

# Consulta de Bases de Datos Relacionales y Documentos XML: Estado del Arte y una Nueva Propuesta

Ana Feroso García  
*Universidad Pontificia de Salamanca, Facultad de Informática*  
*Salamanca (España)*  
[aferoso@upsa.es](mailto:aferoso@upsa.es)

María José Gil Larrea  
*Universidad de Deusto, Facultad de Ingeniería (E.S.I.D.E.)*  
*Bilbao (España)*  
[marijose@eside.deusto.es](mailto:marijose@eside.deusto.es)

Luis Joyanes Aguilar, Vidal Alonso Secades, Roberto Berjón Gallinas, Alfonso López Rivero,  
Marcelo Vallejo García  
*Universidad Pontificia de Salamanca, Facultad de Informática*  
*Salamanca (España)*  
[ljoyanes@fpablovi.org](mailto:ljoyanes@fpablovi.org), [vidalsec@upsa.es](mailto:vidalsec@upsa.es), [rberjon@upsa.es](mailto:rberjon@upsa.es), [lrivero@upsa.es](mailto:lrivero@upsa.es),  
[mvallejo@upsa.es](mailto:mvallejo@upsa.es)

Pablo de la Fuente Redondo  
*Universidad de Valladolid, Departamento de Informática*  
*Valladolid (España)*  
[pfuente@infor.uva.es](mailto:pfuente@infor.uva.es)

## Resumen

Actualmente gran cantidad de información está almacenada en Bases de Datos Relacionales, pero otra gran cantidad lo está en otros formatos, entre los que destaca XML, que se está convirtiendo en el estándar más utilizado para el intercambio de datos en el mundo de la web y los negocios a través de Internet.

Se hace imprescindible trabajar conjuntamente con diversos formatos de información, lo que conlleva una serie de problemas y limitaciones. Para paliarlos una posible solución podría ser un sistema que sirviese para consultar de forma transparente varios formatos, en particular entre SQL y XML, un Modelo de Consulta Integrada. Además, por el número creciente de aplicaciones que exige que la información tenga que ser tratada en un entorno Web, el sistema podría utilizar un lenguaje de consulta basado en XML, y que las respuestas a estas consultas también se devolviesen en el mismo formato XML.

En este trabajo, a partir de un detallado estudio de distintos sistemas existentes en los que se relacionan las Bases de Datos y XML, se estudian las limitaciones de dichos sistemas y se plantean las características que debería tener uno que trate de solventar dichas limitaciones.

**Palabras clave:** XML, Bases de Datos Relacionales, Consulta e Integración de datos en el entorno Web, Lenguajes de Consulta en XML.

## 1. Introducción

La historia de XML está íntimamente ligada a la evolución del World Wide Web [Bray 1998], de hecho XML se ha convertido en el estándar de facto para el intercambio y representación de

datos en el mundo los negocios a través de Internet: comercio electrónico, aplicaciones Bussines to Bussines (B2B), Bussines to Client (B2C) y el resto de variantes.

Por otra parte, a la hora de manejar grandes volúmenes de información, siguen siendo los Sistemas de Gestión de Base de Datos la herramienta más utilizada, y entro ellos el modelo más extendido sigue siendo el Relacional, aunque el modelo Orientado a Objetos vaya tomando fuerza día a día.

En consecuencia, resultaría muy útil poder manejar las Bases de Datos Relacionales, a través de XML, es decir, que podamos consultar las BD a través de un lenguaje basado XML y también utilizar este lenguaje como interfaz para pasar la información entre diferentes bases de datos o sistemas en general, que interaccionan entre sí.

En algunos casos, se puede pensar incluso en la sustitución de las bases de datos por XML como almacén de datos. Esto sólo resultaría útil en ciertas circunstancias [Morrison et al. 2000], por ejemplo cuando la representación de la información sea compleja, e incluso variante, o cuando los datos no sufren muchos procesos de actualización.

En todos los casos anteriores puede resultar ser más eficiente usar XML que una base de datos relacional para almacenar la información, pero en el resto de situaciones sigue siendo mejor continuar con el almacenamiento en bases de datos.

Por otra parte, en ciertos entornos se está convirtiendo en habitual tener que manejar dentro del mismo negocio, tanto datos almacenados en las bases de datos tradicionales, como en documentos XML. Esto sucede por ejemplo en aplicaciones B2B en las que, por un lado, hay que implementar procesos de negocio con la información almacenada en una base de datos relacional , y por otro, se tienen que integrar aplicaciones y negocios con los de otras empresas. Para asegurar la interoperabilidad entre ellas, por ejemplo por que tengan que intercambiar información entre sí siguiendo unos determinados estándares, se suele utilizar XML.

En consecuencia, se trata de establecer un método que sirva como sistema de consulta tanto para información almacenada en cualquier base de datos relacional, independientemente de su complejidad, tamaño y área o tema al que pertenezca la información, como para información almacenada en XML; es decir, que permita consultar información de la misma forma, independientemente del formato de almacenamiento real de esos datos. Para el usuario final este proceso debe ser transparente, como si sólo estuviera accediendo a información en XML.

Una de las premisas fundamentales del método que se propone, y que lo diferencia de otros sistemas de este tipo, es que los datos seguirán almacenados en su formato original, es decir, en las bases de datos o en XML, según donde estuvieran, no siendo necesario migrar a XML, en el

caso de las bases de datos, para poder ser consultados, independientemente de que los resultados de las consultas, e incluso las consultas, si que se hagan en XML. En el caso de las bases de datos relacionales, el método tiene otra ventaja, como al final las consultas se hacen directamente sobre la base de datos y en el lenguaje de consulta tradicional SQL, aunque todo esto resulte transparente al usuario, no se pierde eficiencia ni velocidad a la hora del acceso, ya que se seguirán utilizando para ello los motores de búsqueda de la propia base de datos.

Existen muchos trabajos que tratan de relacionar bases de datos relacionales y XML, y más concretamente, la consulta de información en uno u otro formato. El sistema planteado se puede utilizar también para consultar documentos XML. Esto último también es una novedad, pues la mayor parte de estos sistemas de consulta a través de XML, o servirán para consultar base de datos, previo volcado al formato XML, o para consultar datos en XML directamente, pero no para ambas al mismo tiempo y con la misma filosofía.

Lo importante será por tanto establecer las bases teóricas y conceptos fundamentales que permitan crear el modelo de integración en el que se apoye el sistema de consulta, este será el primer objetivo. Una vez logrado, se sentarán las bases para especificar el proceso de traducción de las consultas de usuario al lenguaje de la base de datos.

## **2. Estado del Arte: XML y las Bases de Datos Relacionales**

En casi todos los trabajos e investigaciones que relacionan las bases de datos Relacionales con XML se pretende hacer accesibles los datos de la base de datos, así como transferirlos a otros sistemas, a través del formato XML, pero en casi ninguno se permite que además el mismo método haga también accesibles los datos que ya están en XML. Algunos como Shanmugasundaram [Shanmugasundaram et al. 1999] hacen justo lo contrario, transformar documentos XML a bases de datos para luego hacer la consulta sobre esa información en un lenguaje como SQL, propio de las bases de datos relacionales.

En otros como el denominado Monet [Schmidt et al. 2000], se plantea un modelo de datos y ejecución para permitir un almacenamiento y devolución eficiente de documentos XML en una base de datos relacional.

Los documentos XML se representan en forma arborescente y la cuestión clave es cómo almacenar los datos de este árbol como una instancia de base de datos que suministre unas capacidades de acceso eficientes. Para conseguirlo se aplican una serie de reglas que permiten almacenar los datos del árbol (ramas o enlaces, nodos y valores finales del documento XML, que están en las hojas del árbol) en forma de tablas de la base de datos, y luego sobre estas

tablas se hacen consultas normales en SQL, que devuelven como resultado la información de los documentos XML consultados.

En X-Ray [Kappel 2000] lo que se plantea es un modelo o metaesquema que permite tanto pasar datos almacenados en una base de datos relacional a documentos XML que siguen un determinado Data Type Definition (DTD), como lo contrario, que datos almacenados en XML siguiendo un determinado formato o DTD, se almacenen en una base de datos relacional, es decir, sirven para migrar datos de XML a base de datos como antes, y al revés. El metaesquema de X-Ray está formado por tres componentes, el *DBSchema*, que contiene la información del esquema de la BD, el *XMLDTD*, que contiene la información del DTD de los documentos XML y el *XMLDBSchemaMapping* (esquema de correspondencias), que contiene la información para migrar la base de datos a un documento XML y viceversa. Sin embargo, esta migración de la base de datos no tiene porque ser sólo a un determinado DTD, es decir, a un determinado tipo de documento XML, pero sí que tiene que existir un esquema de correspondencias y un esquema con la información del DTD, *XMLDTD*, para cada DTD distinto. Por otra parte, también puede ocurrir que quieran convertirse diferentes bases de datos, cada una con su esquema, al mismo DTD, por ejemplo porque pertenecen al mismo dominio de información, en este caso también es necesario un *DBSchema* y un esquema de correspondencias, para cada base de datos.

El *DBSchema* contiene información sobre cada una de las relaciones de la base de datos, *DBRelations*, de los atributos que las componen, *DBAttributes* y sobre conexiones entre relaciones, *DBRelationShip*.

El *XMLDTD* guarda información sobre el DTD al que se quiere migrar a través de sus elementos, *XMLElementType*, y los atributos de estos elemento, *XMLAttributes*. Los elementos pueden ser simples, compuestos o vacíos.

Finalmente, el *XMLDBSchemaMapping*, establece las correspondencias entre los dos esquemas anteriores, tal que por ejemplo, un atributo de la base de datos puede estar asociado a un elemento simple, a otro atributo, o con el contenido de un elemento compuesto, del documento XML resultante.

En este modelo la correspondencia entre XML y la base de datos relacional, gira en torno al establecimiento de una relación base, que es la que se va a tomar como raíz del árbol, y que va a condicionar el cómo se ve el resto de la base de datos desde esta relación, y por tanto esto supondrá una limitación.

En general XML es ante todo una forma de almacenar y modelar datos. Así Peter McBrien y Alexandra Polovassilis [McBrien y Poulouvasilis (1) 1999] [McBrien y Poulouvasilis (2) 1999] [McBrien y Poulouvasilis 2001] en trabajos relacionados con el tema, utilizan un formato gráfico especial denominado *HDM (Hypergraph Data Model)* formado por tres elementos: nodos, arcos y restricciones. El manejo de este grafo y sus elementos lo hacen a través de una serie de primitivas que permiten consultar, añadir o eliminar elementos de dicho gráfico.

Estableciendo la correspondencia entre los elementos de cualquier modelo de datos (DER, UML, Modelo Relacional, ...) y los elementos de *HDM*, se pueden modificar los datos; es decir, se puede pasar de unos modelos a otros usando *HDM* como intermediario. Esto supone, que también a través e *HDM* se puede pasar del Modelo XML al Relacional y viceversa. Para ello parte de un esquema inicial de un determinado modelo de datos, y se quiere obtener otro distinto en otro modelo, y siempre que ambos pertenezcan al mismo universo de datos. Esto se consigue renombrando nodos y arcos y añadiendo y/o borrando nodos y arcos del grafo origen para obtener el de destino. Una vez pasado de un modelo de datos a otros, o incluso estando en el mismo modelo, por ejemplo porque lo que se quiere es adaptar la información de un esquema a otro del mismo universo y modelo de datos, se trata de migrar todas las operaciones, por ejemplo las consultas o actualizaciones, que se hacían sobre el sistema original y que queremos que se sigan haciendo sobre el destino.

Christophides y Cluet [Christophides y Cluet 2000] proponen utilizar XML como herramienta de integración en un sistema de conversión de datos semiestructurados denominado Yat, sobre los que utilizan un lenguaje de consulta especial, el YatL.

Este sistema integrador Yat, representa la información a varios niveles: (Modelo, Esquema y Datos) y está formado por unos wrappers y mediadores que se comunican a través de estructuras, datos y operaciones que utilizan una sintaxis XML.

Las consultas hechas en lenguaje YatL utilizan tres tipos de sentencias: *MATCH*, que asocia una variable a cada campo que participa en el filtro de la consulta, *MAKE*, que construye el resultado y *WHERE*, que añade las condiciones.

Además añaden al sistema una serie de funciones algebraicas en XML, lo que se conoce como el álgebra Yat, que se suman a las operaciones de álgebra relacional tradicionales, como *Select* y *Join*, que siguen soportándose igual. Estas tres nuevas operaciones algebraicas son:

- *BIND*: obtiene la estructura de las tablas que se representa en forma arborescente.
- *TREE*: selecciona ciertos subárboles de la estructura obtenida anteriormente.

- *Funciones Skolem*: sirve para crear nuevos identificadores, una especie de campos calculados

Aunque la representación de las operaciones es arborescente, también se incluyen herramientas para ejecutar consultas tanto en YatL, como en XML.

SINGAPORE [Dittrich y Domenig 1999] [Dittrich y Jonscher 2000] es otro sistema que va más allá de las bases de datos relacionales, al proponer un método de integración y consulta tanto para datos estructurados de distintos tipos, bases de datos relacionales u orientadas a objetos, como para datos semiestructurados. Para conseguir esta integración usa una estructura especial denominada mediador, donde se implementa un modelo de datos global que permite homogeneizar toda la información. La consulta se hace en un lenguaje especial, SOQL, basado en SQL, pero más potente. Para crear el metaesquema en el que guardar la información de cualquier fuente de datos que se desee integrar en el sistema, se utilizan conceptos ontológicos, estudio de perfiles de usuario, etc.

SilkRoute [Fernández y Tan 2000] es una herramienta que permite consultar en XML una base de datos, con un lenguaje propio, el RXL.

Con la primera consulta RXL se crea una vista de parte de la base de datos, sólo de la afectada por esa consulta, con los datos ya en XML. Es decir, a XML no se pasan todos los datos de la base de datos, sino sólo los que interesan. Esto limita las posteriores consultas de usuario a esa vista virtual de la base de datos (*XML View*).

La arquitectura y procesamiento de una consulta en este sistema, se podría resumir de la siguiente forma:

Con la consulta RXL se crea la *XMLView* de la base de datos, que es lo que ve el usuario. Las consultas siguientes se hacen en XML-QL, se pasan al “Componedor de Consultas”, que en función de la *XMLView*, obtiene la “Consulta Ejecutable” o compuesta, otra vez en RXL. Esta consulta será la combinación de la consulta inicial RXL que da lugar a la *XMLView* y la emitida por el usuario en XML-QL. Una consulta RXL tiene una parte *from* y otra *where* como las de SQL y una parte *construct* como la de XML-QL.

Finalmente, la consulta ejecutable es convertida, por el “Traductor”, a una o más sentencias SQL que son ejecutados sobre la base de datos real por el motor de BD y los resultados obtenidos se traducen a XML por el “Generador XML”

M. Carry, D. Florescu et al. [Carry et al. 2000] en su proyecto XPERANTO permiten consultar datos almacenados en bases de datos Objeto-Relacionales en un lenguaje de consulta XML y devuelven los resultados también en XML, es decir, todo se hace en un formato XML puro.

El sistema se subdivide en una serie de componentes:

- Traductor de consultas, que es el componente más importante.
- Traductor de los resultados de la consulta de la base de datos a XML.
- Generador del esquema XML (XMLSchema), a partir del esquema de la base de datos.
- Servicios de la XML-View, que nos ayudan a obtener el XQGM (Modelo gráfico de consulta en XML) a partir de una consulta en XML-QL y la información del XMLSchema.

El Traductor de Consulta, el componente más importante, a su vez se divide en las siguientes partes: el parser XML-QL, que recibe la consulta XML-QL y obtiene el gráfico XQGM a partir de la información del XMLView. El Optimizador de la Consulta, que obtiene el XQGM ya optimizado para la consulta concreta realizada. El Traductor a SQL, que traduce a SQL la consulta reflejada en el XQGM que se recibe.

Una base de datos Objeto-Relacional se caracteriza porque además de los conceptos típicos de una base de datos Relacional como tablas, vistas, esquemas, columnas, tipos básicos de datos, añade otros conceptos propios de la Orientación a Objetos como herencias, referencias, “tipos” de tablas y vistas, ...

El proceso de traducción del esquema de la base de datos a un XMLSchema de XML, que sería el XMLView de esa base de datos, sería el siguiente:

- Los tipos estructurados de la base de datos objeto-relacional se convierten a “ComplextypeElement” en el XMLSchema.
- Las tablas normales de la BD se convierten a “element” en el XMLSchema.
- Para determinar que una tabla es un conjunto de elementos de un tipo estructurado, también se usa un “ComplexDataType”, diciendo que esa tabla esta formada por un conjunto de ocurrencias (maxoccurs=’\*’) del tipo estructurado.

Para pasar la consulta realizada de XML-QL a SQL se usa como elemento intermedio el gráfico XQGM, que facilita la traducción a SQL al optimizar el XML-View en función de la consulta XML-QL concreta realizada.

Este gráfico se parece al Modelo Gráfico de Consulta (QGM) del sistema de base de datos objeto-relacional de DB2 UDB, aunque se puede usar luego en otros sistemas. EL gráfico optimiza la información del XML-View para adaptarlo a la consulta XML-QL realizada, para lo cual se elimina del XML-View los elementos o atributos que no van a aparecer en el resultado de la consulta porque no se piden en la consulta real realizada.

Finalmente, para pasar a SQL la consulta final, y devolver los resultados en el orden que interesa para XML, cada subconsulta WHERE de la consulta XML-QL, se convierte a una subconsulta SQL y el resultado es la UNION de todas las subconsultas SQL. Para que quede clasificado como interesa en el XML-Result, se usa el ORDER-BY.

El orden de las subconsultas es desde la tabla principal de la consulta, a las secundarias, usando como claves las que se han obtenido en las tuplas devueltas de la tabla principal.

Además de los trabajos mencionados también han aparecido otros que desarrollan una serie de lenguajes de consulta, que se van convirtiendo más o menos en estándares según el éxito de cada uno, y que sirven para acceder a información que ya está en XML, aunque originalmente puede provenir por ejemplo de una base de datos relacional.

Maier presenta en [Maier 1998] algunos de los requisitos que debería tener cualquier lenguaje de consulta para XML. Entre ellos destacan: la presentación del resultado en XML, la posibilidad de seleccionar, extraer, combinar y transformar el contenido del documento XML (en este caso basta con que lo pueda consultar, no es necesario que también lo pueda modificar), la capacidad de actuar sobre datos sin esquema predefinido (en el modelo planteado si se va a obligar a que el documento este validado contra un determinado DTD o Esquema) y la posibilidad de que los resultados sean representados en XML.

M. Fernández, J. Simeon y P. Wadler [Fernández et al. 1999] también presentan las características que debería tener un lenguaje de este tipo y además ponen diversos ejemplos y comparan algunos de dichos lenguajes entre sí. Algunos de estos lenguajes proceden del entorno de BD y tienen más similitudes con otros lenguajes de este entorno como SQL. Dentro de este tipo está Lorel, XML-QL, YatL o Quilt. Otros lenguajes de consulta proceden del entorno de XML, como XSL, XQL o XQuery, y otros son lenguajes gráficos como XML-GL [Ceri et al. 1999].

XQuery [DeRose 1998] es un lenguaje que tiene influencias de lenguajes procedentes del entorno de BD como Quilt, que a su vez también está influenciado por YatL y Lorel, y del propio SQL, utilizando el típico patrón de sentencia SELECT-FROM-WHERE. Además está influenciado también por otros lenguajes de consulta del entorno XML, como XQL y XPath para la sintaxis de caminos jerárquicos, y de XML-QL para crear variables de enlace. Por todos estos motivos podría llegar a convertirse en el lenguaje estándar de consulta XML, el problema es que, a día de hoy, todavía sigue siendo un lenguaje de laboratorio que se sigue refinando y para el que aún no existen herramientas de implementación.



Uno de los lenguajes más utilizados es Lorel [Abiteboul et al. 2000] [McHugh 1997] [Abiteboul et al. 1997], que tiene influencias de OQL, un lenguaje de consulta para bases de datos orientadas a objetos y añade nuevas características como la búsqueda por aproximación (exclusivas de documentos semiestructurados como los de XML). Lorel usa los *Dataguide* imitando al esquema de una base de datos.

XML-QL [Deutsch 1999] es otro lenguaje proveniente del entorno de las bases de datos, también muy utilizado como Lorel. Una consulta consiste en una cláusula *WHERE*, que indica lo que se quiere seleccionar, y una cláusula *CONSTRUCT*, que especifica el resultado a devolver.

Quilt es otro lenguaje de consulta a documentos XML que suelen contener información procedente de base de datos relacionales, y con influencias de otros lenguajes como Lorel, Xpath, ... En [Manolescu et al. 2001] se dan las pautas para convertir una consulta en Quilt sobre un documento XML a SQL, y que sea ejecutada ya sobre la base de datos asociada al documento XML consultado. Sin embargo, en este trabajo no se da ninguna solución al proceso previo de traducción de la información de la base de datos relacional a XML. Para conseguirlo se podría usar cualquiera de los métodos propuestos para lograr esta traducción, por ejemplo SilkRoute, si sólo se quiere tener una vista de la base de datos, o X-Ray si se quiere pasar toda la base de datos a XML. En cualquier caso, se sigue partiendo de que la consulta se hace sobre datos ya en XML.

Los lenguajes XSL y XQL [Robie 1999], que proceden del entorno XML, se basan en la utilización de caminos o *path* para indicar lo que se busca, y filtros para establecer las condiciones de la búsqueda.

Otro método de acceso a las bases de datos relacionales es a través de páginas web con ASP (de Microsoft), JASP (en JAVA) o PHP (las más genéricas). Todas siguen la misma filosofía aunque utilicen distintos lenguajes de implementación. Este método permite consultar las BD a través de SQL, el lenguaje habitual de consulta de las bases de datos, y después convierten a XML los resultados devueltos en un “recordset”, pero no permiten hacer consultas en lenguaje XML, ni a documentos XML.

Por otro lado, algunos de los sistemas gestores de base de datos más importantes como Microsoft SQL-Server o Oracle9i, disponen ya de herramientas que permiten hacer consultas a las bases de datos en SQL y devuelven los resultados en XML.

## 2.1. Esquema Comparativo de los Diferentes Sistemas que Relacionan XML y Bases de Datos Relacionales.

Los trabajos expuestos, sistemas que relacionan XML y bases de datos relacionales, se pueden clasificar en cuatro categorías, los que permiten hacer conversiones entre distintos formatos, de XML a base de datos y viceversa. Los que además de permitir la conversión, incorporan un lenguaje para consultar los datos ya en XML, los que analizan una serie de lenguajes de consulta de información almacenada en XML y finalmente, los métodos que permiten hacer consultas a la base de datos en SQL, pero que transforman y devuelven los resultados en formato XML.

### 1. SISTEMAS DE CONVERSIÓN DE INFORMACIÓN ALMACENADA EN DISTINTOS FORMATOS.

Dentro de esta categoría se distinguen dos tipos:

- a) *Sistemas que permiten convertir información en XML a Base de Datos.* Se caracterizan por almacenar la información, que inicialmente estaba en XML, a una BD, de forma que puede ya ser consultada con un lenguaje de base de datos como SQL.

Dentro de este grupo estarían, Monet [Schimidt 2000] y el sistema de Shanmugasundaram [Shanmugasundaram et al. 1999].

- b) *Sistemas que permiten convertir información en XML a Base de Datos y viceversa.* Permiten la traducción en ambos sentidos mediante algún componente especial.

Así por ejemplo, en X-Ray [Kappel et al. 2000] se incluye un metaesquema especial denominado *XMLDBSchemaMapping*.; en los trabajos de Peter McBrien y Alexandra Poullovassilis [McBrien y Poullovassilis (1) 1999] [McBrien y Poullovassilis (2) 1999] [McBrien y Poullovassilis 2001], el componente que ayuda a hacer la conversión entre diferentes modelos, es un modelo de datos gráfico especial denominado *Hypergraph Data Model (HDM)*.

### 2. SISTEMAS QUE LLEVAN INTEGRADO UN SISTEMA DE CONVERSIÓN Y UN LENGUAJE DE CONSULTA PARA LOS DATOS YA CONVERTIDOS A XML.

Este tipo de sistemas van a permitir por un lado, la conversión entre datos de distintos formatos, que no tiene porque ser de base de datos relacional a XML, y por otro lado, proponen un lenguaje de consulta para el tratamiento de la información ya traducida a XML. Dentro de este grupo están:

- El modelo de conversión Yat [Christophides y Cluet 2000] que sólo trata datos semiestructurados, normalmente ya en XML y un lenguaje de consulta propio para los datos ya convertidos, denominado YatL.
- El modelo de integración SINGAPORE [Dittrich y Domenig 1999][Dittrich y Jonscher 2000] que además de tratar datos semiestructurados, como los de formato XML y BDR, también permite integrar datos de bases de datos orientadas a objetos. Incluye un lenguaje propio denominado SOQL, que además de usarse para consultas, también permite realizar actualizaciones sobre la información ya transformada.
- SilkRoute [Fernández y Tan 2000] permite consultar información que inicialmente procede de una base de datos, en un lenguaje propio denominado RXL. Para poder realizar la consulta la información de la base de datos tiene que ser traducida previamente a XML, aunque sólo la parte afectada por la consulta (*XMLView*). Para realizar la conversión se permite utilizar cualquiera de los métodos anteriores, con tal que al final la información esté en XML para su consulta.
- XPERANTO[Carry et al. 2000] permite consultar datos de bases de datos Objeto-Relacionales en un lenguaje de consulta XML, y los resultados los devuelve también en XML. El proyecto se basa en aplicar un proceso de traducción de la consulta inicial en lenguaje XML-QL a SQL, a través de un gráfico intermedio denominado XQGM. Antes de realizar la consulta hay que obtener el XMLSchema en XML, asociado a la Base de Datos.

### 3. LENGUAJES DE CONSULTA PARA INFORMACIÓN EN XML

Si la información original procede de BD, estos lenguajes suelen tener una sintaxis más parecida al lenguaje SQL, y sino usan una sintaxis más semejante a XML con patrones y caminos. Estos lenguajes se pueden clasificar por tanto, según los dos entornos fundamentales de los que puede derivar, el entorno de bases de datos o del entorno XML.

#### a) *Lenguajes de Consulta procedentes del entorno de Base de Datos.*

- YatL del sistema Yat [Christophides y Cluet 2000]
- SOQL, del sistema SINGAPORE [Dittrich y Domenig 1999][Dittrich y Jonscher 2000]
- RXL, del sistema SilkRoute [Fernández y Tan 2000]
- XML-QL [Deutsch 1999]

- Lorel [Abiteboul et al. 2000] [McHugh 1997] [Abiteboul et al. 1997] que tiene influencias de OQL, un lenguaje de consulta del entorno de bases de datos orientadas a objetos.

*b) Lenguajes de Consulta procedentes del entorno de XML.*

- El único lenguaje gráfico, XML-GL [Ceri et al. 1999].
- Xpath, lenguaje de patrones y caminos.
- XSL que permite crear hojas de estilo e internamente también utiliza Xpath.
- XQL [Robie 1999] que es la combinación de XSL y XPath y esta claramente orientado a la consulta.
- XQuery con influencias tanto de lenguajes de consulta basados en XML, como de BD. Es quizás el más completo de todos, aunque aún esta en proceso de definición.

#### 4. HERRAMIENTAS DE CONSULTA DE INFORMACIÓN DE BD, QUE DEVUELVEN LOS RESULTADOS DE LA CONSULTA EN XML.

Este tipo de herramientas permiten consultar una base de datos en SQL, pero obtienen los resultados de la consulta en XML. Dentro de este grupo hay dos tipos, aquellas que incorporan los propios gestores de bases de datos y las que están más orientadas a obtener la información en un formato Web.

*a) Obtención de la información de consulta en XML a través de paquetes especiales de los motores de búsqueda o sistemas de gestión de base de datos.*

Los motores de búsquedas o Sistemas de Gestión de Base de Datos potentes como Oracle o SQL Server, incorporan en sus aplicaciones, herramientas especiales para el tratamiento de información en XML. Oracle9i tiene por ejemplo un paquete especial (XML Development Kit o XDK9i) que permite, entre otras cosas, hacer una consulta a una base de datos en SQL y obtener el resultado en XML, a través de lo que denominan páginas XSQL.

*b) Obtención de la información de consulta en XML a través de páginas web.*

Se trata de un tipo especial de páginas Web, que permiten consultar la base de datos en SQL y obtener el resultado en XML en una página Web especial, que se denomina página “Activa”. Dentro de este tipo están las páginas ASP (de Microsoft), las páginas JASP (implementación en Java) y páginas PHP (implementación en cualquier lenguaje).

### **3. Limitaciones de los sistemas planteados**

En la introducción anterior del estado del arte, se hace un recorrido por diversas técnicas o modelos que directa o indirectamente relacionan las bases de datos en general, con XML, y por diferentes lenguajes de consulta de datos en XML.

El tema se pretende centrar al final en tres puntos fundamentales, un tipo de bases de datos, las relacionales, por ser uno de los métodos de almacenamiento y tratamiento de datos tradicionalmente más extendido. Su relación con XML, por ser éste otro método de almacenamiento de datos, pero más actual y que cada día cobra más fuerza. Finalmente, si se tiene información de los dos tipos, cómo poder consultarla, que este acceso resulte eficiente e incluso transparente para el usuario final, es decir, que éste simplemente consulte, sin saber en realidad cuál es el sistema real de almacenamiento de estos datos.

Con las consideraciones planteadas, se trata de hacer un análisis de los sistemas mencionados y estudiar cuales son sus limitaciones en relación al tipo de sistema que se desea, cuyas características se acaban de exponer. Al mismo tiempo sin embargo, también pueden resultar útiles en otros aspectos, las ideas en las que estos métodos se apoyan.

Por ejemplo, SINGAPORE o Yat, son más modelos de integración de datos de diferentes tipos, estructurados, como las BDR, o no estructurados, que modelos de consulta, por tanto van más allá del simple acceso a las BDR e información en XML, que es lo que se pretende.

El gráfico HDM se utiliza sobre todo para hacer conversiones de información de distintos formatos pero perteneciente al mismo universo de datos, es decir, información referente más o menos al mismo tema, pero perteneciente a distintas fuentes que al mismo tiempo tienen que colaborar entre sí. En el modelo planteado no importa que los temas de la información a consultar sean diferentes.

En la tecnología ASP, JSP o PHP y en las nuevas herramientas de SGBDR como SQL-Server o Oracle9i, las consultas se hacen directamente en SQL, no en XML, y sólo sirven para bases de datos relacionales, no también para consultar al mismo tiempo documentos XML.

X-Ray permite convertir los datos de la base de datos a XML, aunque con la limitación de que hay que establecer una determinada relación de la BD como eje para convertirla a XML, y una vez que los datos estén en este formato, se puede consultar con cualquier lenguaje de consulta XML. Su limitación esta en esta conversión, pues se trata de que los datos reales de la base de datos relacional no tengan que traducirse en ningún momento ni a XML, ni a ningún otro entorno, que permanezcan en su estado original.

SilkRoute se acerca a uno de los objetivos principales, consultar una BDR con un lenguaje de consulta XML, pero para conseguirlo, una parte de la base de datos (*XML View*), la que interese según la consulta, es traducida previamente a XML para permitir su acceso. Lo ideal sería, como se ha dicho, evitar cualquier tipo de conversión de la información almacenada.

XPERANTO plantea una idea bastante similar a la expuesta, pero las bases de datos con las que trabajan no son relacionales puras, sino objeto-relacionales, con lo que aparecen nuevos conceptos a tener en cuenta en el modelo de datos, como los de herencia y tipos especiales de datos. Este proyecto puede aportar alguna idea, aunque por ejemplo, falta aclarar como se hace exactamente la conversión de la base de datos al XMLSchema de XML. Por otro lado, podría resultar más práctico no tener que usar ningún gráfico intermedio para hacer las conversiones, sino que se puedan hacer directamente desde la información que se deriva del esquema de base de datos que se obtiene del propio sistema gestor de base de datos.

El objetivo final sería permitir la consulta de información almacenada en una Base de Datos Relacional o en un documento XML, por medio de un lenguaje de consulta basado en XML y retornando los resultados de la consulta también en XML. De esta forma todo el entorno tendría un aspecto XML para el usuario, aunque internamente este no fuera el formato real de los datos. Además se trata de lograr todo esto evitando los problemas que hemos comentado, por ejemplo que los datos no tengan que ser convertidos a ningún formato intermedio, que permanezcan siempre en su formato original.

#### **4. Conclusiones y Líneas Futuras**

El estudio de las limitaciones de los modelos descritos justifica aún más la necesidad de crear un sistema que permita consultar datos que pueden pertenecer a Bases de Datos Relacionales o a documentos XML, pero además con el mismo sistema de consulta, y basado en XML, para que todo el entorno tenga la misma apariencia.

Lo que se plantea como trabajo futuro es establecer por un lado, el modelo que nos permita migrar la Base de Datos a un formato que la haga accesible desde XML, pero sin tener que hacer ningún tipo de conversión con los datos o registros de información de la base de datos, que era uno de los problemas de algunos de los sistemas anteriores. La idea es migrar el modelo de datos que define a la base de datos, su modelo Entidad-Relación, no sus datos, a un modelo semiestructurado, como son los DTD o XMLSchema de XML.

La segunda tarea fundamental es establecer el lenguaje de consulta, basado en XML, que se va a usar este entorno y sobre todo, plantear como se pasaría de este lenguaje a SQL, que es el lenguaje con el que al final se va a acceder a la base de datos. No se trata de inventar un nuevo

lenguaje, se puede reutilizar alguno de los lenguajes expuestos procedentes del entorno de XML, para que todo tenga el mismo aspecto, y adaptarlo, si se requiere, a nuestras necesidades específicas.

## 5. Bibliografía

S. Abiteboul, P. Buneman, D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan-Kaufmann 2000

S. Abiteboul, D. Quass, J. McHugh, J. Widom, J. Wiener. *The Lorel Query Language for Semistructured Data*. International Journal on Digital Libraries, vol. 1, no. 1, pg (68-88). Abril 1997.

T. Bray et al. *eXtensible Markup Language (XML) 1.0*. World Wide Web Consortium. Disponible en. <http://www.w3c.org/TR/REC-xml> 1998

M. Carry, D. Florescu, Zachary Ives, Ying Lu et al. *XPERANTO: Publishing Object-Relational Data as XML*. WebDB (Informal Proceedings), pg. 105-110. Disponible en <http://citeseer.nj.nec.com/christophides00wrapping.html>. 2000

Vassilis Christophides, Sophie Cluet. *On wrapping query languages and efficient XML integration*. Disponible en <http://citeseer.nj.nec.com/christophides00wrapping.html>. 2000

S. Ceri, s. Comai, E. Damiani, P. Fraternali, S. Paraboshi, L. Tanca. *XML-GL: A Graphical Language for Querying and Restructuring XML Documents*. Proceedings of the International WWW Conference. 1999

S.J. DeRose. *XQuery: A Unified Syntax for Linking and Querying General XML*. Proc. of the Query Languages Workshops (QL98), Cambridge, <http://www.w3.org/TandS/ql/QL98/xquery.html>, Diciembre 1998.

Klaus R. Dittrich,, Ruxandra Domenig. *Towards Exploitation of the Data Universe*. 3<sup>rd</sup> International Conference on Business Information Systems, Poznan, Springer, Abril 1999

A. Deutsch, M. Fernández, D. Florescu, A. Levy, D. Suciu. *A query language for XML*. International WWW Conference. 1999

Klaus R. Dittrich,, Dirk Jonscher. *All Together Now: Towards Integrating the World's Information Systems*. V Jornadas de Ingeniería del Software y Bases de Datos., Valladolid (España). Noviembre 2000.

Mary Fernandez, Jerome Simeón, PhilipWadler. *XML Query Languages: Experiences and Exemplars*. Disponible en <http://www.db.research.bellabs.com/user/simeon/xquery.ps> , 1999.

Getti Kappel, Elisabeth Kapsammer, Werner Retschitzegger. *X-Ray: Towards Integrating XML and Relational Database Systems*. International Conference on Conceptual Modeling, the Entity Relation Ship Approach (pag. 339- 353). 2. Julio 2000

D. Maier. *Database Desiderata for an XML Query Language*. W3C Workshop on Query Languages for XML. 1998

Michael Morrison, et al. *XML Al descubierto*. Prentice Hall 2000

J. McHugh, S. Abiteboul, R. Goldman, D. Quass, J. Widom. *Lore: A Database management system for semistructured data*. ACM SIGMOD Record, vol. 26, no. 3, 1997

Ioana Manolescu, Daniela Florescu, Donald Kossmann *Pushing XML Queries inside Relational Database*. Tech. Report no. 4112, INRIA. Disponible en [www.caravel.inria.fr/Epublications.html](http://www.caravel.inria.fr/Epublications.html), 2001

P. McBrien, A. Poulovassilis. *Automatic Migration on Wrapping of Database Applications. A schema Transformation Approach*. Proceedings of ER99, vol. 1728 of LNCS, PG. (96-113). Springer-Verlag 1999

P. McBrien, A. Poulovassilis. *A uniform Approach to Inter-model Transformations*. 11<sup>th</sup> International Conference on Advanced Information Systems Engineering (CAiSE'99), vol. 1626 of LNCS pg. (333-348). Springer Verlag 1999

P. McBrien, A. Poulovassilis. *A Semantic Approach to integrating XML and Structured Data Sources*. 13<sup>th</sup> International Conference on Advanced Information Systems Engineering (CAiSE'01). 2001

Mary Fernandez, Wang-Chiew Tan. *SilkRoute. Tradding between Relations and XML*. WW9/Computer Networks journal, vol. 33, no. 1-6, pg (723-745). 2000

J. Robie. *The design of XQL*. Disponible en <http://www.textel.no/whitepapers/xql-design.html>. 1999

Albretch Schimidt, Martin Kersten, Menzo Windower, Florian Lucas. *Efficient Relational Storage and Retrieval of XML documents*. WebDB 2000, pg.( 47-52). 2000



Jayavel Shanmugasundaram, Kristing Tufte, H. Gang, Chung Zhang. *Relational Databases for Querying XML documents: Limitations and Opportunities*. The VLDB journal on the Proceedings of the 25<sup>th</sup> VLDB Conference, pg. (302-314). 1999

. Direcciones de Internet:

- <http://www.w3.org/>: World Wide Web Consortium
- <http://www.w3.org/xml>: W3C Extensible Markup Language (XML) 1.0
- <http://www.w3.org/TR/xpath>: XML Path Language (XPath)
- <http://www.w3.org/TandS/QL/QL98/pp/xql.html>: XML Query Language. (XQL).
- <http://www.w3.org/TR/xquery>: W3C XQuery 1.0: An XML Query Language
- <http://www.w3.org/Style/XSL>: W3C XSL and XSLT.