The IMPRESS DDT: A Database Design Toolbox based on a Formal Specification Language

Jan Flokstra, Maurice van Keulen, Jacek Skowronek
Computer Science Department
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands
{flokstra,keulen,skow}@cs.utwente.nl

1   Prototype description

The Database Design Tool prototype is being developed in the IMPRESS project (Esprit project 6355). The IMPRESS project started in May 1992 and aims at creating a low-level storage manager tailored for multimedia applications, together with a library of efficient operators, a programming environment, high-level design tools and methodology. The DDT is part of this last effort.

The project focuses on the field of Technical Information Systems, where there is a need for tools supporting modeling of complex objects. Designers in this field usually use incremental design or step by step prototyping, because this seems to be best suited for users coping with complexity and uncertainty about their own needs or requirements. The IMPRESS DDT aims at supporting the database design part of this process.

2   TM and its role in the DDT

The DDT uses the object-oriented database specification language TM as its kernel[BaBV92]. TM has a complete formal semantics [BaBZ93]. This entails all expressions in the language having a precise and unique meaning, without which it is impossible to build a reliable toolset. A designer using TM need not necessarily have knowledge of TM's underlying formal basis to achieve correct specifications of TM schemas, but it is this basis that enables him to perform complex semantic analyses of schemas, thus paving the way to a complete debugging of the conceptual design. As a design language, TM is equipped with powerful structuring primitives which enable a user to arrive at natural and intuitively correct designs.

3   Functionality

The DDT consists of the following components:

- Graphical TM Interface for editing a specification using the TM diagram language.

- Typechecker for TM specifications (TM is statically typecheckable)

- Safeness Detector for checking a specification for the presence of unsafe expressions.

- Prototyping Environment for doing simulation experiments in a generated prototype database.

The following components are being developed:

- Constraint Analysis for semi-automatic verification of correctness-preserving properties after invocation of database transactions using an existing proof checker.

- Documentation facilities based on WEB.

References

[BaBa93] R. Bal & H. Balsters, "A Deductive and Typed Object-Oriented Language", in *Third International Conference on Deductive and Object-Oriented Databases*, December 6–8, 1993, Scottsdale, Arizona, USA.

[BaBZ93] H. Balsters, R. A. de By & R. Zicari, "Typed sets as a basis for object-oriented database schemas", in *Proceedings Seventh European Conference on Object-Oriented Programming*, July 26–30, 1993, Kaiserslautern, Germany, 1993.

[BaBV92] H. Balsters, R. A. de By & C.C. de Vreeze, *"The TM manual"*, University of Twente, technical report INF 92–81, Enschede, 1992.

[BaFo91] H. Balsters & M.F. Fokkinga, "Subtyping can have simple semantics", *Theoretical Computer Science 87 (September, 1991)*, pp. 81–96.

[BaVr91] H. Balsters & C.C. de Vreeze, "A semantics of object-oriented sets", in *The Third International Workshop on Database Programming Languages: Bulk Types & Persistent Data (DBPL-3)*, August 27–30, 1991, Nafplion, Greece, P. Kanellakis & J.W. Schmidt, eds., Morgan Kaufman Publishers, San Mateo, CA, 1991, pp. 201–217.