

XML and information on demand

Recently, we often hear the concept of 'information on demand.' Information on demand is understood in the sense that the organization's information system must meet all user requirements, from leadership to professional staff; from statistical information, forecast to detailed information for daily activities ...

Recently, we often hear the concept of 'information on demand.' Information on demand is understood in the sense that the organization's information system must meet all user requirements, from leadership to professional staff; from statistical information, forecasting to detailed information for daily activities . This information must be quickly responded to any changes coming from outside the organization, such as changing rules, changing items or business methods, actions of competing firms .

To fulfill the information required, the world IT (IT) industry has moved application software from project-based architecture to service-oriented architecture (SOA).

Project-oriented applications are the implementation of relatively independent applications. For example, own asset management application with financial management application, own personnel management application with business application . Applications are built with own project management, do all the work From design, construction to procurement of deployment equipment . After a while, separate applications have taken effect. However, there is a requirement that arises and is the need of every organization, which is to integrate those applications into a unified whole. The organization itself is a unified body, thus doing so can meet the general requirements of business, respond quickly to fluctuations of the market, of competitors .

So, to build a system of 'information on demand', the most important job is to integrate individual applications. These applications can run on different platforms, written in different programming languages, put into use at different times. If the number of these applications is small, people will edit them to 'talk' to each other. But in fact this job is not simple, even impossible to do. So what is the exit for the need to integrate applications? That is service-oriented architecture.

Service-oriented architecture is an architecture that takes services as a kernel. Services are applications that have been 'disintegrated' to the point that the smallest unit provides information (how small is it depending on the designer). To perform a certain task, people will assemble a series of services in a certain order. When the workflow changes, just rearrange the order in which the services are executed (even the user can decide for themselves how to perform those services for his work, without relying on IT professionals, thanks to available tools.

Features of service-oriented architecture are:

- It is liquid contact between systems, independent of each other.
- Independent platform and language.
- Reuse existing technology infrastructure, so it's simple, cheap.

- Supported by many industrial companies (IBM, Microsoft, SAP, Oracle .).
- Everything is based on Web services / XML and other industry standards.

There is a concept of 'services', so there will be a place to provide services (a source of information) and a place to use those services (business functions). So how to use those services? Just as the main program calls functions or subroutines to complete a procedure, there is also the concept of calling the service. Here comes the Web service (Web Service) and related standards. In general, Web services are a standardized way to call functions on the network without knowing where the function is, running on any operating system platform, writing in what language.

Service invocation is carried out by sending a message, where the service is requested to send an information request message to the service provider, which provides the result package into a message and returns it to the requested location. . Applications integrate with each other by sending questions and answers, without caring about how to get that answer. Therefore, applications on different platforms, different languages ?? want to integrate must understand the message language together. The language for writing these messages is the 'eXtensible Markup Language' (XML).

The simplest way to understand XML is to compare it with HTML. HTML is a hypertext markup language, used for the purpose of 'presenting' data. This means how the data will be seen in the user's browser. Meanwhile, XML aims to say what it is 'Content'. Here are some key concepts of XML language.

- Is the standard for describing the meaning of data, what is this data. For example, also a piece of text but using XML, we know where they are, what is the name; With a sequence of numbers, you know whether it's the price or the amount .
- Tags in XML are not pre-defined, we can define tags themselves.
- Technology independence, regardless of the hardware and operating system and the different software.
- XML ??can connect to applications written in other languages.

At this point, many documents can be formatted in XML language, from configuration files, documents, spreadsheets . The purpose is for different systems to exploit the documents. This is easy. In the case of service-oriented architecture, it is the language for applications to 'talk' to each other.

Because this article relates to the database, which XML documents are also data, we should mention a little more carefully about it. XML documents are semi-structured data (hierarchical models or tree models) and sequential (compared to relational data are structured sets), the type of XML data is an option while type in Relational model is required, its standard is regulated by W3C, and the relational model is specified by ANSI / ISO.



Figure 1. Representing the XML and the relationship of the two records

On Figure 1, we see that the rows on the unordered table are pre-ordered (based on set theory), while on XML there is an order. Structured tables, rows with the same number of columns, and XML are not necessary (the first record has a phone number, the second record is not available) (Figure 1).

Thus, the importance of the 'information-on-demand' system is seen. To build this system, integration of existing applications and new applications is required. But for application integration, it must be based on 'service-oriented architecture'. And to build a service-oriented architecture, Web services (and standards) and XML are fundamental technologies. Because XML is so important, in fact, having a database for XML is an issue that many businesses care about (with the high features and requirements of a database management system). material). There are three ways that are used to store and search XML records, here are three ways.

XML-enabled database (XML-enable)

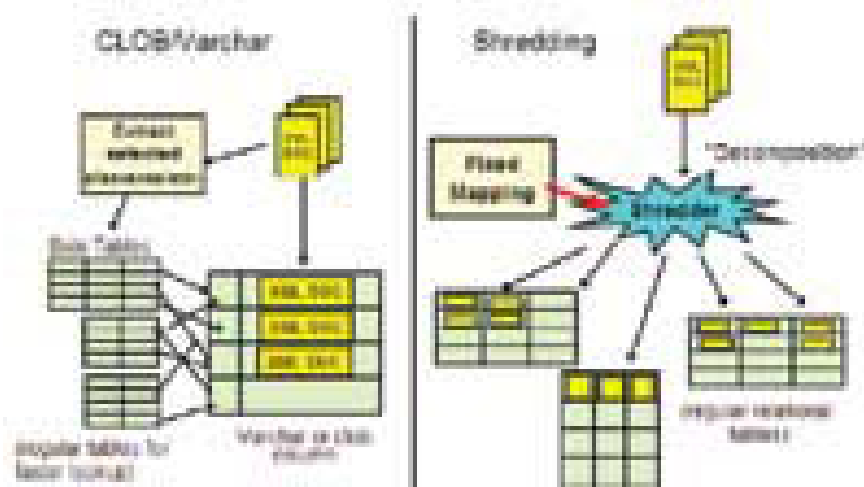


Figure 2. Two options for storing XML data in XML-enabled databases

The first two ways belong to this group, an XML-enabled database will use the relational model as the main model for storing data. This requires mapping between the XML data model and the relational data model, or storing XML data as a Varchar / CLOB-type large object. Figure 2 explains in more detail the two options for XML-enabled databases. (Figure 2)

The figure at the left shows the method of storing XML documents in the database 'CLOB and Varchar'. The XML document using this method is saved by both the CLOB image and the Varchar type column in the database. This method is not flexible because it is difficult to find the inner elements of the image. Its performance is not high because XML images are often very large, thus consuming a lot of memory (it's convenient to save and retrieve the full XML document).

The second option for XML-enabled databases is the decomposition method, which is illustrated in the figure to the right of Figure 2. The entire XML document using this method is broken down into parts (elements) and then save in the tables. Using this method, the hierarchical model of the XML document is transformed into a relational model. This method is also not flexible, a change in the XML document is not easy to spread to the corresponding tables and may have to create many other tables as needed. This method will not work if you need to retrieve the original XML document.

Pure XML database (pureXML)

This is the third way, the pure XML database uses the hierarchical model of XML data to store and process XML inside. The storage format is the same as the processing format, with no mapping to the relational model, and does not save XML documents as images. When using XPath or XQuery statements, a separate mechanism will handle those statements and not switch to SQL. This is why these databases are referred to as pure XML databases. Currently, DB2 9 is the only commercial database that provides this feature.

DB2 9 database

DB2 Version 9 Database Management System is a database that both supports a relational model and supports a hierarchical model. It was published in 2006, and is the only commercial database (up to now) that supports all of the three XML storage methods mentioned above.

DB2 9 stores XML data into a column of type XML, it has a translator that can handle XQuery statements (the standard language for querying XML documents), works with different Cobol languages, C, Java, . can index with XML data . The following examples demonstrate the power of data access language

DB2 Express-C is a free, unlimited and easy-to-use DB2 edition. The C in DB2 Express-C stands for Community. The DB2 Express-C community includes individuals and companies that design, develop, deploy, or use database solutions.

DB2 9 alters the way in which XML information is managed (storage, search, and processing) that is highly effective in integrating XML with relational databases. It allows bringing data services to a new level at a cheaper cost, making it lighter, and improving business profoundly, making DB2 9 an important component of consider information to be an information technology infrastructure service.

You finished reading the article "**XML and information on demand**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

