as a solid arrow with a message name (2:*DC_PROF*). The function *TriggerSet*(*Belongs*, *DELETE*,*Before*) is called again to calculate the new *TG*(). In this case, *Belongs* has one Before trigger which is represented in the sequence diagram as a *MessageToSelf* with a message label (3:*T*1(*B*/*R*)). If *TG*() has an instance of *T*1(*B*/*R*) then a non termination state is detected and a message is sent to the user. In any other case, a new instance in *TG*(1) = {*T*1(*B*/*R*)} is created. According to the SQL:2003 recommendation Before triggers is used to read from a database or to correct an error produced in the processing of data input (see section, **Activation Time**) therefore, we do not need to check whether the action of *T*1(*B*/*R*) may produce new events. The event *cascade*(*DELETE*) is applied to *Belongs*. Then the function *TriggersSet*(*Belongs*,*DELETE*,*After*) is called to calculate the new *TG*(). There is only one After trigger defined on *Belongs:* (4:*T*2(*A*/*S*)) represents this trigger. If *TG*() has an instance of *T*2(*A*/*S*) then a non termination state is detected, otherwise a new instance in *TG*(2) = *T*2(*A*/*S*) is created. If a new event is issued from the action of *T*2 the algorithm should be repeated again calling the function *TMapping* with the pair (new event, table) as parameters. If there is not any cascade event in the trigger action then the algorithm is finished and the termination is verified by sending a message to the user.

4.3.2.2. Scenario 2: (Fig. 10B). In this scenario the action of *T* is modified to incorporate the event *DELETE* from *Professor*. In order to avoid repetition, in this scenario, the sequence of operations is similar to the previous one until it reaches the message (4:*T*2(*A*/*S*)). Until now, the trigger set *TG*() has two instances {*T*1(*B*/*R*), *T*2(*A*/*S*)}. Because the action of *T*2 has the new event *DELETE* from *Professor,* the function *Sub TMapping*(*Professor*,*DELETE*) is called again which immediately calls *Sub TriggersSet*(*Professor*,*DELETE*, *Before*) to calculate the new instances in *TG()*. Because there is not any Before trigger applied to *Professor* the function *Sub TriggerSet* is finished, and *TG*() is returned with only the previous instances. At this point, the algorithm checks again whether there is a cascade event produced by *(5:DELETE)* or not. The referential action (6:*DC_PROF*) is executed on *Belongs*. Then the trigger (6:*T*1(*B*/*R*)) is fired and added to the triggers set as *TG*(3) = *T*1(*B*/*R*). Now, the *TG*() has three instances {*T*1(*B*/*R*), *T*2(*A*/*S*),*T*1(*B*/*R*)} this means that there are two instances which have been applied to the same object. In this case, a non termination state is detected and a message is sent to the user to warn him about the existence of this problem. When a non termination is detected the mapping is finished immediately.

5. Conclusions

Although the database CASE tools have been developed to resolve the database modelling problem and to provide automatic processes to develop all phases supported in a database methodology, the current state of these tools is that they provide conceptual models with more abstraction and are concerned with expressing the semantics of the real world more accurately. However, the move from the conceptual level to the logical level is not supported by these tools, and the generated code needs to be modified to comply with the requirements of the real world.

It is true that various studies have lead to important results such as the creation of the current commercial CASE tools and some research prototypes to support maintaining mechanisms to preserve integrity constraints in the logical models. Nevertheless, in the context of Relational databases we consider that current practice is below the needs of the requirements of active technology. These requirements need to have a verification process which is considered as important as development.

On the other hand, although the Relational database has been widely used in the commercial DBMS and the most important commercial Object Oriented database systems utilize the Relational tables to store objects, we consider that most proposals have been developed to respond to the needs of Object Oriented databases development.

Therefore, to fill in some of the gaps that the current CASE tools leave during the development of active Relational Databases, we present the OCL2Trigger tool as a support to the theoretical approach which follows the phases proposed in the MDA software development, by completely transforming the OCL constraints into triggers. These phases are as follows: specifying OCL constraints in the UML class diagram, transforming the OCL constraints into SQL:2003 standard triggers, transforming the standard triggers into target DBMS triggers. In addition, this tool can represent and verify trigger execution by using UML sequence diagrams. Thus, this work unites the UML aspects that are widely accepted and is supported by many CASE tools for aspects of Relational databases that have wide presence in commercial DBMS.

Our approach has some limitations which are explained as follows: (a) although we believe that applying MDA makes the transformation of any type of OCL constraints to triggers easier, currently the OCL2Trigger tool supports only the OCL invariant constraints. Specifically, three patters have been proposed: attribute value constraints, multiplicity constraints and generalization constraints. Other types of constraints such as aggregations and compositions, pre conditions, and post conditions will be included in future work; (b) Including complex OCL expressions in which many relations are involved may result a difficult task to generate triggers. We think that this limitation could be solved by incorporating more patterns to our approach to cover such expressions. The article presents a first effort to check the viability of this approach through three of the most widely used constraints in the conceptual model; (c) The triggers execution analysis focuses only on detecting the non termination problem and the user himself needs to redefine and reconstruct triggers definition to verify the termination. We think that this could be a limitation especially for users without experience in triggers implementation. Thus, a part of our future work will be apart from detecting the non termination problem trying to provide some alternatives for the solution. (d) The user needs to define the OCL constraints, which can not be directly specified in the graphical model, manually into the corresponding class by using the Oclarity editor. This task requires experienced users in OCL although the Oclarity editor could perform syntactic verification. Therefore, we think as future work incorporating a new module to make easier the transformation of the CIM (Computation Independent Model) of the constraints specification to PIM.

Our approach makes it easier for the database developer to generate maintaining mechanisms directly from the generation of the schema in question. Moreover, when the integrity constraints of this schema are modified, the corresponding triggers are also automatically modified. Using this approach, the developers will obtain both the best system performance because active mechanisms are implemented as part of the database schema rather than in the application, as well as the best data independence because the integrity constraints are also embedded in the database schema rather than in external applications.

Furthermore, we will design experiments to validate our tool. These experiments focus on showing the usefulness of using it to facilitate maintenance and design tasks. Therefore, we propose two kinds of experiments: the first concerns the usefulness of checking semantics with triggers. The second is concerned with the user interface showing triggers and sequence diagrams. We want to know whether the designer understands the proposed diagrams and detects what each one does.

References

Al-Jumaily, H.T., 2006. Active technology application to control constraints in database development. Ph.D. Thesis, Carlos III University of Madrid, Spain.

ArgoUML, 2007. <http://argouml.tigris.org/>.

Baralis, E., Widom, J., 2000. An algebraic approach to static analysis of active database rules. ACM Transactions on Database Systems 25 (3), 269–332.

Baralis, E., Ceri, S., Widom, J., 1993. Better termination analysis for active databases. In: Proceedings of the First International Workshop on Rules in Database Systems, Edinburgh, Scotland, pp. 163–179.

Birgit, D., Heinrich, H., 1999. Using OCL constraints for relational database design – The unified modeling language. In: Proceedings of the Second International Conference, LNCS 1723. Springer, pp. 598–613.

Budgen, D., Thomson, M., 2003. CASE tool evaluation: experiences from an empirical study. Journal of systems and software 67 (2), 55–75.

Cabot, J., Teniente, E., 2006. Constraint Support in MDA Tools: A Survey, Model Driven Architecture – Foundations and Applications, LNCS, pp. 256–267.

Ceri, S., Fraternalli, P., 1997. Designing Database Applications With Objects and Rules: The IDEA Methodology. Addison-Wesley.

Ceri, S., Widom, J., 1990. Deriving production rules for constraint maintenance. In: Proceedings of VLDB Conference IBM Almaden Research Center.

Ceri, S., Fraternali, P., Paraboschi, S., Tanca, L., 1994. Automatic generation of production rules for integrity maintenance. ACM Transaction on Database Systems 19 (3).

Chen, P., 1976. The entity-relationship model – toward a unified view of data. ACM Transactions on Database Systems 1 (1).

Cuadra, D., Nieto, C., Castro, E., Martinez, P., Velasco, M., 2003. Preserving relationship cardinality constraints in relational schemata. Database Integrity: Challenges and Solutions. Ed: Idea Group Publishing.

Decker, H., Martinenghi, D., Christiansen, H., 2006. Integrity checking and maintenance in relational and deductive databases and beyond. In: Intelligent Databases: Technologies and Applications, Idea Group, pp. 238–285.

Domínguez, E., Lloret, J., Zapata, M.A., 2002. Integrity constraint enforcement by means of trigger templates. In: Second International Conference, Advances in Information Systems, ADVIIS, 2002.

Elmasri, R., Navathe, S., 2000. Fundamentals of Database Systems, Third ed. Addison-Wesley.

EmPowerTec, 2006. <http://www.empowertec.de/products/rational-rose-ocl.htm/>.

Gogolla, M., Richters, M., 1998. Transformation rules for UML class diagrams. The Unified Modeling Language, UML'98 – Beyond the Notation. In: First International Workshop, Mulhouse, France.

Hickey, T., 2000. Constraint-based termination analysis for cyclic active database rules. In: Proceedings of the Sixth International Conference on Rules and Objects in Databases, LNAI, vol. 1861. Springer, pp. 1121–1136.

IBM Rational, 2007. XDE Developer. <http://www-306.ibm.com/software/awdtools/developer/rosexde/>.

ISO/IEC 9075 Standard, 2003. Information Technology – Database Languages – SQL:2003 International Organization for Standardization.

Kulkarni, K., Mattos, N., Cochrane, R., 1998. Active Database Features in SQL3, Active Rules in Database Systems. Springer-Verlag, New York. pp. 197–218.

MetaEdit+, 2007. <http://www.metacase.com/>.

Objecteering/UML, 2007. Objecteering/SQL Designer User Guide Version 5.2.2. <http://depinfo.u-bourgogne.fr/docs/Objecteering522/SQLDesigner.pdf>.

OCL22SQL, 2007. Dresden OCL Toolkit. <http://dresden-ocl.sourceforge.net/>.

Olivé, A., 2003. Integrity constraints definition in object-oriented conceptual modeling languages. In: Conceptual Modeling – ER 2003, 22nd International Conference on Conceptual Modeling, Chicago, IL, USA.

OMG, 2007. Object Management Group, Inc. <http://www.omg.org/mda/>.

Oracle, 2007. Oracle® Database SQL Developer User's Guide <http://download.oracle.com/docs/cd/B19306_01/appdev.102/b31695/dialogs.htm#BACIGFBJ>.

Paton, N., Díaz, O., 1999. Active Database Systems. ACM Computing Surveys 31 (1).

Rational Enterprise Edition, 2003. <www-306.ibm.com/software/rational/>.

Rational Software, 2000. Mapping Object to Data Models with the UML Mapping Objects to Relational Databases. <http://www.uml.org.cn/oobject/tp185.pdf>.

Teorey, T.J., 1999. Database Modeling and Design, third ed. Morgan Kaufmann Series in data management systems.

Türker, C., Gertz, M., 2000. Semantic Integrity Support in SQL-99 and Commercial (Object) Relational Database Management Systems. UC Davis Computer Science Technical Report CSE-2000-11, University of California.

Verheecke, B., Straeten, R., 2003. Specifying and implementing the operational use of constraints in object-oriented applications. In: Proceedings of TOOLS PACIFIC 2002, vol. 10, p. 23.

Visual Case Tool, 2007. <http://visualcase.com/index.htm/>.

Wahler, M., Koehler, J., Brucker, A., 2006. Model-Driven Constraint Engineering, Workshop on OCL for (Meta-)Models in Multiple Application Domains (OCLApps), Models 2006.

Harith T. Al-Jumaily. Since 1999, he has worked at the Advanced Databases Group in the Computer Science Department at the Universidad Carlos III of Madrid. In 2006, he obtained a Ph.D. in Information Science from the Universidad Carlos III of Madrid. He is currently teaching File Structure and Database Design. His research interests include Advanced Database Technologies, Information Retrieval and Software Engineering.

Dolores Cuadra received the M.Sc. in Mathematics from the Universidad Complutense of Madrid in 1995. Since 1997, she has worked as assistant lecturer at the Advanced Database Group in the Computer Science Department of the Carlos III University of Madrid. In 2003 she obtained a Ph.D. in Computer Science from the Carlos III University of Madrid. She is currently teaching File Organization, Database Design and Data Modelling. Her research interests include data models, conceptual and logical modelling and Advanced Database CASE environments. She has been working in the Computer Science Department at Purdue University of West Lafayette (Indiana) for nearly a year, where she has applied her research in Spatio-Temporal databases.

Paloma Martínez Fernández obtained a degree in Computer Science from the Universidad Politécnica of Madrid in 1992. Since 1992, she has been working at the Advanced Databases Group in the Computer Science Department at Universidad Carlos III of Madrid. In 1998 she obtained a Ph.D. in Computer Science from the Universidad Politécnica of Madrid. She is currently teaching Database Design and Advanced Databases in the Computer Science Department at the Universidad Carlos III de Madrid. She is working in several European and National research projects on Natural Language Processing, Information Retrieval, Advanced Database Technologies and Software Engineering.