VisTool: a Visual Tool for Querying Relational Databases

Francesca Benzi, Dario Maio, Stefano Rizzi

DEIS, University of Bologna
Viale Risorgimento, 2 - 40136 Bologna - Italy
E-mail: srizzi@deis.unibo.it

ABSTRACT

In this paper we describe *VisTool*, a prototype tool for querying relational databases by means of the visual query language VISIONARY.

INTRODUCTION

VISIONARY [1] is based on an iconic-diagrammatic paradigm used for both data and query representation. The external data model is called vision and is made up of the visual primitives of concept and association; the external query model is based on the definition of a dynamic and adaptive perspective for accessing data. The internal data model is relational, and the internal query model is SQL. The mapping between the visual and the relational worlds is established through an intermediate graph-based level. VISIONARY is mainly directed to naive and occasional users, who have no experience with data models and only have a general idea about the contents of the database; for this reason, the expressive power of VISIONARY is limited to restricted join queries.

VisTool can be mounted on top of any relational DBMS supporting ODBC. The tool acquires an existing database scheme from the DBMS and assists an expert user in building a vision by allowing the proper definition of the logical and visual aspects of the different entities. Finally, a naive user can access and query the vision; the visual queries are translated into SQL queries and handed on to the DBMS, which executes them and returns the results to VisTool. The architecture of the system and its use by expert and naive users are sketched in Figure 1.

The design of VisTool was supported by the object-oriented design methodology OMT (Object Modeling Tecnique); the Windows95 implementation was carried out in C++ language. The interaction between VisTool and the underlying relational DBMS is based on the ODBC protocol.

THE VISIONARY DATA MODEL

The internal data model is relational; the external data model is based on an iconic-diagrammatic visual metaphor. An intermediate graph-based level establishes the mapping between the relational and the visual models.

The intermediate model is a non-directed graph, called *database graph*, whose vertices and edges correspond, respectively, to relation schemes of the database scheme and

to potential links between pairs of relation schemes, mainly determined by referential integrity constraints.
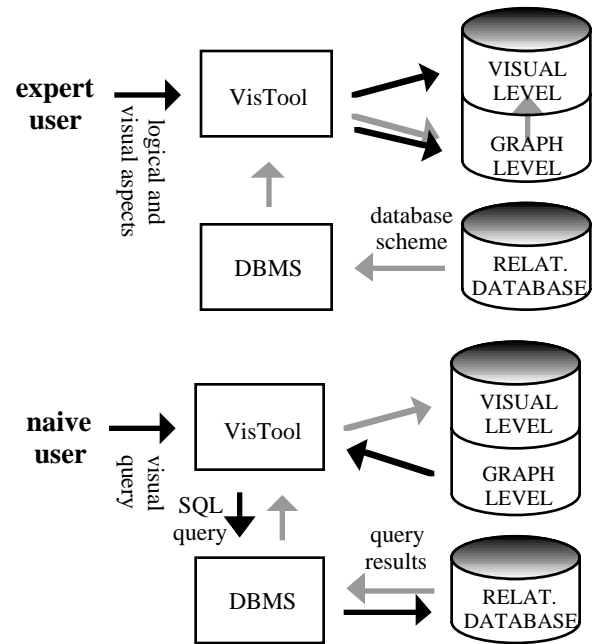

Figure 1: User interaction with VisTool.

The external model is called *vision*. A vision is a visual object aimed at conveying a high-level, clear and expressive representation of a relational database, largely independent of the structuring of data within the database schema. A vision is built by an expert user taking into account both the semantics of the database scheme and the needs of the inexperienced users who will access the database through the vision itself. A vision on a given database graph is a graph whose vertices and edges are, respectively, concepts and associations. Concepts represent relation schemes, while associations express relationships between two or more concepts.

A *concept* is defined by an icon, a name and a set of attributes, and corresponds to a relation scheme. It is represented visually by displaying its icon and its name; the attributes of the concept may be viewed by double-clicking the icon.

An *n-ary association* is defined by *n* names, *n* multiplicities

and a set of attributes, and corresponds to a set of potential links in the database graph. Each association is represented visually in one of its directions, by displaying an arrow connecting one end to the others, tagged with the name of the association when read in the direction of the arrow; the direction in which an association is displayed within a vision can be changed by right-clicking on the association. The attributes of the association may be viewed by double-clicking the association name; the user can further "explore" the association by left-clicking on it, which leads to visualizing the association multiplicity.

Two types of binary associations are treated differently from the others within the vision: PART-OF associations, which express the aggregation between concepts and are represented visually by framing them within a triangle, and IS-A associations, which express the specialization of a concept into one or more derived concepts and are represented visually by framing them within an ellipse.

THE VISIONARY QUERY MODEL

Naive users access the database through the visions previously defined by expert users. The intuitiveness and expressivity of visions help them understand the database semantics, and assist them in formulating queries without knowing the relational theory.

In our approach to query inference the user, when formulating a query, can specify a *primary concept* which defines a *viewpoint*, i.e., a perspective for accessing data. Within a viewpoint, the primary concept is connected to every other concept through exactly one path of associations; thus, in a query interpreted on a viewpoint, it is possible to reference attributes belonging to any concept without explicitly formulating the necessary joins. A default viewpoint is built automatically for each choice of the primary concept by privileging the query interpretations more frequently adopted by the user so far. The user may then modify this viewpoint by means of the visual language; in particular, (s)he may disable some associations, enable some others, and duplicate some concepts.

If the database graph is acyclic, the default viewpoint is univocally determined, and only one interpretation is possible for each concept. If the database graph is cyclic, a number of different viewpoints may correspond to each primary concept; in order to give exactly one interpretation of each concept, a criterion must be used to select one viewpoint so that the resulting interpretations are the most reasonable, that is, those which the user most probably expects. For this purpose, we estimate the *soundness* of the possible interpretations by considering the properties of the potential links involved; the default viewpoint associated to a primary concept is the one which maximizes the soundness for the queries formulated from that primary concept [2].

A viewpoint on a vision is represented visually by

1. highlighting the primary concept;
2. orientating and drawing in black each enabled association;
3. drawing in grey each disabled association.

Figure 2 shows the default viewpoint associated to primary concept *Product* on a vision representing product orders.

The formulation of a visual query consists of five steps. The first two are aimed at defining which joins will be *potentially* part of the query. The remaining steps, besides their specific functions, all go towards determining a multiset of concepts "mentioned" within the query. The associations belonging to the paths connecting the primary concept to each concept mentioned are called *active associations* and displayed in red within the viewpoint, to emphasize the fact that they determine which joins will be actually formulated for the current query.

In the following we outline the five formulation steps.

1. Choose a primary concept (implicit formulation of default joins). This step is executed visually by right-clicking on a concept; the default viewpoint is calculated and displayed.
2. Edit the viewpoint (implicit formulation of overridden joins - optional). This step must be executed if the enabled associations contained in the default viewpoint are not those the user is interested in. Suppose that the user wants to know, for each product, the orders signed by the vendors who sell the product. In the default interpretation proposed by the system with primary concept *Product*, *Order* denotes the orders made for a product; thus, the user must disable the association *belongs to* from *Product* to *Order* and enable the association *signs* from *Vendor* to *Order*. This action changes the interpretation given to queries by forcing the formulation of paths of joins different from those provided by the default viewpoint. Enabling a disabled association may lead to an attempt to form a cycle, that is, to having two enabled associations enter the same concept *c*; this happens in general when the user is interested in considering contemporarily different interpretations of *c*. In this case, concept *c* is duplicated within the viewpoint. Cyclic queries can be formulated visually by dragging and dropping one instance of the duplicate concept on the other.
3. Choose attributes to be retrieved (projection). This step is carried out by double-clicking on concepts and associations and by dragging one or more attributes inside the *projection window*.
4. Formulate selections on attributes (optional). A selection condition can be formulated by double-clicking on a concept or an association, dragging an attribute within the *selection window* and writing a Boolean predicate involving an operator and one or more values.
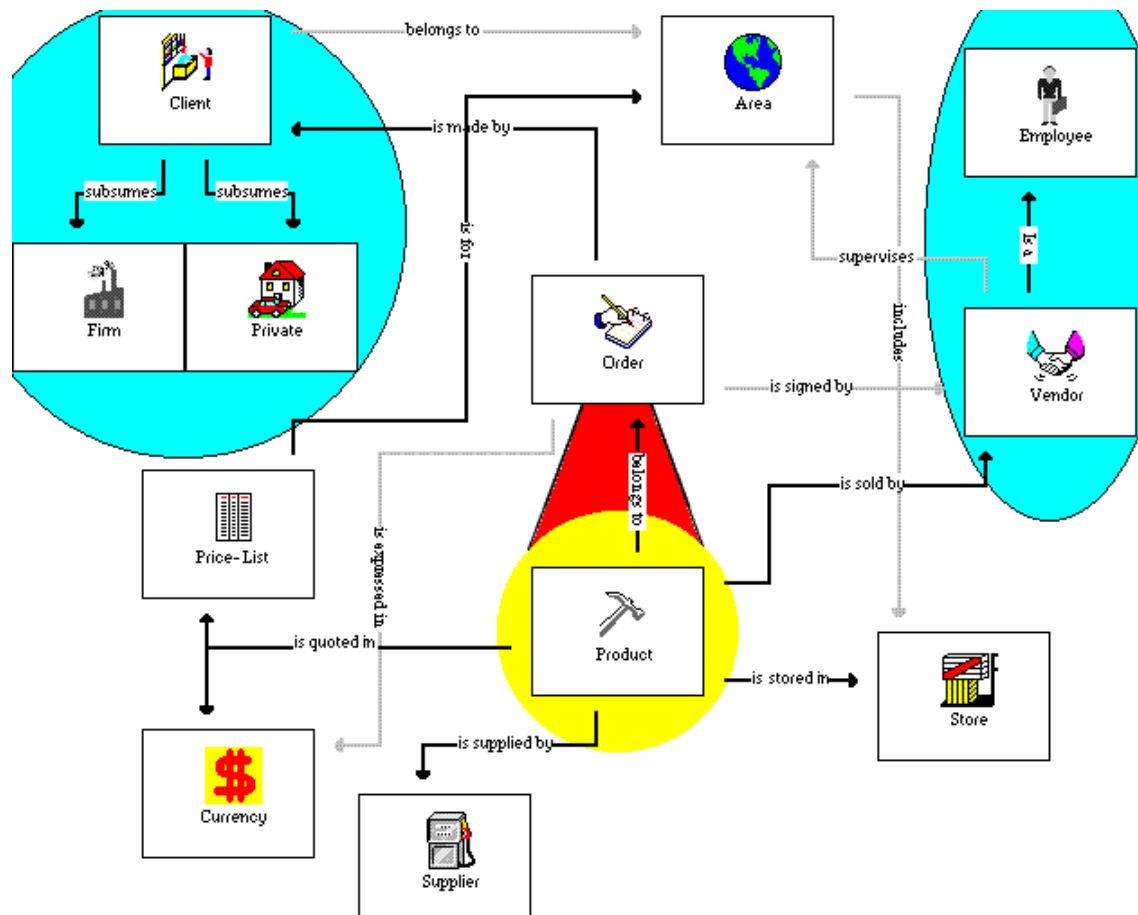
Figure 2: Default viewpoint corresponding to primary concept *Product*.

5. <u>Order and/or group the results</u> (optional). The last step may be performed on a *group-and-sort window* showing a preview of the results in tabular form. When choosing to order or group data according to a specific attribute, the user will see the results of his/her action on a sample set of data.

The SQL interpretation of each query is built on the viewpoint and handed on to the DBMS. Once the query has been executed, its results are shown to the user in tabular form.

CONCLUSIONS

In this paper we have described a prototype tool for querying relational databases by means of the visual query language VISIONARY. The user perceives the database through a metaphor called vision, which is made up of concepts and associations. Queries are formulated by choosing a primary concept, deciding the attributes to be retrieved and expressing selection predicates.

In [3], an experiment aimed at comparing QBI and QBD* in terms of their usability is described. The experiment consists in proposing six queries with different degrees of complexity to a set of users after briefly training them on the use of the visual language. We have reproduced the

same experiment with VisTool; the average formulation time with VisTool is about 90% that of QBI and 55% that of QBD*.

REFERENCES

1. Benzi, F., Maio, D. and Rizzi, S. VISIONARY: a visual query language based on the user viewpoint approach, in Proc. Third Int. Workshop on Interfaces to Databases (Edinburgh, July 1996), Electronic Series Workshop in Computing, Springer, http://www.springer.co.uk/eWiC /Workshops/IDS3.html.

2. Bellavia, G., Maio, D. and Rizzi, S. Minimizing the cost of query formulation through user viewpoint Relations, in Proc. Secondo Convegno Nazionale su Sistemi Evoluti Per Basi Di Dati (Rimini, Italy, June 1994), pp. 141-159.

3. Badre, A.N., Catarci, T., Massari, A. and Santucci, G. Comparative ease of use of a diagrammatic vs. an iconic query language, in Proc. Third Int. Workshop on Interfaces to Databases (Edinburgh, July 1996), Electronic Series Workshop in Computing, Springer, http://www.springer.co.uk/eWiC/Workshops/IDS3.html