The LOGRES prototype *

F. Cacace, S. Ceri, S. Crespi-Reghizzi,
P. Fraternali, S. Paraboschi, L. Tanca

Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza L. Da Vinci, 32 20133 Milano, Italy
{ceri,crespi,fraterna,parabosc,tanca}@ipmel2.elet.polimi.it

Logres is a new-generation database system integrating features from deductive and object-oriented databases [1, 2, 3, 4, 5]. The data model of Logres supports structural and semantic complexity through a rich collection of concepts from object-oriented models. The rule language allows for the manipulation of complex objects, the generation of new objects, and the definition of passive and active constraints. The application of set of rules to database states is controlled by means of qualifiers, which dictate the side effects of rules; qualifiers are the unique procedural feature of Logres, otherwise a fully declarative language.

The Logres prototype is built on top of Algres ([6]), a *Non-First Normal Form Relational Database*; the Algres language includes classical relational operators (selection, join, projection, etc.) plus aggregate functions, built-in predicates, nesting and unnesting operators, and a fixpoint operator. Algres stores the *Extensional and Intensional Logres Database* and provides the run-time environment of Logres applications; when an application is issued, the programming chain is the following:

1. The Logres application is the input of the *Logres Compiler*, which analyzes its correctness, detects and loads the part of the Intensional Database which is needed in order to execute the application, and finally produces a module, written in Alice (ALgres In C Embedded).

2. The *Alice Compiler* receives the module and translates it into C executable code (containing calls to the Algres run-time support). The application has now been translated into an executable format, and can be stored in a library of executable applications.

3. When the application is executed, the Logres application manager produces updates (when needed) to the Data Dictionary and to the Intensional Database and sends the executable code of the application to the *Alice Run-Time Support*, which is responsible for doing changes to the Extensional Databases and for producing query results.

The Logres compiler is composed by:

- The *Data Model Compiler*, which analyzes the correctness of the data structures introduced in the application with respect to the persistent data dictionary, produces schema-derived integrity constraints, and translates the Logres structures into Algres data structures.

- The *Rule Compiler*, which receives the syntactic tree and the integrity constraint rules generated by the data model compiler, loads part of the intensional database, and compiles/optimizes rules, producing an Alice module.

The Logres system has a user-friendly graphical interface, GRILLO, that allows syntax-directed editing of the Logres applications, and a graph-based visualization of the Logres Database schema; from the interface, it is possible to invoke the compilation and running of Logres applications, with windows displaying diagnostic messages and query results.

*This work was partially supported by the project LOGIDATA+, of the National Research Council of Italy, and by the Esprit project STRETCH, sponsored by the EEC.

SIGMOD /5/93/Washington, DC,USA
© 1993 ACM 0-89791-592-5/93/0005/0550...$1.50

References

[1] Cacace, F.: "Implementing an Object-Oriented Data Model in Extended Relational Algebra: Choices and Complexity", *Report n. 90-009*, Politecnico di Milano, January 1990.

[2] Cacace, F., S. Ceri, S. Crespi-Reghizzi, L. Tanca, R. Zicari: "Integrating Object-Oriented Data Modeling with a Rule-Based Programming Paradigm", in *Proc. ACM-SIGMOD Int. Conf. on the Management of Data*, SIGMOD 1990, Atlantic City, May 1990.

[3] Cacace, F., S. Ceri, S. Crespi-Reghizzi, L. Tanca: "Designing and Prototyping Data-Intensive Applications in the Logres and Algres programming Environment", in *IEEE Transactions on Software Engineering*, 18:6, June 1992.

[4] Cacace, F., S. Ceri, and L. Tanca: "Design and Implementation of Logres" *Report n. 92-040*, Politecnico di Milano, 1992.

[5] Cacace, F., S. Ceri and L. Tanca: "Consistency and non-determinism in a Database Programming Language", in *Proc. MFDBS Conference*, Rostock, LNCS 495, Springer-Verlag, May 1991.

[6] Ceri, S., S. Crespi-Reghizzi, G. Lamperti, L. Lavazza, R. Zicari: "ALGRES: An Advanced database system for complex applications", in *IEEE-Software* 7:4, July 1990.

Logres Demo

The demo shows an example concerning a knowledge base for automated tutoring of helicopter pilots [3]. The figure shows part of an application which queries the database for details about symptoms, fixing tools, and repair time of malfunctions in the helicopter engine.

Windows can be opened on Logres schemas, rules, and databases. The leftmost window shows the structure of three intensional relations representing the result of queries. These are expressed by means of rules; the upper window on the right shows the Logres rules defining the relations `query-time` and `query-tool`. The bottom window on the right shows part of the query answers.