EER Designer for Deductive Databases

José R. Paramá Gabía

Nieves R. Brisaboa

María F. Becerra

LFCIA (Laboratorio de Fundamentos de Computación e Intelixencia Artificial)

Facultade de Informática, Universidade da Coruña

15071 A Coruña, Spain

Fax: (981) 132736

brisaboa@udc.es

Abstract

A tool called *Entity-To-Deductive (ETD)* for helping Deductive Database (DDB) design and implementation is introduced. Users will design the conceptual model of the database using the interface ETD which is based on Enhanced Entity Relationship Model (EER) diagrammation system. Analysing this design, ETD generates a set of modules for a Deductive Database System, CORAL, which implements the constraints expressed in the EER model such as the entity and referential integrity constraints, the structural constraints of cardinality and participation in a relationship, and the constraints on specialization and generalization of disjointness and completeness.

1. Introduction.

Users will design the conceptual model of the database using the interface that ETD provides which is based on the Enhanced Entity Relationship Model (EER) diagrammation system. ETD uses transformation rules for the constraints expressed in the EER model to translate them into CORAL modules. The rules are valid for any DDB.

Some of these modules run when a new fact is introduced, such as the module which implements the entity integrity constraint. But other modules are evaluated periodically and at the end of the session for instance the module that checks the total participation of an entity in a M:N relationship.

2. The Tool.

ETD is being developed on a *Client-Server* architecture. The client provides a *graphical interface* to design the EER model and the *translator-generator* of the CORAL modules, whereas the UNIX server is the part of ETD which gives access to the deductive database CORAL.

3. Examples of Translation from EER Model into Deductive Model.

For each entity in the EER model, one CORAL rule is generated to check whether *the entity integrity constraint* is maintained. This rule is executed when a new fact of the p predicate is introduced.

```
int_ent_cons_p(X):- not (p(X,Y), p(X,Z), Y<>Z), X<>null
```

The p predicate represents the entity; X is the term which represents the set of attributes that composes the entity identifier; Y and Z are the terms that represent the other attributes. The result of executing this rule is true if the integrity constraint is maintained and false otherwise.

In case of $1:1$ and $1:N$ relationships, the predicate that allows the evaluation of the referential integrity constraint is defined through the following CORAL module:

```
module ref_int_cons_q.  
  ref_int_cons_q (X2, X1) :- q(X2, Z, X1), p(X1, Y).  
  ref_int_cons_q (X2, X1) :- X1 = null.  
end_module.
```

The result of executing this module is true when the referential integrity constraint is maintained and false otherwise. This module is executed when a new fact of the q predicate is introduced in the database.

Interfaces to Databases (IDS-3)