el enriquecimiento semántico. El objetivo del Procesador de Conversión es resolver las heterogeneidades sintácticas, corresponde al Procesador de Transformación de [11]. En la fig. 1 suponemos que las BDs componentes no usan el MCD como modelo de datos nativo. Si este fuera el caso, no sería necesario definir los Esquemas Componentes.

Una vez obtenidos los Esquemas Componentes, se inicia un proceso de negociación para alcanzar un acuerdo sobre la porción de datos que cada BD componente exporta a la federación. La negociación la realizan los administradores federal y local con la ayuda del Procesador de Negociación que soporta la autonomía de asociación, obteniendo como resultado una colección de Esquemas de Exportación.

A partir de los Esquemas de Exportación de diferentes BDs componentes se construyen uno o más Esquemas Federados. Este proceso se realiza por un Procesador de Integración que sugiere cómo integrar los Esquemas de Exportación ayudando a resolver heterogeneidades semánticas, proporcionando transparencia de distribución. El Procesador de Integración se sitúa al mismo nivel que el Procesador de Construcción en la arquitectura de referencia [11] pero, mientras que el Procesador de Construcción tan solo soporta la resolución de problemas relacionados con la heterogeneidad semántica de los datos, el Procesador de Integración también ayuda a solucionar heterogeneidades semánticas entre el rango de niveles de clasificación (número de niveles, semántica, etc.) que cada sistema nativo de seguridad multinivel tenga. La obtención de los Esquemas Federados y de Exportación no es un proceso lineal; se necesita que los Procesadores de Integración y Negociación colaboren en un proceso iterativo donde participen los administradores locales y federados [5].

Después de obtener los Esquemas Federados, el Procesador de Seguridad genera a partir de cada Esquema Federado (donde se tienen las estructuras de datos clasificadas según el rango de niveles de clasificación de la federación) un Esquema de Autorización para cada nivel perteneciente al rango de la federación al que pueda estar autorizado un usuario.

El Procesador de Derivación deriva a partir de cada Esquema de Autorización diferentes Esquemas Traducidos para especificar la parte de información en un Esquema de Autorización que es relevante para un grupo de usuarios federados y/o aplicaciones según sus necesidades de acceso.

Por último, cada Esquema Traducido potencialmente puede ser traducido desde el MCD de la federación al modelo nativo de un determinado usuario (o grupo de usuarios federados) dando lugar a varios Esquemas de Usuario. Si un usuario federado usa habitualmente el MCD, entonces su Esquema de Usuario sería directamente el correspondiente Esquema Traducido.

2.2 Arquitectura de ejecución

Cuando la federación ha sido construida, una arquitectura diferente, llamada *arquitectura de ejecución*, es la que entra en juego. La fig. 2 muestra esta arquitectura a un nivel *funcional*, representando todos los módulos funcionales y el flujo de datos desde la consulta original (*consulta federada*) a través de los módulos y los SGBDs hasta obtener el resultado final (*resultado federado*).

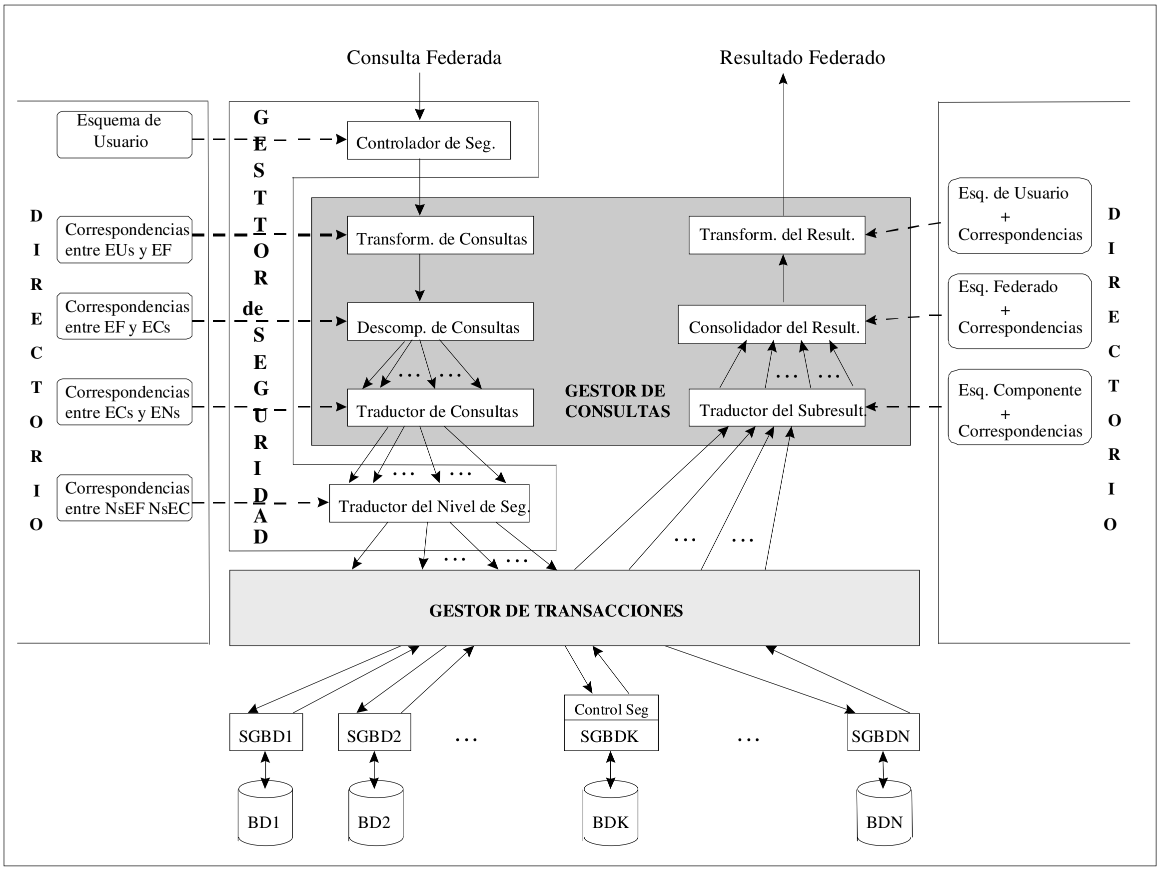

Figura 2. Arquitectura de ejecución.

En la fase de operación, en primer lugar se lleva a cabo la descomposición de la consulta en subconsultas a las distintas fuentes de datos y la traducción de las subconsultas a la forma de los datos y al lenguaje de cada fuente (acciones realizadas por el *Gestor de Consultas* con la colaboración del *Gestor de Seguridad*). El control de la ejecución de las subconsultas lo realiza el *Gestor de Transacciones*. Por último, la transformación de los subresultados, y combinación de los mismos para producir la respuesta consolidada al usuario es llevada a cabo de nuevo por el Gestor de Consultas. La fase de operación es presentada con mayor nivel de detalle en [6].

3 Construcción e integración de almacenes de datos (data warehousing)

Aunque el término "data warehouse" o "almacén de datos" es relativamente nuevo (se considera a Bill Inmon [2, 3] como el padre del "data warehousing"), el concepto representado no es nuevo en absoluto: Un *almacén de datos* es una BD que contiene una copia de datos de los sistemas operacionales con una estructura especialmente adecuada para realizar consultas y análisis (ofrece una visión histórica de los datos de los sistemas operacionales). El ámbito es uno de los elementos característicos de los almacenes de datos: Si su ámbito es toda la empresa lo llamaremos *data warehouse*. Se usa el término *data mart* para designar a los almacenes de datos cuyo ámbito es más reducido, normalmente un departamento o área específica dentro de la empresa.

En la fig. 3 se presenta un resumen de la arquitectura de construcción de almacenes de datos propuesta en [4]. Los datos procedentes de las aplicaciones, BDs componentes y otras fuentes de información son integrados en el *Almacén de Datos Operacionales* (ADO, "Operational Data Store"), que es una colección de datos que contiene datos detallados para satisfacer las necesidades operacionales de acceso colectivo e integrado de los datos de la corporación a la que sirve. Generalmente, a partir del ADO se define el data warehouse corporativo, aunque también existe la posibilidad de obtener los datos a cargar en el data warehouse directamente de las aplicaciones y BDs componentes. Los data marts son definidos a partir del data warehouse.

Es muy frecuente encontrar asociado el concepto de almacén de datos con su implementación sobre BD relacionales o bien multidimensionales (construidas sobre BD relacionales o bien con estrategias específicas de almacenamiento multidimensional de los datos). Sin embargo, consideramos que el uso de las BDOO puede resultar de gran utilidad para el modelado e implementación de los almacenes de datos.

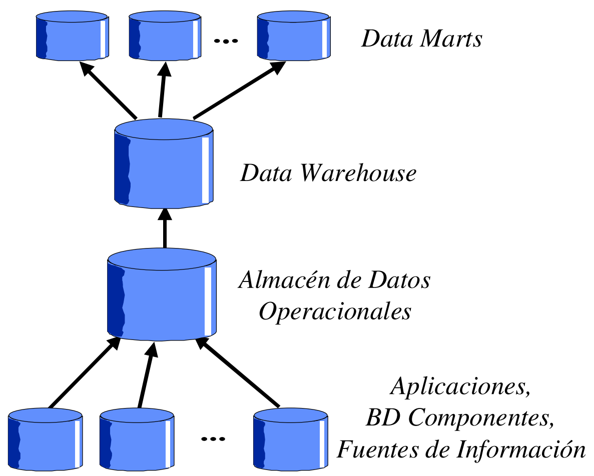

Figura 3. Arquitectura de construcción de almacenes de datos.

Aunque se reconocen importantes elementos en común entre los almacenes de datos y otras áreas previamente estudiadas, no se aprovechan en su totalidad los avances conseguidos en estas áreas para su aplicación a la construcción de almacenes de datos (data warehouses y data marts), en concreto, nos referimos especialmente a las BD federadas, BD temporales, así como a la definición de esquemas externos. Situando a los almacenes de datos en el contexto adecuado se puede subsanar esta situación: Éste es el objetivo de nuestro trabajo.

Como primer paso en esta línea de investigación [10], partiendo de la arquitectura para BD federadas definida dentro del proyecto BLOOM (ver sección 2), hemos definido una arquitectura integrada de BD que incorpora el concepto de data warehouse y de data mart. En esta arquitectura, **el data warehouse y los data marts forman parte de la BD**, son una funcionalidad más de la BD en lugar de ser sistemas independientes construidos utilizando una BD. A diferencia de otras propuestas, la nuestra permite que conceptos de BD federadas, temporales y orientadas a objetos, así como de definición de esquemas externos, puedan ser aplicados directamente a la definición y construcción de almacenes de datos. Según la arquitectura de BD definida, el esquema de los almacenes de datos se define usando el MCD (en este caso el modelo BLOOM, un modelo de datos orientado a objetos semánticamente rico); sin embargo, los usuarios pueden trabajar sobre los almacenes de datos usando otros modelos de datos, según sus necesidades.

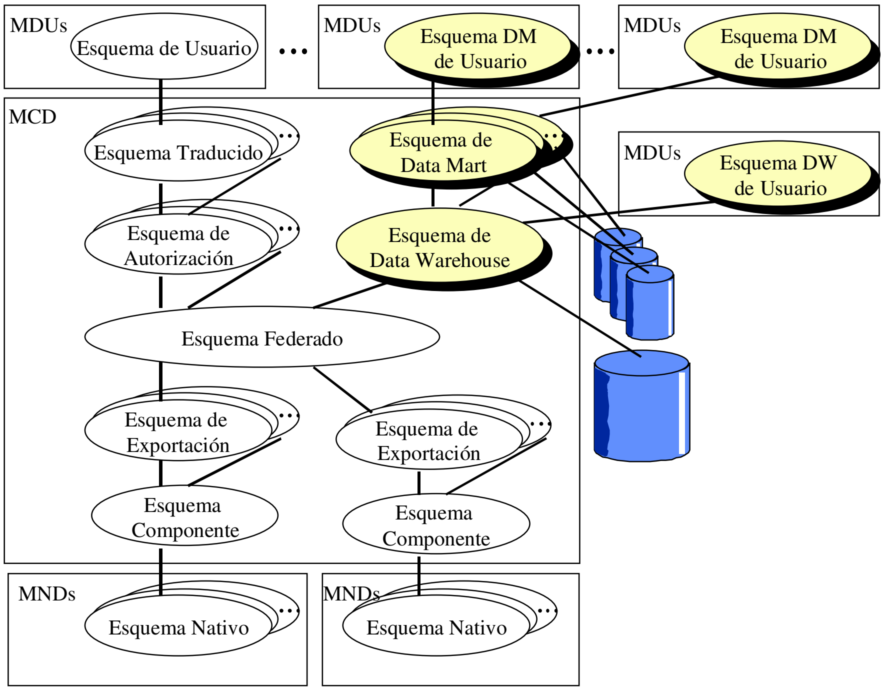

Figura 4. Arquitectura integrada de BD federadas y almacenes de datos.

En la fig. 4 se muestra nuestra arquitectura integrada considerando un solo esquema federado para simplificar la representación. El *esquema de data warehouse* y los *esquemas de data marts* se sitúan en paralelo con el resto de esquemas y pueden ser definidos a partir de cualquier esquema federado. El esquema federado juega un papel similar al del ADO en la arquitectura propuesta en [4], con la gran diferencia de que sus datos no están materializados. En caso de existir aplicaciones que requieran un acceso más óptimo a los datos, éstos pueden materializarse, definiendo el ADO a partir del esquema federado. El data warehouse puede ser definido directamente tanto a partir del esquema federado como del ADO y, a diferencia del esquema federado, el data warehouse y los data marts se encuentran siempre materializados (ya que, por definición, almacenan la historia de los datos obtenidos de de los sistemas operacionales).

En la defininción del *Esquema de Data Warehouse* se contemplan las características de estructuración y almacenamiento de los datos del data warehouse, asimismo se ofrece la posibilidad de considerar los distintos niveles de seguridad definidos. Los diferentes *Esquemas de Data Mart* son derivados a partir del Esquema de Data Warehouse permitiendo definir sus características particulares de estructuración y almacenamiento.

Los esquemas de Data Warehouse y de Data Marts son definidos usando el MCD de la federación; al igual que los Esquemas Traducidos, éstos pueden ser traducidos al modelo nativo de un usuario o grupo de usuarios federados, obteniéndose así los *Esquemas de Data Warehouse o de Data Mart de Usuario* respectivamente.

La arquitectura propuesta integra tanto el acceso en tiempo real como los almacenes de datos, el resultado es un **sistema cooperativo para la integración de fuentes heterogéneas de información y almacenes de datos.**

4 Trabajo futuro

En este apartado se revisan algunos de los elementos más relevantes en los que actualmente estamos trabajando o planeamos trabajar para llevar a cabo la implementación de la arquitectura integrada propuesta en los puntos anteriores.

4.1 Control de acceso cooperativo

Aunque nuestro sistema cooperativo base su mecanismo de protección de datos, y más concretamente de control de acceso, en un sistema de seguridad multinivel (por su propio funcionamiento también ayuda al control del flujo de la información), no sólo nos preocupa la integración de sistemas de seguridad multinivel heterogéneos entre si, sino que también pretendemos estudiar la forma de resolver las heterogeneidades para el caso de una fuente de información con un sistema de seguridad basado en un control de acceso o bien discrecional o bien basado en los "roles".

Como consecuencia de la incorporación del concepto de data warehouse y de data mart a la arquitectura para BD federadas, se hace imprescindible el estudio de cómo podría aprovecharse el sistema de seguridad de la federación para permitir sólo accesos autorizados a los almacenes de datos.

Desde el punto de vista de la seguridad también tienen importancia los cambios que se puedan producir debidos a la evolución de las fuentes de información, así pues será necesario analizar los distintos cambios que puedan darse así como la forma de incorporarlos en el sistema de seguridad del sistema cooperativo en funcionamiento.

4.2 Gestión de transacciones cooperativas

Una de las funciones básicas de un sistema gestor de base de datos convencional es la de permitir que múltiples usuarios puedan interactuar a la vez con el sistema para obtener datos almacenados en la base de datos, bien ejecutando programas de aplicación, bien ejecutando consultas interactivas. Estas ejecuciones se traducen en un conjunto de operaciones sobre unidades de datos que recibe el nombre de *transacción*. A su vez, las transacciones son la unidad de trabajo fundamental de un sistema gestor de base de datos. Esta función se puede trasladar a sistemas cooperativos de fuentes heterogéneas de información, teniendo en cuenta que además de los usuarios locales a cada fuente, aparece la figura del usuario global del sistema cooperativo que deseará obtener en tiempo real datos almacenados en las distintas fuentes que integran el sistema cooperativo.

Por lo tanto, los distintos sistemas de bases de datos continuarán funcionando *autónomamente* unos de otros, prestando sus servicios a los programas y usuarios preexistentes al mismo tiempo que soportan la ejecución de las subconsultas derivadas de los accesos integrados al sistema cooperativo. Ello supone nuevos problemas de control de concurrencia y recuperación, una nueva definición de “transacciones” del sistema cooperativo (diferentes de las transacciones convencionales de los SGBDs), y un *gestor de transacciones* a nivel cooperativo capaz de gestionarlas adecuadamente.

4.3 Definición de esquemas externos y clases derivadas en BDOO

En nuestra propuesta, la superación de la heterogeneidad entre modelos de datos se consigue mediante un MCD, el modelo BLOOM, un modelo orientado a objetos. De ahí nuestro interés en investigar sobre *clases derivadas* y *esquemas externos* en *Bases de Datos Orientadas a Objetos* (BDOO), ya que estos son los mecanismos en los que se basa el Procesador de Derivación definido en nuestra arquitectura.

La arquitectura de tres niveles de esquemas: *interno*, *conceptual* y *externo*, de ANSI/SPARC ha sido aplicada extensamente en las BD relacionales. En BDOO, los esquemas interno y conceptual han sido estudiados profundamente; no así los esquemas externos, a pesar de que la independencia lógica de datos que ofrecen es también un requerimiento para las BDOO. Además, tan solo unos pocos trabajos hacen referencia la arquitectura ANSI/SPARC y usan su terminología. En nuestros trabajos anteriores [8, 9] ha sido definida de forma preliminar una nueva metodología de definición de esquemas externos tomando como base la arquitectura ANSI/SPARC.

4.4 Evolución del esquema en BDOO

Un sistema cooperativo no se construye de una vez por todas y luego opera sin más: no puede ser *estático*, sino que debe adaptarse a la *dinámica* de los componentes que integra, como se recoge en el reciente *Cooperative Information Systems: A Manifesto* [1]. Los cambios en los esquemas son particularmente importantes, por lo que uno de nuestros objetivos es investigar sobre *evolución de esquemas*.

Los esquemas externos son derivados a partir del esquema conceptual de la BD, permiten presentar la información contenida en la BD de forma que se adapte mejor a las necesidades específicas de los usuarios finales en cuanto a su estructura y contenido. De cara a los usuarios finales, los esquemas externos pueden ser utilizados para simular algunos tipos de cambios en el esquema conceptual. La información en los esquemas externos ha de ser derivada a partir de la información en el esquema conceptual, por tanto, los cambios que los esquemas externos permiten simular son exclusivamente aquellos en los que toda la información del esquema resultante siempre se puede obtener a partir del esquema original: El esquema externo tiene la misma capacidad de información (las operaciones que lo producen son “capacity preserving transformations”) o menor capacidad de información (“capacity reducing transformations”) que el esquema conceptual [12].

En ocasiones, la necesidad de información de los usuarios finales aumenta y pasan a necesitar nueva información que no puede ser derivada a partir de la información existente en la BD (esquemas que son resultado de “capacity augmenting transformations”). En BDOO, esta situación se soluciona modificando el esquema conceptual mediante la definición de nuevas clases o bien la modificación de clases previamente existentes para que incorporen dicha información. En este caso no se dispone de un medio que permita simular los cambios, sino que éstos afectan directamente al esquema conceptual. Por tanto, se dispone de menor flexibilidad que en el caso anterior, ya que realizar cambios en el esquema conceptual es más costoso que realizarlos exclusivamente en algún esquema externo. Ésta es una limitación de los sistemas actuales sobre la que trabajamos para subsanarla.

5 Conclusiones

En este artículo hemos presentado un enfoque que soporta la construcción de almacenes de datos y el acceso a un conjunto a BDs que forman un sistema federado de BDs. El resultado de nuestra propuesta es una arquitectura innovadora que da lugar a un **sistema cooperativo para la integración de fuentes heterogéneas de información y almacenes de datos.**

Agradecimientos

El trabajo de los miembros de la Universitat Politècnica de Catalunya y de la Universitat de Lleida ha sido parcialmente apoyado por el programa PRONTIC, bajo el proyecto TIC96-0903, así como por una beca del Comissionat per Universitats i Recerca de la Generalitat de Catalunya (1998FI00228).

Referencias

[1] G. De Michelis, E. Dubois, M. Jarke, F. Matthes, J. Mylopoulos, M. Papazoglou, K. Pohl, J. Schmidt, C. Woo, E. Yu. “Cooperative Information Systems: A Manifesto.” In M. Papazoglou, G. Schlageter (Eds.) *Cooperative Information System: Trends and Directions*, Academic Press, 1997.

[2] W. Inmon. *Building the Data Warehouse*. Wiley-QED, 1992.

[3] W. Inmon. “What is a Data Warehouse?.” *PRISM Tech Topic*, Vol.1, No.1, 1992.

[4] W. Inmon, C. Imhoff, R. Sousa. *Corporate Information Factory*. John Wiley & Sons, Inc. (1998).

[5] M. Oliva, F. Saltor: "A Negotiation Process Approach for Building Federated Databases". In: 10th ERCIM Database Research Group Workshop on Heterogeneous Information Management, Prague 1996, ERCIM-96-W003 CRCIM, 1996, pp 43-49.

[6] E. Rodriguez, M.Oliva, F. Saltor, B. Campderrich: “On Schema and Functional Architectures for Multilevel Secure and Multiuser Model Federated DB Systems”. In S. Conrad et al. (eds), *Proc. of the Int’l CAiSE'97 Workshop on Engineering Federated Database Systems* (EFDBS'97), pp 93-104. Otto-von-Guericke-Universitat Magdeburg, Fakultat fur Informatik, 1997.

[7] F. Saltor, M. Castellanos & M. Garcia-Solaco: “Suitability of Data Models as Canonical Models for Federated DBs.” *ACM SIGMOD Record vol 20*, #4, pp. 44-48 (special refereed issue: A. Sheth (guest ed.): Semantic Issues in Multidatabase Systems, Dec. 1991).

[8] J. Samos. “Definition of External Schemas in Object-Oriented Databases.” In *Proc. 2nd. Int’l Conf. on Object Oriented Information Systems* (OOIS'95, Dublin), 1995, pp. 154-166.

[9] J. Samos, F. Saltor, J. Sistac. “Definition of Derived Classes in OODBs.” In B. Eaglestone, B. Desai, J. Shao (eds.): *Proc. of 1998 Int’l Database Engineering and Application Symposium* (IDEAS98, Cardif, July 1998). IEEE-CS Press, pp 150-159.

[10] J. Samos, F. Saltor, J. Sistac & A. Bardes: “Database Architecture for Data Warehousing: An Evolutionary Approach”. In *Proc. 9th Int’l Conf. on Database and Expert Systems Applications* (DEXA98, Viena, August 1998), pp 746-756.

[11] A. Sheth, J. Larson. “Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases.” *ACM Computing Surveys, Vol.22*, No.3, (Sept. 90).

[12] M. Tresch, M. Scholl, “Schema Transformation without Database Reorganization,” *SIGMOD Record*, vol. 22, no. 1, pp. 21-27, 1993.