AN INTERACTIVE INTEGRATED SYSTEM TO DESIGN AND USE DATA BASES

Antonio Albano and Renzo Orsini
Istituto di Scienze dell'Informazione
Università di Pisa
Corso Italia 40
I-56100 PISA, ITALY

Recent works on languages for modeling complex data base application environments show overlapping issues with other research areas such as Artificial Intelligence and Programming Languages. Moreover, a lot of attention is nowadays given to another important field, the overall data base design process, which, as it will be shown, furthermore extends the above connections.

These researches, in fact, point out two aspects:

1. An important tool in data base design is the language used to specify the design, since its features influence the way designers perceive the application they are modeling. The features of the language must be such that the produced description is easy to be understood, used, verified, and manipulated.
2. The language efficacy depends also on the existence of a set of methodologies and automatic tools available during the data base design process and its cycle life [New Orleans 79, NYU 78].

However there is another aspect, which, in a way, appears to be neglected, while it presents a remarkable interest in this context:

3. The tools and the language must be strongly integrated in a unique interactive programming system in order to provide an effective user-designer environment. This coexistence appears as fundamental in those applications which require frequent maintenance to adapt the data base to the new needs and to accomodate changes, but it is also important to design and debug complex applications that are relatively static.

Keeping in mind these aspects, interesting overlapping issues appear with the field of Software Engineering. In fact, current researches in this area point out the importance of interactive, integrated tools to support software production methodologies in order to bring down the complexity of designing, developing, testing and maintaining large software systems. Moreover, in order to fully exploit the facilities offered by an integrated programming system, according to the most accepted approach, a unique language is provided, with different levels of extension, both to write programs and to implement or to personalize the tools for their development [Cheatam 74, DOD 79].

Currently we are engaged in a project to design and to implement an interactive integrated system to design and use data bases. The project is related to another one which is currently under development in Pisa with the objective to build a proto-

type interactive integrated programming system to support program design, development and analysis [Asirelli 79, Degano 79].

The system is based on the following design choices which involve topics relevant to the workshop:

1. There is an unique language, with different levels of extension, to colect user requirements, to design and access the data base. The language is intended to be used by experts for designing both the applications and the tools for the interactions.

2. The system will be used during the design process with two kinds of perspectives:
   a) to execute the formal specification in order to checkout the design on test cases;
   b) to retrieve information on different aspects of all the kinds of objects which have been described.

3. The design language has features to deal with different aspects of dynamic and complex data base applications and it belongs to the category of so-called "Semantic Data Models". Particularly, it is "object" and not "name" oriented, and it allows the description of the following facts [Albano 79]:
   a) Data, defined both declaratively with abstraction mechanisms (aggregation, generalization and collection) and procedurally.
   b) Semantic integrity constraints, both standard (unique identification, value dependencies) and described by a general assertion mechanism.
   c) Operations, to describe the behavioral semantics both of the objects and of the application being modeled.
   d) Activation of operations on data base in response to a detected situation (process triggering mechanism).
   e) Incapsulation of definitions, to structure the schema in levels of abstraction with a mechanism, called "environment", which has certain similarities both to the to the "context", as used in Artificial Intelligence, and to the "module" in Programming Languages. Our application of this mechanism to conceptual modeling is novel in several aspects. (i) We are interested in using it as a organizational principle in order to partition data and procedures for area of interest. (ii) To define application oriented views of the data base. (iii) To define new environments in terms of given ones, using a set of predefined operators which are similar to those used by Burstall and Goguen [77] in their specification language CLEAR.

   All the descriptions are objects of the language and, therefore, they can be manipulated. This features allows the use of the language in designing new tools which operate on the specifications.

4. The system integrates both tools peculiar of a software development environment and others more specific for data base designing. According to their use, they can be classified in the following categories:

   Static analysis.
   a) Traditional and syntax driven editors, to create and modify definitions.
   b) Static semantics analyzers, to check the definition of the objects and their use in the procedures.
   c) Documentation generators, to display definitions, to produce cross-references among data and procedures, etc.

   Dynamic analysis.
   a) Interpreter, to inspect the behavior of the specifications on user stimuli.
   b) Interactive monitor and debugger, to discover and correct semantic errors at execution time.

   Schema manipulation tools.
   a) To support methodologies to give the specifications incrementally, with a

top-down or a bottom-up approach [Baldissera 79, Batini 80].

b) To assist in transforming the schema, in order to reformulate the specifications in terms of a subset of the language, which can be efficiently implemented. Examples of these transformations are those reported in [Balzer 80].

c) To support the implementation of the nucleus of the language, which include only objects and procedures definitions, in terms of a DBMS target, if this is not integrated in the system.

ACKNOWLEDGEMENT

This work has been partially sponsored by the "Progetto Finalizzato Informatica" of the "Consiglio Nazionale delle Ricerche".

REFERENCES

Albano A. and R. Orsini [80], "Un formalismo per la definizione di schemi concettuali", Rapporto Tecnico STMDB-RT-79-ISIPI-2, Pisa 1979.

Asirelli P., P. Degano, G. Levi, A. Martelli, U. Montanari, G. Pacini, F. Sirovich and F. Turini [79], "A Flexible Environment for Program Development on a Symbolic Interpreter", Proc. 4th Int. Conf. on Software Engineering, Munich, 1979.

Baldissera C., S. Ceri, G. Pelagatti and G. Bracchi [79], "Interactive Specification and Formal Verification of User's Views in Data Base Design", Proc. 5th Int. Conf. on VLDB, Rio de Janeiro, 1979.

Balzer R. [80], "An Implementation Methodology for Semantic Data Base Models", Proc. Int. Conf. on Entity-Relationship approach to System Analysis and Design, Nort-Holland, Amsterdam, 1980.

Batini C. and G. Santucci [80], "Top-Down Design in the Entity-Relationship Model", Proc. Int. Conf. on Entity-Relationship Approach to Systems Analysis and Design, North-Holland, Amsterdam, 1980.

Burstall R.M. and J.A. Goguen [77], "Putting Theories Together to Make Specifications", Proc. Int. Jt. Conf. on Artificial Intelligence, Boston, 1977.

Cheatham T.E. and J.A. Townley [74], "A Proposed System for Structured Programming", in (G. Goos and J. Hartmanis, Eds.) Lecture Notes in Comp. Science 19 (1974), Springer Verlag.

Degano F., G. Levi, G. Pacini, F. Sirovich and F. Turini [79], "An Integrated System to Support Program Design Development and Analysis", Rivista di Informatica IX, 4, 1979.

U. S. Department of Defense (DOD) [79], "Requirements for ADA Language Integrated Computer Environment, Preliminary "Stonemen" ", Nov. 1979.

New Orleans Data Base Design Workshop Report [79], IBM Report RJ 2554, 1979.

N.Y.U. Symposium on Data Base Design [78], New York, 1978.