IVQS: a visual client system to query remote multimedia databases through the web with an iconic interface

Marina Moscarini, Flavio Fontana, Marcello Valeri

Dept. of Computer Science, University of Rome "La Sapienza"

1.Introduction

The Web is large, dynamic, information and data intensive, and ranges from completely unstructured (HTML pages) to fully structured (DBMS). Despite that, today's global society looks at the World Wide Web for information acquisition. The ever-increasing popularity of the WWW, in terms of data available and number of users, raises critically the problem of making it easier for the end user to search information on the Net. Besides, the problem of integrating and querying data coming from multiple heterogeneous sources has been the subject of many studies in the last years [13]. In this context the data warehouse technology has been developed [2], usually intended as a database system that provides query capability to analysts engaged in extracting strategic information about a enterprise.

Therefore, studies and researches have been carried on to develop advanced visual interfaces to simplify the formulation of an information request [3,4,11,15,19].

We developed a Visual Query System (VQS), called IVQS (Iconic Visual Query System), that allows the user to query remote and different multimedia RDBMS through the Web using an iconic interface. IVQS is intended for Web end-users who can have a quick and easy way to access and query a database over the Net thanks to an advanced visual iconic interface. In addition IVQS can be seen as a tool for data warehousing since it's possible to collect data from different remote RDBMS and eventually to customize the result display for the analyst [20,21]. In fact, a special system feature is the possibility to execute a query on tables of two or more different databases (which can be data warehouses allowing a sort of distributed data warehousing). This means, for instance, that it is actually possible to execute a join between a table in a Oracle database running on a Unix Workstation and a table in a Access database running on a PC.

IVQS has been developed as an applet Java so it's completely platform independent and it doesn't need any installation on the client side.

2.System Overview

The IVQS prototype is a Visual Query System developed at ENEA's Usability Lab to access and query remote databases through the Web.

The system allows to create and execute queries to remote databases and to have a result visualization.

The system is a Java Applet, executable directly through a Java-enabled Web Browser (without needing any installation).

The system provides two kind of iconic querying (fig.1):

- *Query icon*. Each icon represents a query. More icons can be grouped inside a folder so that the icons are structured in a tree. The user can execute the query double clicking on the icon.
- *Entity icon*. Each icon represents an entity (a table in a relational DB). Between two entities a query can be associated. Dragging an icon entity and dropping it on to another one, the associated query, if exists, is executed. For example, if we have the entities (tables) DEPARTMENT and EMPLOYEE, we could associate a query that is the join on the common field DEPARTMENT_ID.

An Icon Editor is provided for the user to easily associate an icon to a query or to an entity. In

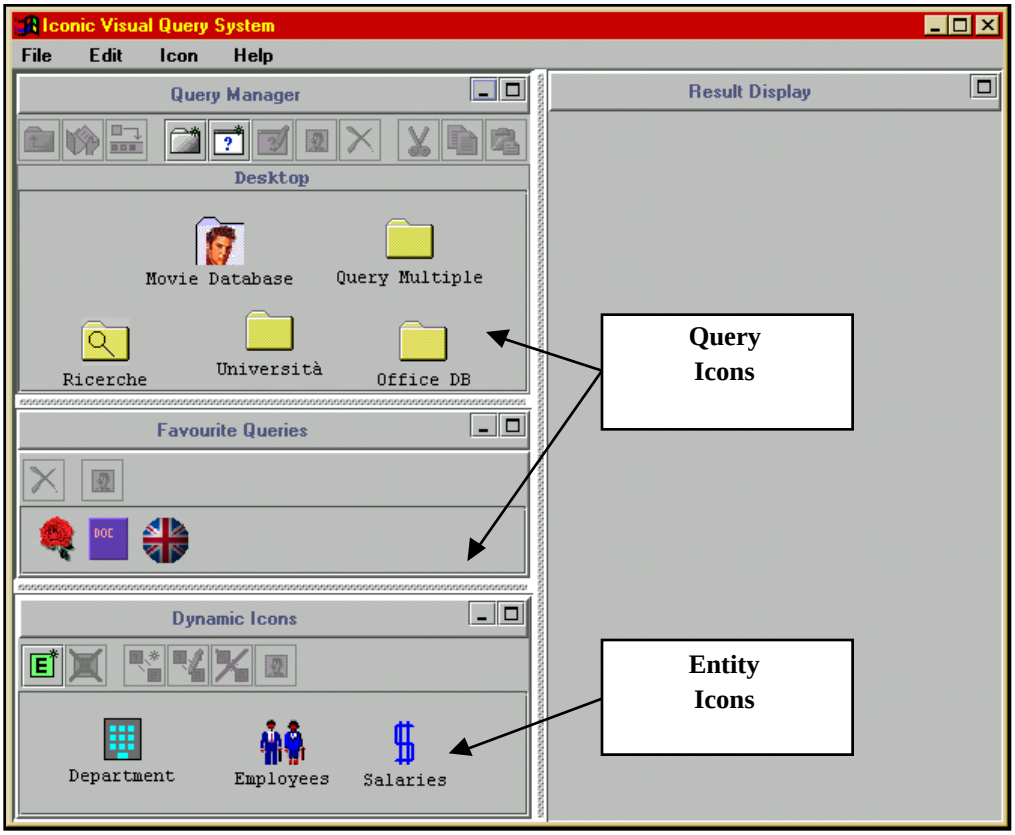


Fig.1 - Two kind of Iconic Querying in the IVQS interface

this way the user can create his own icon or import an existing image specifying an URL.

The system provides a multi-user environment with userid / password authentication. Each user can have a list of his own favorite queries.

Two kind of user are handled:

- Expert User (*Super User*). This kind of user can manage the system to create and modify queries, entities and can set up the result display layouts. He has a good knowledge of relational databases and of their structure. He can create queries visually selecting databases, tables, attributes and conditions (fig.2). The system can execute queries to the most known relational databases such as Oracle 7.x, 8.x, MS SQL Server, MS Access, Informix, Sybase and ODBC-compliant data sources). In particular, choosing two or more different database, he's able to build a query involving tables from different remote RDBMS as we suggested earlier in the introduction.

  He can use the system by himself or as an administrator for a set of end users.
- End User (*User*). This kind of user uses a structure already set up by a super user. He deals with predefined queries and entities. Typically he's neither a computer nor a database expert; he can use the system without knowing the internal structure of the databases. He can have his own set of favorite queries.

Every data related to the user is stored in the system database.

IVQS provides two kind of result display: table and form. In the form result display (fig.3) it's possible to customize the visualization layout in terms of object position, font type, size, style and color.

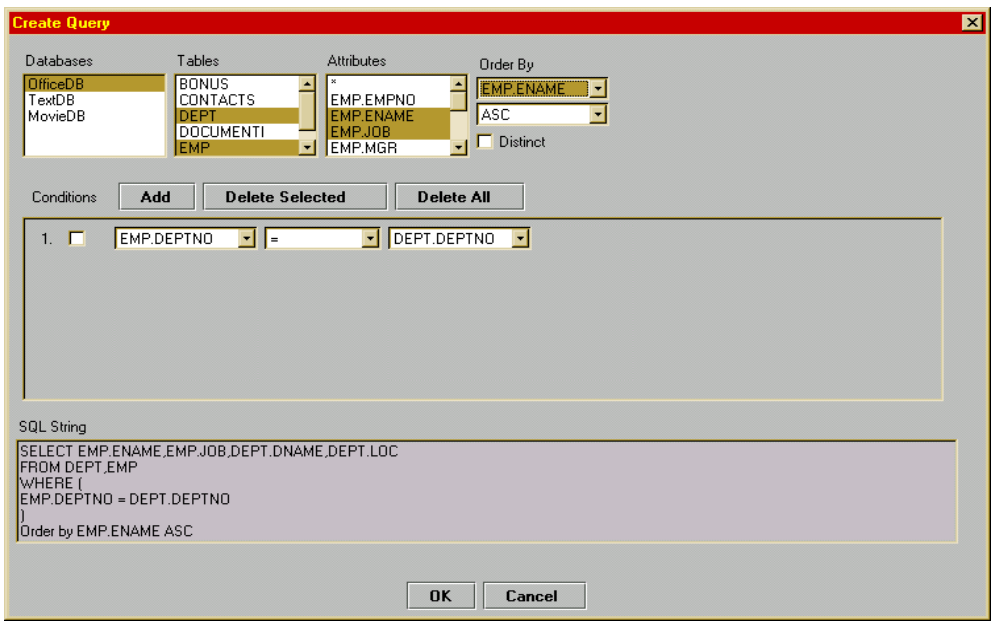


Fig.2 - The create query dialog

Objects can be text labels, images, Web resource links (web pages, e-mail address, executable files) as URLs. In the latter case the user can open the resource that will be handled by the browser.

Multimedia objects, such as movies or sounds, are placed in a specific area (Media Area, fig.4).



Fig.3 - Form Result Display

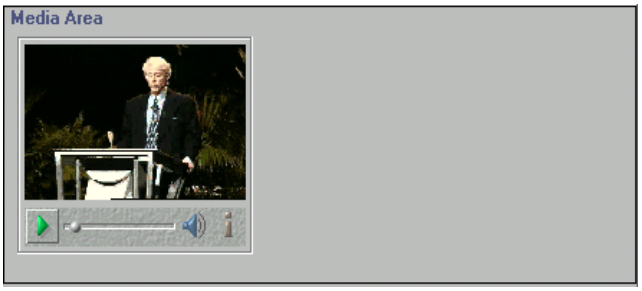


Fig.4 - Media Area

We can now look at two typical uses of IVQS:

- IVQS as a visual database interface for the end user, who can access a set of remote databases in a quick and easy way using predefined queries pre-built by the administrator.
- IVQS as a tool for distributed data warehousing (fig.5), that is when not only the source database but also the data warehouse is distributed (for example, each store of a large grocery chain should keep a local data warehouse concerning product sales of that store). In this way the analysts give specifications to the IVQS system administrator for building queries to specific and strategic enterprise data warehouses and for convenient result display of these queries.

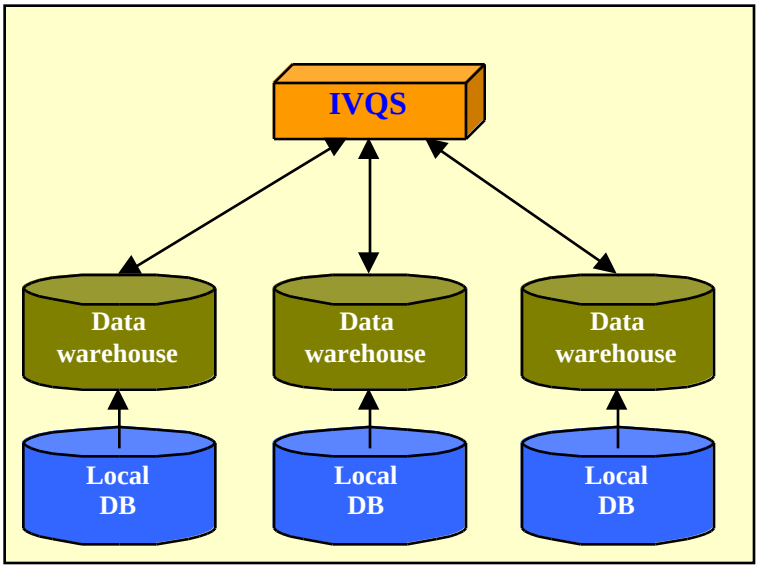


Fig.5 – IVQS as a tool for distributed datawarehousing

3.IVQS Architecture

The system external architecture is shown in fig.6. IVQS applet is stored in the server. When the client Web browser sends the request, the Web server sends back the applet, which starts being executed on the client.

The applet, because of its security limitations [12,17,18], can open socket connections only to the originating server. We need a server side application, which can access remote database and talks with the applet. We used Symantec DBAnywhere Server, which can access various kinds of RDBMS (Oracle 6.x-7.x-8.x, MS Access, Informix, etc.).

In this way the applet communicates with DBAnywhere Server to send query and to get back results.

Internally IVQS is structured as in Fig. 7. We'll now analyze each module:

- *Login/Logout.* This module has the duty to check user's authentication and to get user data from the module *User Data* and to dispatch them to *Query Manager*, *Favourite Queries* and *Dynamic Icons.*
- *User Data.* It has access to the system database for reading and writing user data. The database includes tables to store the user data in a proper way.
- *Query Manager*. This module takes care of handling queries in a tree-structure. Each query or folder is represented by an icon. Query icons can be double-clicked and sent to the module *Query Executor*. Query Icons can be created and edited through the module *Query Generator*. Each icon can be created and edited by *Icon Editor*.

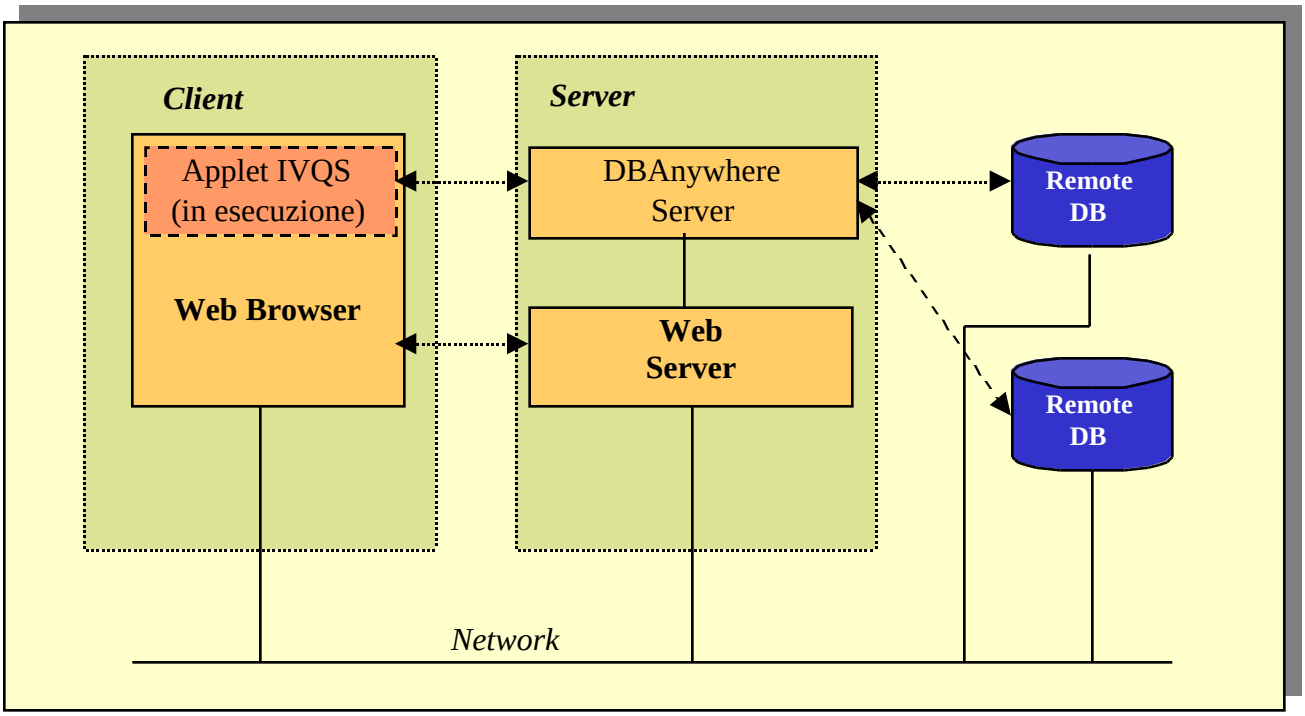


Fig.6 - IVQS External Architecture

- *Favourite Queries*. It keeps a collection of the user most favorite and used queries.
- *Dynamic Icons*. This module handles the entity-icons. Dragging an icon entity and dropping it on to another one, the associated query, if exists, is sent to *Query Executor*. The icon associated to an entity can be created and edited by *Icon Editor*.
- *Icon Editor*. It allows to create icons from scratch, or to import from the web specifying the URL.
- *Query Generator*. This module handles the query formulation. User can specify:
  - *Databases*
  - *Tables*
  - *Attributes to show*
  - *Sorting key*
  - *Conditions*

  The module then takes care of building the SQL string from those parameters.
- *Query Executor*. This module executes the query on the DB and dispatch the result to the module *Result Display*.
- *Result Display*. This module handles the result from a query. It allows two kind of visualization: table or form view. The user can customize the form layout in terms of font size, style and color and label position.

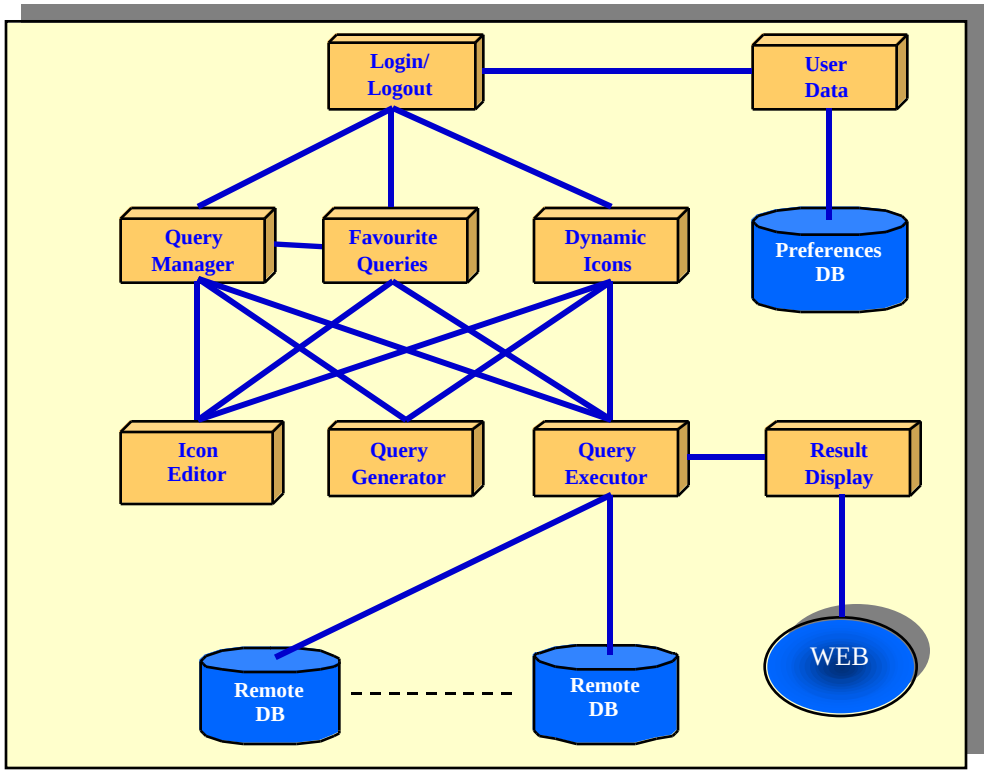


Fig.7 -IVQS internal architecture

4.Experimental Results

IVQS has been used on a set of existent databases, each one very different from the other in terms both of information and target users.

Here we list the databases where IVQS has been applied:

1. G7 Global Inventory Project (GIP). An Internet inventory of G7 IT projects. The target users are Internet ones.
2. Prosoma (PLUS), EU Web site, a multimedia database containing the most relevant project results in Europe. The target users are researchers, enterprises, technology operators and the EU.
3. Dying out animals database for primary school children within the Italian Internet-School Project.
4. Users Database of FAD (Long distance learning), ENEA web site for long distance learning on the Internet.
5. Cometa database. A material database designed for field experts.

In all these experiences, IVQS has been tested by the users to measure its usability [5]. The results we obtained, in terms of user satisfaction (fig.8) and user efficiency [9], showed that a query iconic representation allows a better perception in user's mind making it easier to memorize [6,7,8].

5.Conclusions and future work

The increasing number of Web users is looking for convenient ways to search information on the Net and to collect and query data from different sources through the Web [1,10,14,16]. We believe it's necessary to develop new alternative tools for such tasks.

We presented IVQS, a Visual Query System to query remote database through the Web using an iconic interface, developed and tested at ENEA Usability Lab in collaboration with University of Rome "La Sapienza". IVQS is a Java applet that allows the user to access remote databases through the Web with an iconic interface. That is alternative to the form-based web pages which are the most common interface nowadays.

The system has proved that an iconic interface is really easier to use for large end-user groups.

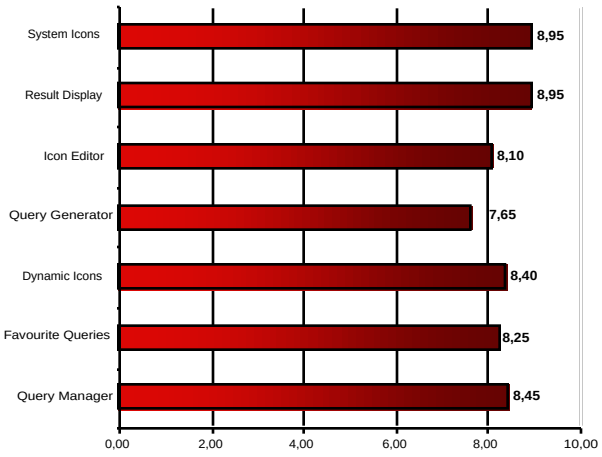


Fig.8 - User Satisfaction Grade

References

1. F.Capobianco, M.Mosconi, L.Pagnin. *Looking for convenient alternatives to forms for querying remote databases on the Web: a new iconic interface for progressive queries.* Dipartimento di Informatica e sistemistica - Università di Pavia, 1996.
2. S. Chaudhuri, U. Dayal, *An Overview of Data Warehousing and OLAP Technology,* ACM SIGMOD Record 26(1), March 1997.
3. F.M.Ferrara. *VENUS: Visual Enquiry User-oriented system. Technical Annex.* GE01-01-0002 , ENEA 1991.
4. F.Fontana, M.L.Bargellini, C.Bucci, F.Ferrara, P.A.Sottile, G.Scavino, R.Mancini, S.Levialdi. *Venus Project, Experimentation at ENEA's pilot site,* ENEA 1995.
5. International Standards Organization DIS 9241-11, *Ergonomic requirements for office work with visual display terminals (VDTs), Part 11: Guidance on usability.*
6. C.Jung. *The Man and his symbols.* Cortina editore, 1983
7. R.R.Korfhage, M.A. Korfhage. *The Nature of Visual Languages.* Proc. of 1984 IEEE Workshop on Visual Languages, Hiroshima, Japan, pp. 177-183
8. S.Levialdi, T.Catarci, S.Chang, M.F.Costabile, S. Levialdi, G. Santucci. *A Graphical Framework for Multiparadigmatic Visual Access to Databases.* Reprit, IFE Transaction on Knowledge and Data Engineerig, Volume 8 Number 3,28-42,1996.
9. M.MacLeod, R.Bowden, N.Bevan, I.Curson. *The MUSiC performance measurement method.*
10. A.Massari, S.Pavani, L.Saladini. *QBI: An Iconic Query System for Inexpert Users.* Dip. Informatica e Sistemistica, Università di Roma "La Sapienza", 1996
11. K. Mullet, D. Sano. *Designing Visual Interfaces,* SunSoft Press.1995
12. P.Naughton. *The Java Handbook.* McGraw-Hill, 1996
13. T. Nguyen, V. Srinivasan. *Accessing Relational Databases from the World Wide Web.* SIGMOD Conf. 1996.
14. C.Pancake, M.Newsome, J.Hanus. *Simplifying web access to remote scientific databases.* Oregon State University, 1997
15. B.Shneiderman. *Designing the User Interface*. Addison Wesley, 1994
16. B.Shneiderman, E.Tanin, R.Beigel. *Incremental Data Structures and Algorithms for Dynamic Query Interfaces.* University of Maryland, 1996
17. G.L.Steele Jr. . *Java and the Evolution of Web Software.* ICSE, 1997
18. A.S.Tanenbaum. *Computer Networks.* Prentice Hall,1996
19. H.Thimbleby. *User Interface Design.* Addison-Wesley.1991
20. Jennifer Widom, *Research Problems in Data Warehousing*, Int'l Conf. on Information and Knowledge Management 1995.
21. M.C. Wu, A.P. Buchmann, *Research Issues in Data Warehousing*, BTW'97, Ulm, March, 1997.