2. Language pre-compiler. This subsystem parses an SPL program and sends information to the file manager for the internal storage of databases and programs.

3. File manager. It stores in internal structures the schema of a database, the objects that populate it, and the rules that manipulate them. Any update of the database is controlled by this manager.

4. Memory manager. When a program is executed, this subsystem builds in main memory the internal representation of the program rules and it keeps there the structure of the classes, as well as the collection of objects.

5. Inference machine. This subsystem keeps the control of the rule evaluation by means of the process described in paragraph 4.2. It performs the processes of unification and application of inference methods when a program is evaluated. It is also responsible of structured types manipulation (lists, tuples, etc.) when they appear in rules.

The inference machine and the memory manager are in continuous communication between each other. If a rule evaluation requires to create, update or delete an object, this is done immediately in connection with the memory and file subsystems.

7 Conclusions and Future Work

In this paper, we have presented a new deductive and object-oriented database management system, SABDOOD, whose implementation is already underway. One of the principal contributions of our proposal is to present a general purpose programming system for real databases (not only in main memory but in secondary storage), where we integrate an object-oriented data model with a declarative deductive language. With this system, we have a single paradigm to define and build deductive object-oriented databases, and to manipulate them declaratively by means of applications.

Moreover, SABDOOD, in contrast to other DOOD systems, provides a comprehensive language that supports the implementation of complex queries in a concise manner without having to translate them to associated host languages.

The current version of the system is intended for a single user; however, this version is only the first part of a more ambitious project. Next version of SABDOOD will consider among other issues, concurrency, graphical interfaces, client-server environments, etc. In summary, our last aim is the integration of multiple computational disciplines.

Acknowledgments

We would like to thank Mariana Díaz, Iván Hernández, Andrés Ramme and Carlos Solís, who are implementing SABDOOD subsystems as part of their thesis work, and Carlos Zozaya, Vicky Vergara and Leo Joskowitz for their helpful comments to improve the paper.
References


