SUPER - Visual Interaction with an Object-based ER Model

Annamaria Auddino, Yves Dennebouy, Yann Dupont, Edi Fontana, Stefano Spaccapietra and Zahir Tari
Ecole Polytechnique Fédérale - DI - Laboratoire Bases de Données
IN - Ecublens 1015 Lausanne Switzerland
auddino@elma.epfl.ch

Abstract
SUPER is a project aiming at the specification and development of a consistent set of visual user interfaces covering all phases of the database lifecycle.
In this paper we discuss the basic principles which, in our opinion, should underline a global approach to visual interaction with advanced data models. Visual interaction in SUPER environment is based on direct manipulation of objects and functions, providing users with maximum flexibility during schema definition as well as query formulation. Graphical interactions are easy to manage, and take advantage of the support of a simple but powerful modelling paradigm. Visual data manipulation is assertional and object-based. The environment offers multiple interaction styles, well-suited for various categories of users. Interaction styles are consistent over the various functions and editors.
To support the discussion, SUPER schema and query editors are analyzed, focusing on functionalities, and the underlying design choices, rather than precisely describing how they operate. An example of query formulation shows the rules used to govern interactions with users.

1 Introduction
Visual interaction had a drastical evolution during the eighties. WYSIWYG techniques (What You See Is What You Get) are nowadays standard for personal computing, while the WIMP metaphor (Windows, Icons, Menus, Pointing devices) governs user interaction with larger systems on workstations (and is moving into personal computing as well). Consequently, researchers try to master the many existing possibilities for human computer interaction. User Interface Management Systems (UIMS) are becoming popular as an answer to this question.
Despite this evolution, users of database management systems are still bound to classical textual languages, namely SQL. Although proposals for visual languages have been well known at least since 1975, thanks to QBE [37], research on visual interfaces still has to produce a global, recognized framework, consistent with the actual state of the art in data modelling techniques.
Indeed, graphical data definition techniques is the only area where a large consensus has been achieved on the marketplace. A number of tools exist, which offer graphical facilities for the definition of a database schema, according to concepts of the entity-relationship (ER) approach. They are eventually complemented with an automatic translation into a relational schema. Some tools also provide functionalities for describing application processes (usually through dataflow diagrams, sometimes with
Petri nets) and consistency checks. As far as we know, no commercial tool proposes a graphical manipulation language.

Several prototypes providing graphical DBMS interfaces have also been developed. Some of them only support graphical data definition: [8], [2]. DDEW [26] extends the definition process to all phases of database design, providing an integrated environment from user requirements to physical design. Some other tools provide both schema definition and visual querying facilities: ISIS [13], SNAP [6], [28], Pasta-3 [18, 19].

A few prototypes support visual data browsing, rather than query formulation: [21], ZOO [27], OdeView [1]. Finally, some prototypes only provide an aid for query formulation, to relieve users from constraints of textual syntax: [20], for instance, uses syntax graphs to guide users through the formulation of a relational query. Outside the scope of this presentation are toolkits for the design of graphical DBMS interfaces, like FaceKit [17], which belong to research in UIMS.

Existing prototypes can be classified according to the underlying data model (the following list of prototypes is not meant to be exhaustive):

- Entity-Relationship model: [8], [35], [36], [10], [21], [28], [26], [7], [14], [18], [9], [19];
- Object-Oriented model: [11], [27], [24], [17], [1];
- Semantic Data model: [16], [13], [6], [2];
- Relational model: [37], [15], [20], [23], [29].

The SUPER project is based on ERC+, an object-based extension of the entity-relationship model designed to support complex objects and object identity [33]. The goal of this project is to produce an integrated CASE tool (whose underlying model is ERC+) supporting interactions during all the life cycle of a database. To that extent, we first built a graphical definition and manipulation interface, providing users with a consistent approach to both functionalities. A data manipulation and a data browsing tool will deal with updates and navigation at the occurrences level. Moreover, a view definition tool will allow users to build views over an existing schema. Conversely, a view integration tool will allow to build an integrated schema from a set of user views. This last tool will be the kernel of a future database design tool, covering, as in DDEW, the various phases in this activity. At this purpose, some others tools are planned for schema normalization, restructuring and evolution.

This global approach will provide the user with the same interaction paradigms all along his/her dialogs with the different components of the DBMS, during both design and operation on the database. Moreover, ERC+ modelling provides the user of a database with the same objects as the real world of interest to him/her.

It should be clear that the main concern of SUPER is to design clear, clean, easy, precise and uniform user interaction methods. SUPER may be used as a front-end to a relational or an object-oriented DBMS. It is not our aim to develop it as a self-contained complete DBMS.

The next section discusses the basic principles governing the design of DBMS user interfaces, and shows which choices have been made in SUPER. Section 3 briefly recalls the characteristics of the ERC+ approach, which will be used in the sequel to illustrate the functionalities of the graphical interface. Section 4 presents the main characteristics of the data definition interface (the schema editor), while section 5 gives an overview of how users can define queries on the database. Finally, the conclusion summarizes the main features of the project and presents future and ongoing extensions.