**Proceedings of the IASTED International Conference**
**APPLIED INFORMATICS**
**February 18-21, 2002, Innsbruck, Austria**

# GETTING INFORMATION FROM DISPERSED DATABASES THROUGH HYPERQUERIES

J.J. BOUMA[1] and J.H. TER BEKKE[2]
[1] Partheon.com BV, Van der Feltzpark 3, 9401 HM Assen, The Netherlands, admire@analoog.com
[2] Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands, j.h.terbekke@its.tudelft.nl

## ABSTRACT

The concept of hyperlink determines the way the world wide web is used today. Hyperlinks let users navigate from web page to web page, from document to document. In its present form the web is predominantly a hypertext web. In addition, navigating seems to imply that users follow static links predefined by others, not dynamic paths created on the fly by users while they surf on the web. We propose to use hyperlinks as queries to dispersed databases on the Internet, called hyperqueries. In this way databases can collaborate and users can define their own applications by setting chains of hyperqueries.

**KEYWORDS:** collaborative databases, hyperquery, Internet application, semantic modeling.

## 1. INTRODUCTION

The World Wide Web (now popularly known as “the web”) was originally developed by a research group headed by Tim Berners-Lee at CERN in Geneve in early 1990 as a large-scale hypermedia information system for all kinds of documents, telephone lists, conference announcements, research papers and library books. He noticed that the information was scattered throughout the organization without a consistent unifying system to access all dispersed data. And such a system was exactly what was needed. The idea to connect the data stored on all computers around the world resulted in the exponential developments of the Internet in recent years. The Internet enabled us to access all pieces of information on any computer and anywhere in the world.

Through this vision, the Internet has developed into the web as we know it today. Internet is a universal communication structure for computers to be interconnected. Computers can communicate with each other on the basis of standardized set of protocols, i.e. common rules by which computers can send data to each other. The data can be distributed via a diversity of media, like telephone lines, television cables or satellite channels. It does not matter what kind of data, text, email messages, sounds, images, software, etc. is transmitted.

The basic principle behind the web is that someone at a certain moment publishes something and that this document, database, image, sound, or video is accessible (of course depending on proper authorization) for anyone regardless of the type of computer that is used. It is also possible for everyone to make a reference to something so that others will be able to find it. This is called a hyperlink. Web users can make bookmarks for every location, to be able to remember them and retrieve them. In any document they can create links to other documents [1].

The idea of navigating through a series of virtual pages dispersed on various unknown computers was a completely new approach. People were used to finding information, but without the web, they would never create links to other computers. If they wanted to do so, they had to program a long and complex list of instructions to access the other computer. People have started to use a computer with a completely new perspective. Through global hypertext we have started to think in hyperlinks instead of programming instructions.

A hyperlink is a simple identification, an URL, which contains all essential details of an information address. The URL is the most fundamental innovation of the web. It is the only interface that is used by any web application, client or server when someone follows a hyperlink. As long as a document has an URL, it can be stored on a web server and found by any web browser. On the computer screen a hyperlink is only a colored word. But hidden behind it is an URL, which tells the browser where to find the required document [1].

Hyperlinks make a web expandable and decentralized. By using a browser that can follow external hyperlinks, one can create new webs. Existing computers can be connected with each other. All new computer systems can cross their own borders by referring to other computers. In addition any user can add a new hyperlink when surfing, thus creating a new node in the information network. The Internet also has another basic property: it is fundamentally decentralized. It is the only way a user can access the web anywhere without having to ask someone permission first. And it is the only way the web can expand without involving the risk of congestion as more and more people make use of the web.

These computers contrasts strikingly with many existing computer systems, which depend on a central node with which they are connected. The main disadvantage of this centralized concept is that the capacity will eventually limit the expansion of the whole system. The web is completely different. It possesses unlimited flexibility. New links are added every day and a web of hyperlinks can be evenly dispersed over the whole world. Any node can be connected with any other node.

## 2. DATABASES AND THE INTERNET TODAY

The most striking aspect of the way the web is used today is that it appears to be a network that principally connects documents with each other. By way of hyperlinks users navigate from web page to web page, from brochure to brochure, from document to document. In its present form the web is predominantly a hypertext web. Secondly, web surfing implies following existing links set by others. One can follow a chain of links randomly, because others have set the nodes. Users have to collect the fragments of information and combine them into a useful result.

Web pages fit smoothly in the communication with existing back office databases. The web is mainly a hypertext web. Yet most information we want to access is stored in structured back office databases. Normally the interaction with these existing databases proceeds through window clients, by means of screen forms for data entry and reports for retrieving data. Web pages can easily imitate this way of communication and replace the traditional window-client interface. For that purpose several techniques have been developed to present information from a database on a web page. What they all have in common is that a connection with the database is established by means of a script in the web page that is requested from the web server. This is the case with active server pages (by Microsoft), where Visual Basic scripts handle the connection with the database. Java Server Pages (by IBM) contain Java code to collect information from a database. The result from a database, mostly a table, is usually transformed to HTML by a print module on the web server and is inserted in the web page. Another way to achieve this is to embed the script in an XML page. This is the method used for XSQL-pages (by Oracle). The SQL-queries, which are passed on to the database, are incorporated in XML pages. The query in the XML page is replaced by the query result, and the XML is subsequently transformed to HTML by a XSL style sheet.

A practical and feasible way to publish reports produced by back office databases on the web is to introduce a central node. This is possible because web pages fit into the traditional way of communication with databases. This way the advantages of databases are brought to the scale of the Internet, i.e. a well-accepted form of computer-interaction and formulating queries on intelligent connections between database data. Characteristic of this way of communicating with a database is that a web page is considered as a form for data-entry of a report for presenting database data. In this concept forms and reports are connected with one database. Navigating with hyperlinks means moving from form to form, from report to report.

## 3. COLLABORATIVE DATABASES

Until the advent of ERP-systems, databases formed islands within an organization. The only possible way of communicating between them constituted of mailing transactions or copying data. Direct links between data in different databases were not possible. At that time organizations searched for a solution to concentrate all relevant company wide data into one single massive database. In principle this can be realized within one company, but it is not feasible to connect databases of different companies. Fortunately communication between organizations mainly consist of transactions. Transactions can be seen as notifications, like orders and bills, that used to be sent by letter, but can nowadays be delivered by means of email. Evidently the most common situation is that databases operate independently and are dispersed. If any communication exists, it is achieved by sending transactions.

Yet there is a need to extract connections between data of different organizations. Think for instance of assembling economic statistics. Or consider product catalogues. It is impossible to concentrate all product information from manufacturers and suppliers in one nation-wide database. This is also feasible for: population registers, public regulations, telephone books, travel brochures, public transport time tables, patient data or information on public authorities, etc.

A typical example of databases that operate independently yet cooperate by nature is a collection of product catalogues. In a distribution chain all manufacturers, distributors and suppliers usually create their own product catalogue. In addition customers usually compile their specific catalogue for the purpose of purchasing and stock management. Until now every party of the supply chain used to fill its own catalogue with the data of the manufacturer or supplier, i.e. one is condemned to copying product data.

Product catalogues form a network. This can be seen by the way product information is used when it is exchanged by telephone. Customers ask questions about specifications, stocks and delivery periods. They ask questions like: “What white television sets do you sell?” or “Do you sell Electrolux refrigerators?” etc. When a supplier does not exactly know the answer, he will contact his distributor, who in turn will contact the manufacturer. The answers are collected, the customer is called back and given the answer.

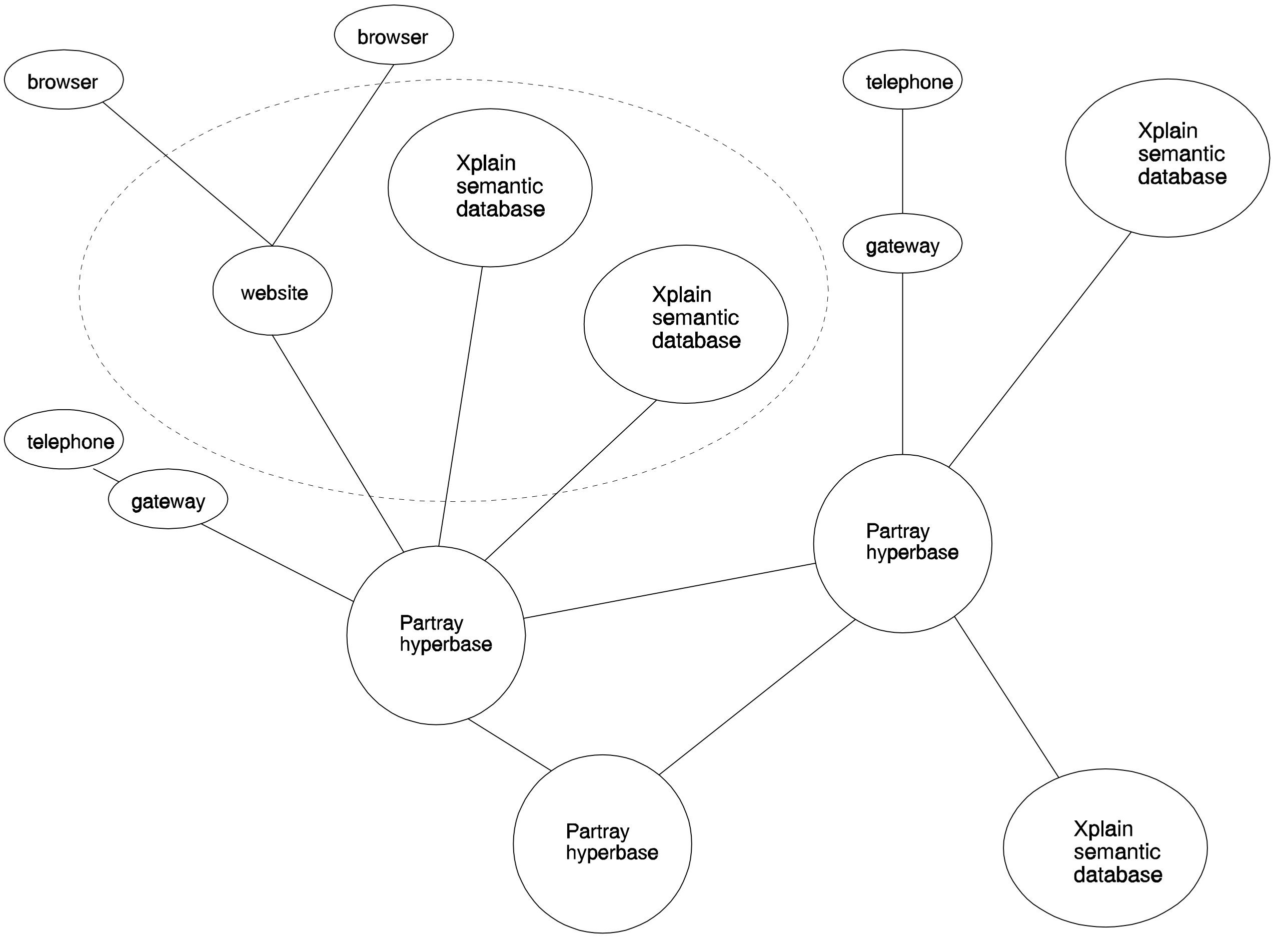


*Figure 1: Dispersed databases*

Similarly, a catalogue network would be able to pass on questions from customers to manufacturers. It is even possible for a supplier to build his catalogue merely from questions to his distributor, like trade agents who represent certain manufacturers, or stock catalogues for customers that refer to supplier catalogues for product specifications.

Suppose we would create a hyperlink that does not navigate from web page to web page (using hypertext), but from database to database on the Internet. We would consider a hyperlink as a question that is directly submitted to a database. Or, to take it further, a query could be submitted to a web of databases, in other words, a hyperlink would be used as a hyperquery.

Imagine that we would no longer regard a web page as a static document, but as an answer to a database query. This answer could then also trigger the next question to a database, just like a clever journalist uses each answer to pose a new question. Consequently, if we would be able to let a hyperlink navigate from database to database, what opportunities would be created for web applications?

With such a hyperquery we would be able to consult all kinds of databases on the Internet or within an Intranet as if one were working with one large virtual database. Just like a web server that interprets hyperlinks, an intermediary server is needed, which understands hyperqueries. This is called a hyperbase. So the hyperbase itself is not a database; it does not store data for client applications. It only manages the communication of questions and answers. The hyperbase is able to receive a question from a browser, a website, an application or via a telephone call. The question can be forwarded to a database directly connected to the same hyperbase, or to another hyperbase. The answer will be received as HTML, WML or, if a person puts the question through a telephone, by means of a digitally composed voice [2]. Just like common web servers, all hyperbases in a web can be technically identical. Thus, a web of hyperbases can expand as naturally as web servers have proliferated to establish the current Internet.

In this framework each user would be able to compose his own applications. Each database connected to a hyperbase should be designed to tell the user which questions it can understand and reply. This enables users to submit their own personal chain of queries to the hyperbase. The user himself decides which databases or applications he wants to connect to or which users he permits access to his data. This resembles the way people behave when they decide which people to communicate with. To create his personal application the user only has to submit his chain of queries to the hyperbase. His application will operate robustly. Only the configuration of the hyperbase is altered, not the hyperbase software itself. A consequence would be that we would be able to store knowledge locally, at dispersed locations and yet the data would be as easily accessible as with massive databases as central nodes in an organization. Knowledge that the central office would need, could be asked and received from local operations. Computers would be able to ask questions to each other. A web of databases would emerge that as a whole would combine more knowledge than any independent central database could ever contain. There would be no need to copy local information to a central database node. Knowledge can be stored at the location where it is created and managed. There is no need to email answers, for answers that can be asked, do not have to be stored and remembered.

Hyperbases operate as a communication center in a web of databases. The user is allowed unrestricted authority for the location, security and management of his websites and databases. The consequence is that participating websites and databases can remain unchanged. The information and knowledge of participating organizations remain within their own security domain. Also the existing way in which databases operate, can be preserved. The only requirement is that connected databases are able to interpret hyperqueries, i.e. they must be able to interpret a hyperlink as a query.

The interaction with other databases will lead to cooperation, where organizations will adapt their language of questions and answers to each other. It is also to be expected that databases will be made more ‘intelligent', as the number of questions from various user perspectives will continue to grow.

If hyperqueries would be common, people would no longer be forced to operate a computer with mouse and keyboard, but would be able to ask questions to an organization that in turn automatically directs queries to a database. The web would spontaneously evolve towards a level of intelligence that is requested by the users. Databases that communicate will learn from each other. Users will only ask questions to computers that produce optimal answers. Just like people learn which person to address if they want to know something.

As more and more intelligent databases will connect, a knowledge web will eventually develop and expand in a natural way, just the way the telephone system has evolved. This is due to the fact that all databases are equivalent nodes in the network. There is no central node, as is the case with the current massive central databases.

Special databases will be developed that manage information that everyone makes use of, like interpreters, translators, dictionaries, indexes and classifications. These databases will operate as search engines, for example for public regulations, music, museums, product catalogues, and estate agents etc.

Finally, if we would create a hyperquery that would navigate from database to database, then we would be able to use the answer of one database for a question to another database. We would be able to use the answer of one organization for a question to another organization. Various databases that up till now operate independent of each other could then be treated as one virtual database.

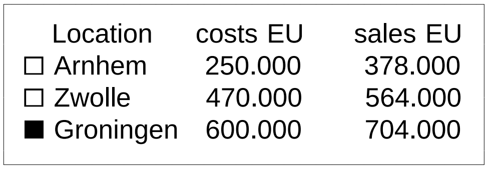
## 4. SOME APPLICATIONS

Using the hyperquery principle we have actually built an application that collects management information from collaborative databases. We used a fictitious chain of supermarkets as an example. We restricted ourselves to a company that operates in one country, where it has outlets in all major towns. The case is restricted to the management questions about sales figures of the various supermarkets. For simplicity, operations like customer relations, accounting, purchasing, logistics and personnel management are ignored.

What information is interesting to the management of a supermarket? For example, a shop manager would be interested in turnover of products related to season, marketing campaigns or economic prospects. For financial managers changes in turnover, market share and profit margins of product groups would be of interest.

In our example a browser sends a question to a hyperbase which forwards the question to all local offices [4]. The databases at the local offices receive questions in the form of an URL and send the answers wrapped in XML to the hyperbase at the central office [3].

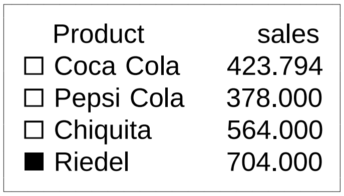
The question of the shop manager is for instance: What is the annual turnover of soft drinks of every local shop? Any local database can derive the answer from its business model. The hyperbase collects the answers.



| Location | costs EU | sales EU |
|---|---|---|
| □ Arnhem | 250.000 | 378.000 |
| □ Zwolle | 470.000 | 564.000 |
| ■ Groningen | 600.000 | 704.000 |

*Figure 2: Diagram of sales of soft drinks per location*

The next hyperquery shows the annual turnover per brand of the chosen local supermarket.



| Product | sales |
|---|---|
| □ Coca Cola | 423.794 |
| □ Pepsi Cola | 378.000 |
| □ Chiquita | 564.000 |
| ■ Riedel | 704.000 |

*Figure 3: Diagram of sales of soft drinks per brand*

You may have noticed that no codes appear in the answers to the managers. Nor a code of a product group, or a product code. The questions are formulated in normal language, using names of product groups of brands. Next one could zoom in on the annual sales of soft drinks of a particular brand per local shop. Similarly we could ask a figure for the turnover of all shops together. In that case the hyperbase collects all local answers and presents the total.

## 5. IMPLEMENTATION

The example above was implemented with the Partray Hyperbase. Any database that can interpret a hyperlink as an hyperquery and is able to wrap the reply in XML can be connected. In this case we used the Xplain DBMS, a semantic DBMS (see [6]), which is particularly suitable to produce answers to any management question. In particular questions like:

- How much coffee has been sold in the month after the marketing campaign?
- Have the sales of frozen foods increased this year, or decreased?
- What were the profits of the sales of meat?
- Which shop sells most cosmetics?
- Which shop has the lowest distribution costs?

lead to reliable results because the semantic language will avoid query pitfalls [5]. In addition Xplain DBMS is able to inform the Partray Hyperbase about:

- Which questions can be posed,
- Which terms the database understands,
- The meaning of the XML schema used to wrap the answer.

Our research focuses on implementing this kind of queries on the web with the Partray Hyperbase connected to collaborative Xplain databases.

## CONCLUSION

A hyperbase is a web server that submits hyperqueries as questions to databases. Answers are sent to the user in HTML, WML or voice. This way dispersed databases can collaborate as if they form one virtual database. Users can instruct the hyperbase which chains of hyperqueries to follow, thus creating their own personal application. A wide range of database applications on the web can be realized with Xplain semantic databases and Partray hyperbases.

## REFERENCES

[1] Tim Berners-Lee, *Weaving the Web* (London, Orion Business Press, 1999).

[2] Brett McLaughlin, *Java & XML* (Sebastopal CA, Oreilly & Associates, 2001).

[3] Rusty Harold & W. Scott Means, *XML in a nutshell* (Sebastopal CA, Oreilly & Associates, 2001).

[4] S. St. Laurent, J. Johnston, E. Dumbill, *Programming web services with XML-RPC* (Sebastopal CA, Oreilly & Associates, 2001).

[5] Bert Bakker and Johan ter Bekke, Foolproof query access to search engines, *Proc. 3rd Int. Conf. on Information Integration and Web-based Applications & Services (IIWAS 2001)*, Linz, Austria, 2001, 389-394.

[6] J.H. ter Bekke, *Semantic data modeling* (Hemel Hempstead, Prentice Hall, 1992).