ACTIVE MECHANISMS FOR CHECKING PARTICIPATION CONSTRAINTS IN UML

Djamel Berrabah, Charles-François Ducateau
CRIP5, Paris5 University, 45 rue des Saints Pères, 75270 Paris cedex 06, France

Faouzi Boufarès
LIPN, Paris13 University, 99 avenue J.B.Clément, 93430 Villetaneuse, France

Keywords: Participation constraints, integrity constraints, triggers, active databases.

Abstract: The automation of the database design process using CASE tools is among the multiple efforts devoted to face the problems of database modeling. These tools often do not take into account all information presented in a conceptual schema. Consequently, the relational elements obtained during these processes do not coincide completely with the conceptual elements, and that produces some semantic losses. The idea is to enrich these tools and to improve them in order to solve some problems of modeling. The goal of this work is to propose an efficient approach to generate mechanisms that preserve the participation constraints during the transformation of a conceptual schema into a relational one.

1 INTRODUCTION

Database (DB) design methodologies (Elmasri, 2004, Toby, 1999) present processes devoted to translate a conceptual schema (CS) into a relational one (RS). The relational elements obtained during these processes do not coincide completely with the conceptual elements, and that produces some semantic losses (Boufarès, 2005). This problem often arises when most of the constraints established in the CS are not translated correctly. Among these constraints, we find "multiplicity constraints". This type of constraints was discussed in (Boufarès, 2001). He describes how to use assertions in order to check minimum and maximum multiplicities. (Al-Jumaily, 2004) uses triggers to preserve the minimum multiplicities. Another type of constraints is concerned; they are "participation constraints" (PCs), also called "interrelationship constraints". They have dynamic aspects which must be translated in the RS such as some integrity constraints for checking the DB modification operations (Insert, Delete and Update). Today's most current commercial CASE tools, such as Power AMC (Sybase 2005) and Rational Rose (Rational 2005), do not fully take these constraints into account and only generate a significantly simpler database schema.

Our aim in this paper is to provide an efficient mechanism which deals with PCs automatically (to check and control them). These mechanisms consist in creating trigger systems. Thus, an automatic module to generate triggers has been thought to be a good idea to implement PCs defined in a CS and check them during DB manipulations. In a previous work, (Berrabah, 2005) used OCL (Object Constraint Language) (OMG, 2005) to translate PCs.

This paper is structured as follows. Section 2 introduces the basic principles of constraints and their role in preserving the semantics of the universe of discourse and provides an overview of active mechanisms. Section 3 presents the syntax and the semantics of participation constraints and describes how to transform a conceptual schema into a relational one. In section 4, the essential part of this paper, the trigger-based rules are considered to show how to generate active mechanisms for expressing and maintaining participation constraints. Finally, section 5 presents our conclusions and perspectives.

Berrabah D., Ducateau C. and Boufarès F. (2006).
ACTIVE MECHANISMS FOR CHECKING PARTICIPATION CONSTRAINTS IN UML.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - DISI*, pages 310-313
DOI: 10.5220/0002453903100313

2 CONSTRAINTS AND TRIGGERS

A constraint constitutes a condition or a semantic restriction, expressed in a linguistic instruction form, in a textual language. In general, a constraint is linked to one or several elements of the CS. It represents semantic information associated with these elements. A CS includes a set of all suitable constraints to represent correctly the semantics of the universe of discourse. These constraints have to be defined without **conflicts** (Boufarès, 2005). The graphic elements offered by the CASE tools do not allow expressing the totality of the constraints. In addition, no mechanism is generated to verify the global coherence of the expressed constraints.
Let us consider the participation constraints. Unfortunately, they are not expressed during the translation process, and that provide a loss of semantics. Triggers (Cochrane, 1996) constitute powerful systems to deal with these constraints. In SQL 2003 (Eisenberg, 2004), a trigger is expressed using event-condition-action (ECA) rules (Cochrane, 1996, Ceri, 1990, Horowitz, 1994). It is activated during DB transition state. Each trigger is associated to one or more events on a table. It is activated, if one of these events is performed on this table. Once the trigger is activated, its condition, that is an assertion on the data or the state of the DB, must be evaluated. If the condition is evaluated as "true", then the action is performed. An action is a sequence of SQL statements performed on the DB tables or a "raise error" which rejects the event that activated the trigger. If the event is rejected, the data of the DB do not change. Triggers can access to the old and the new attribute values affected by the triggering event and use them in SQL statements.

3 MAPPING PCS TO RELATIONAL DB INTEGRITY CONSTRAINTS

3.1 Participation Constraints in UML

A PC frequently relates to the coexistence of occurrences of class objects in one or several associations (Figure.1). In the literature, several participation constraints were presented such that exclusion, inclusion, simultaneity and totality. More detailed definitions of this category of constraints are given by (Berrabah, 2005, and Nanci, 2001). The introduction of these constraints into a CS must be taken into account in order to preserve the semantics of the real world. Consequently, PCs will be translated in the DB generation script.

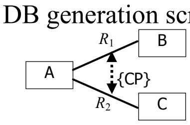


Figure 1: Participation constraints.

3.2 From Conceptual Schema to Relational One

Given the following transformation Rules:
Rule 1: Any class is transformed into a table with a primary key.
Rule 2: Any binary association which does not contain maximum multiplicity equal to 1 is represented by a table, the primary key of which is made up of both the primary keys of the concerned classes. These primary keys constitute foreign keys.
Rule 3: Any binary association with a maximum multiplicity equal to 1 is represented in the form of a foreign key.
The three classical types of couple of multiplicities considered are: 1) one-to-many: Only one multiplicity has a maximum equal to 1; noted 1-N or N-1, 2) one-to-one: Both multiplicity constraints have a maximum equal to 1; noted 1-1. This case is similar to 1-N one, 3) many-to-many: All maximum multiplicity constraints are not equal to 1; noted N-M.
Figure.2 summarizes the Relational sub-Schemas (RsS) associated with the CS1 according to the multiplicity constraints defined on its associations. Only the tables concerned by our study are taken into account in this figure.

Case	Association Type	relational Sub-Schema
Case I	N-1 & N-1	A(PKA,FKB,FKC, AttrA)
Case II	N-M & N-1	T(FKB, FKA, AttrR) A(PKA, FKC, AttrA)
	1-N & N-1	T= B(PKB, FKA, AttrB) A(PKA, FKC, AttrA)
	N-1 & 1-N	A(PKA, FKB, AttrA }) T= C(PKC, FKA, AttrC)
	N-1 & N-M	A(PKA, FKB, AttrA) T(FKC, FKA, AttrR2)
Case III	N-M & N-M	T1(FKB, FKA, AttrR1) T2(FKC, FKA, AttrR2)
	1-N & N-M	T1= B(PKB, FKA, AttrB) T2(FKC, FKA, AttrR2)
	N-M & 1-N	T1(FKB, FKA, AttrR1) T2= C(PKC, FKA, AttrC)
	1-N & 1-N	T1= B(PKB, FKA, AttrB) T2= C(PKC, FKA, AttrC)

Figure 2: Summary of the RsS associated to CS1 according to the various multiplicity constraints.

Where PKX means the Primary Key of the table X, FKX means the Foreign Key of the table X, AttrX means the Attributes referencing the table X and AttrR means the Attributes of association R.
Figure 2 shows that only the considered tables. The latest represent the binary associations on which the PC is defined. Three different cases are distinguished. Case I) The objects of both associations appear in the table A. Case II) The objects of only one association appear in the table A. Case III) No objects of either association appear in the table A.

4 APPROACH DESCRIPTION

This section presents how to translate PCs using triggers. These triggers are represented in a form of ECA rules. The study is done according to the three cases shown in the previous section (figure.2). At each case, us an example, the ECA translation rules of the exclusion PC is given. As a case study, an application example, related to that process in ORACLE DBMS, will be considered. The transformation is made in trigger-based SQL scripts.

4.1 Translation Rules of Participation Constraints

Case I

This case represents the N-1 & N-1 association types. Thus associations R1 and R2 are both translated by the migration of the primary keys of classes B and C respectively as foreign keys (FKB and FKC) in the table A (Figure.2). With this solution, all the objects of both associations appear in the table A.

Example 1

In this case, the exclusion constraint is violated only if an A-object participates in an association while it already participates in the other one. This can occur during an insertion or an update operation. In order to resolve the problem it is necessary to generate a trigger that reacts to these events on the table A. The deletion operation has no effect on this constraint.

```
event: insert or update on A
condition: new value of FKB is not null
       and new value of FKC is not null
action: raise error
```

Case II

In this case one of the two associations is translated by a foreign key (FKB or FKC) in table A (Figure.2). The A-object participation in one of the two associations appears in table A. The tables taken into account in this case are classified in Figure.2 Case II.

Example 2

In Case II three events can violate the exclusion constraint. These events are an update on table A, an insertion or an update on table T. Two triggers must be generated to prevent the violation of this constraint. The first reacts to an update on table A. Its principle is to reject this update if the new value of the foreign key in table A is different from the null value and the value of PKA already exists in table T.

```
event: update on A
condition: new value of FKB is not null
     and the set of rows that T.FKA=
     A.PKA is not empty
action: raise error
```

The second trigger reacts to an insertion or an update on table T. Its principle is to reject these two events if the value of FK, with which the new value of FKA is associated, is different from the null value.

```
event: insert or update on T
condition: new value of FKA is not null
     the FK value is not null where
     A.PKA=T.FKA
action: raise error
```

Case III

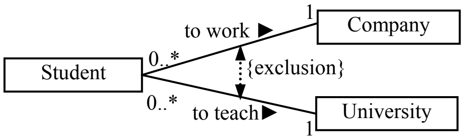
In this case none associations will be translated by a foreign key in the table A i.e. the participations of A class objects will not appear in the table A. Let us consider two tables T1 and T2 which represent respectively the transformation of the associations R1 and R2. The tables taken into account in our study are classified, in the table above, according to the association types (Figure.2 Case III).

Example 3

Four events can violate the exclusion constraint in Case III, an insertion or update of the table T1 and an insertion or update of the table T2. Therefore, two triggers must be generated in order to control this constraint. These two triggers have the same principle. The one on the table T1 (resp. T2) rejects the events (INSERT/UPDATE) if the new value of FKA already exists in the table T2 (resp. T1).

```
event: insert or update on T1
condition: new value of FKA exist in T
action: raise error
```

4.2 Application Example



In this example, a student either teaches at the university or works in a company but not both at the same time. He may not do either. To ensure this condition it is necessary to add an exclusion constraint between the associations "to work" and "to teach". "to work" and "to teach" are both many-to-one associations (N-1 & N-1). The trigger generated in this case is as follow:

```
Create trigger Insert_Update_Student
Before insert or update on Student
Begin
    If (NEW.FK_COMP IS NOT NULL AND
        NEW.FK_UNIV IS NOT NULL)
    Then RAISE_ERROR ('exclusion constraint
        violated');
    End If;
End Insert_Update_ Student;
```

5 CONCLUSION

In this paper, we reported a systematic study of the use of PCs for the specification of assertions defined on the behavior of class object participation into two associations. Sometimes, it is necessary to use these constraints in the CS. Thus, we have provided a general framework for transforming PCs into trigger-based SQL scripts. It turned out that triggers are particularly adequate for this purpose. They are being used in a lot of significant ways in current database systems and applications.

We are implementing a prototype as an add-in module for checking PCs. The verification of these constraints is not easy. Therefore, we think that incorporating add-in modules is a good idea to solve PCs modeling problems, and to improve the quality of CASE Tools.

REFERENCES

Al-Jumaily, H.T., Cuadra, D., Martinez, P. "Plugging Active Mechanisms to Control Dynamic Aspects Derived from the Multiplicity Constraint in UML", The workshop of 7th International Conference on the Unified Modeling Language, October 2004, Portugal.

Berrabah, D., Boufarès, F., Ducateau, C.F. "Analysing UML Graphic Constraint, How to cope with OCL", 3rd International Conference on Computer Science and its Applications, June 2005, California, USA.

Boufarès, F., Berrabah, D., Ducateau, C.F., Gargouri, F. "Les conflits entre les contraintes dans les schémas conceptuels de Bases de Données : UML - EER", Journal of Information Sciences for Decision Making, Special Issue of The 8th MCSEAI'04, 2005, N°19 Paper number 234.

Boufarès, F. "Un outil intelligent pour l'analyse des schémas EA". Interne Report. Informatics Laboratory of Paris Nord. University of Paris 13. 2001. France.

Ceri, S. and Widom, J. "Deriving production rules for constraint maintenance". In Proc. of the 16th ht. Conf. on Very Large Data Bases, pages 566-577, Brisbane, Australia, August 1990.

Cochrane, R.J., Pirahesh, H. and Mattos, N.M. "Integrating triggers and declarative constraints in SQL database systems". In Proceedings of the Twenty Second International Conference on Very Large Data Bases, pages 567--578, Mumbai, India, September 1996.

Eisenberg, A., Melton, J., Kulkarni, K., Michels, J., Zemke, F. "SQL: 2003 has been published", ACM SIGMOD Record, Volume 33, Issue 1, March 2004.

Elmasri, R., Navathe, S. "Fundamentals of Database Systems ", Fourth Edition, Addison-Wesley, 2004.

Horowitz, B. "Intermediate states as a source of non deterministic behavior in triggers". In Fourth International Workshop on Research Issues in Data Engineering: Active Database Systems, pages 148-155, Houston, TX, February 1994.

Nanci, D., Espinasse, B. "Ingénierie des systèmes d'information : Merise deuxième generation", (4th edition)". Edition-Vuibert, 2001.

OMG, editor: "UML 2.0 Object Constraint Language Specification", OMG, 2005. http://omg.org.

Rational. http://www-306.ibm.com/software/rational/sw-bycategory/subcategory/SW710.html

Sybase. http://www.sybase.com/products/information management/powerdesigner.

Toby, J. Teorey, "Database Modeling & Design", Third edition, Morgan, Kaufmann Series in data management systems, 1999.