

The ECRC Multi Database System

Willem Jonker*
PTT Research, PO Box 15000,
9700 CD Groningen, Netherlands
W.Jonker@research.ptt.nl

Heribert Schütz
ECRC GmbH, Arabellastr. 17,
81925 München, Germany
schuetz@ecrc.de

The novel feature of the ECRC Multi Database System is the use of an extended integrated deductive and object oriented data model (DOOD) as the common data model. This as opposed to an extended relational or object oriented model used in other systems. It supports the federated database architecture proposed by Sheth and Larson [3].

Basic Data Model

ECRC's multi database system is an extension of the Chimera DBMS prototype developed in the Esprit project IDEA. Chimera provides a deductive and object-oriented data model [1]:

```
class employee
superclasses person
attributes
  manager: employee extensional;
  dependents: set employee
  [ X in dependents ← X.manager = self.
    X in dependents ← X.manager in dependents. ];
end
```

Employees have by inheritance all the attributes of persons and can also be used as persons. Furthermore for every employee a reference to a manager is stored in the database. A set of dependents is derived on demand according to the deduction rules. Intensional attributes like “dependents” take the role of methods in other object-oriented data models.

Extensions

Extensions support access to distributed data and integration of heterogeneous schemas and data models [2].

* Part of the work described here was carried out while the author was at ECRC.

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 of Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
SIGMOD '95, San Jose, CA USA
© 1995 ACM 0-89791-731-6/95/0005..\$3.50

Modules: Modules facilitate logical and physical data separation. Classes live in modules. They are made part of a module's interface by *exporting* and made accessible in another module by *importing*. Communicating modules can live in different databases. The location of modules is transparent to application programmers and almost transparent to schema designers.

Virtual Classes: A virtual class is a view on another class, its “base class”. The objects of the base class also belong to the virtual class, but appear to have a different structure. All attributes of a virtual class are intensional. A pseudo attribute **original** refers to the current object seen as a member of the base class.

If the objects of class person have an attribute providing the person's year of birth, then a view on persons providing the age is defined by:

```
class my-person on person
attributes
  ...
  age: integer [age := 1995 - original.year_of_birth.];
end
```

A virtual class with a superclass is used to include an *existing* base class into the superclass, thereby adapting the schema appropriately.

Relational Database Access: Data stored in a relational database can be made visible in a Chimera module. Tuples can be converted into objects automatically.

References

- [1] S. Ceri and R. Manthey. Consolidated specification of Chimera. Idea project report, Politecnico di Milano, Italy, Nov. 1993.
- [2] W. Jonker and H. Schütz. Support for schema management—tools functional specification. Idea project report, ECRC GmbH, Munich, Nov. 1994.
- [3] A. P. Sheth and J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236, Sept. 1990.