

## THE "SUPER" PROJECT

ANDERSSON M., AUDDINO A-M., DUPONT Y., FONTANA E., GENTILE M., SPACCAPIETRA S.  
Database Laboratory, Departement of Computer Science, - LBD-DI-EPFL -  
Federal Institute of Technology  
CH-1015 Ecublens-Lausanne, Lausanne, Switzerland  
martin@lbdsun.epfl.ch

### Introduction

The SUPER project aims at the specification and development of a set of visual tools to support all phases of the database life cycle. Visual interaction in SUPER is based on direct manipulation of objects and functions, providing users with maximum flexibility and independence, during schema definition as well as query formulation.

The set of tools offers facilities to meet the varied demands from categories of users with different levels of skill. Diagrammatic representations and a basic set of functions are better suited for novice and occasional users, while menus and dialog boxes speed up the dialog for expert users. At the same time, a consistent interaction style over the various functions and editors has been emphasized. The global approach in SUPER is to provide the user with the same interaction paradigms all throughout his/her interaction with the different components, both during design and manipulation of the database. For example, the data manipulation is made by means of the same schema representation as the schema design.

SUPER has been designed as a front-end to a relational or an object-oriented DBMS, i.e., the persistence of data, consistency and concurrency problems, are delegated to an off-the-shelf database management system.

### The Prototype

SUPER is based on ERC+, an extension of the entity-relationship model, supporting complex objects, object identity and generalization. ERC+ modelling allows the user of a database to use concepts of the domain of discourse of interest to him/her. The ER model basis, with its limited and well known set of graphical symbols, is well suited for graphical schema representation. The additional

**Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.**

SIGMOD /5/93/Washington, DC,USA

© 1993 ACM 0-89791-592-5/93/0005/0546...\$1.50

object-oriented concepts provide for enhanced modelling power. ERC+ can be considered as belonging to the object+relationship modelling approach. The data model is complemented with formal manipulation languages: an ERC+ algebra and an ERC+ calculus. These are not intended for exposure to users, but as the underlying basis to more user oriented languages (visual or SQL-like).

The prototype provides multiple simultaneously working users with a consistent user interface and working approach throughout the activities of schema design, data browsing, data manipulation and query editing, each of these functions with a graphical and an alphanumeric interface.

The design editor is a visual interface for defining ERC+ schemas. It provides two modes of interaction, an alphanumeric mode and a graphical mode, each one with a separate display window. A display window is identified by the name of the schema being edited and labeled with the corresponding operation mode. In the alphanumeric mode, the user fills in form-like representations (called object boxes) of the different ERC+ concepts. In the graphical mode, the user builds an ERC+ diagram by direct manipulation. The user creates an ERC+ schema concept, by selecting an icon that is added to the schema in the workspace. Consistency according to the ERC+ inherent integrity constraints is maintained by the editor. The most important features of the design editor is flexibility, which allows each user to follow his own schema design strategy, reusability, which allows the user to benefit from previous work, and backtracking, which allows to undo erroneous actions. The editor contains "File" and "Edit" pull-down menus with the usual facilities. The "Display" menu gives access to various graphical facilities, the "Dictionary" menu provides for browsing the whole schema in search of schema items satisfying a given condition.

The data browser allows navigation in the database. In scrollable windows, the user can see the current object together with all objects related to it, and move from one object to another, within the same type or in a different type, according to navigational criteria (s)he defines at each step of the navigation. Objects are displayed according to the current format as defined by the user. The display

format can be dynamically changed during navigation. Generalization hierarchies may also be used to change the actual point of view on an object.

While browsing the database, the user can change values, insert objects or remove single objects or specified sets of objects. User maintained bookmarks provide comfortable shortcuts.

The visual query editor is based on the assertional approach, i.e., a user specifies the different components of a query independently of other components. The specification and execution of a query includes the following actions: selection of the sub-schema relevant for the query, transformation of this sub-schema into a hierarchical structure, specification of predicates, definition of the resulting type, and presentation of the result. The process is all visual, including the predicate specification.

Another possibility to construct queries is to use a context-sensitive query editor. The editor guides the user by presenting the current choices in the specification of an algebraic expression, including the operands that can be used in the current expression. This is done by presenting the content of the schema that is relevant for the operation that is to be performed.

#### **Future work**

Of particular interest among the tools currently under specification is the view integration tool which is used to build an integrated schema from a set of user views. This tool will constitute the kernel of a future database design environment. Also under specification are tools for view definition, schema normalization, schema restructuring and schema evolution.

Introducing SUPER as a commercial CASE tool on the marketplace is being actively considered. Interested parties should contact the database lab at EPFL.

#### **REFERENCES**

**Stefano SPACCAPIETRA, Christine PARENT (1992)**

*ERC+ : An Object based Entity Relationship Approach*, in *Conceptual Modelling, Database and Case: An integrated View of Information Systems Development*, Peri Loucopoulos and Roberto Zicari Eds., John Wiley, 1992

**Stefano SPACCAPIETRA, Christine PARENT, (1993)**

*View integration: a step forward in solving structural conflicts*, IEEE Transactions on Knowledge and Data Engineering, 1993

**Christine PARENT, Stefano SPACCAPIETRA, (1985)**

*An algebra for a general entity-relationship model*, IEEE Transactions On Software Engineering, Vol. 11, n° 7, July 1985

**Annamaria AUDDINO, Yves DENNEBOUY, Yann DUPONT, Edi FONTANA, Stefano SPACCAPIETRA, Zahir TARI, (1992)** *SUPER: Visual Interaction with an Object-based ER Model*, 11th International Conference on Entity-Relationship Approach, Karlsruhe, October 7-9, 1992, G. Pernul, A.M. Tjoa (Eds)

**Annamaria AUDDINO, Eric AMIEL, Yves DENNEBOUY, Yann DUPONT, Edi FONTANA, Stefano SPACCAPIETRA, Zahir TARI, (1992)** *Database Visual Environments based on Advanced Data Models*, International Workshop on Advanced Visual Interfaces, Roma, May 27-29, 1992

**Annamaria AUDDINO, Yves DENNEBOUY, Yann DUPONT, Edi FONTANA, Stefano SPACCAPIETRA, Zahir TARI, (1992)** *SUPER: A Comprehensive Approach to Database Visual Interfaces*, in *Visual Database Systems - II*, E. Knuth, L.M. Wegner Eds, IFIP Transactions A-7, North-Holland, 1992 (Proceedings of IFIP 2.6 2nd Working Conference on Visual Database Systems, Budapest, September 30-October 3, 1991)