OBJECT NORMALIZATION AS THE CONTRIBUTION TO THE AREA OF FORMAL METHODS OF OBJECT-ORIENTED DATABASE DESIGN

Jan Vraný
Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague
Technická 2, Prague 6, 160 00

Zdeněk Struska, Vojtěch Merunka
Department of Information Engineering, Faculty of Economics and Management, Czech University of Agriculture in Prague
Kamýcká 129, Prague 6 - Suchdol, 165 21

Keywords: Data normalization, object data model (ODM), relational data model (RDM), first object normal form (1ONF), second object normal form (2ONF), third object normal form (3ONF).

Abstract: In the article there is described an overview of current status in the area of formal technique of object database design. It is discussed there, why relational design techniques as normalization, decomposition and synthesis are not able to be easy used in object databases. The article informs with various proposals of object normal forms and it brings own authors evaluation and an example of object normalization.

1 INTRODUCTION

Nowadays many various object databases are used practically. (Authors of this paper have the most of experience with Gemstone system). The programmers and experts have many implicit knowledge and experience concerning an effective application development. Unfortunately the knowledge is expressed very few in a form of formal rules and procedures, which would be accepted in the programming community. Therefore in the practice, we can see wrong usage of relations and hierarchies among objects, breakneck tricks in code, etc. The problem of these applications is not that they do not work. Unfortunately really monstrous constructions work thanks to modern components and development environments. The discussion with designer is all the more worse that he has to rebuild his running system, because he is not disposed to listen the arguments about difficult maintenance, extensibility, reliability, risk of data inconsistency and redundancy.

This is why we decided to discuss the formal techniques of object databases design. Database is the fundamental of practically all software applications and object databases are technology, which will gradually attain the importance. Many myths exist in the community of software producers. For example very popular is the myth that we need not complete any normalization, because object databases allow operations with objects in any form in contrast to the relational.

2 OBJECT NORMALIZATION

Some various papers arised in the turn of 1980s and 1990s (for example (Wai, Yiu-Kai, Embley, 1992)). First papers applied to the enlargement of relational techniques, but we can meet the original papers in last years.

2.1 Nootenboom's OONF

First three normal forms for relational and object databases are universally valid according to Dutch author Henk Nootenboom (Nootenboom, 2002). As a substitute for the fourth and fifth relational normal form (and probably BCNF) he sets up the concept of only one object normal form, which has following definitions:

> *A collection of objects is in OONF if it is in 3NF and contains meaningful data elements only.*

Universal validity of 1NF, 2NF and 3NF is absolutely right idea that we can agree with, but we leave without comment the OONF definition.

Vraný J., Struska Z. and Merunka V. (2006).
OBJECT NORMALIZATION AS THE CONTRIBUTION TO THE AREA OF FORMAL METHODS OF OBJECT-ORIENTED DATABASE DESIGN.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - ISAS*, pages 471-474
DOI: 10.5220/0002489304710474

2.2 Khodorkovsky's ONF, 4ONF, 5ONF and 6ONF

The paper (Khodorkovsky, 2003) sets up the idea of object normal form (ONF), which concerns the "right" relation among data of object and methods of object. Khodorkovsky's rule is added to classical "relational" definitions 4NF, 5NF (and to the 6NF, which is author's enhancement of the 5NF). The author calls these additions to classical definitions as 4ONF, 5ONF and 6ONF.

The paper is considered to more qualified formulation of similar ideas as the example above. It is valid according to the author that 1NF, 2NF and 3NF are common for relational and object databases.

2.3 "Chinese" ONF

The paper (Yonghui, Zhou, 2001) set up one object normal form as substitute for all relational normal forms. It considers the object data model to be similar to the tree-like data hierarchy in the way of XML. We suppose that this approach does not concern with object databases design, as we understand it.

2.4 "Australian-Swiss" ONF

The authors (Tari, Stokes, Spaccapietra, 1997) set up one ONF by the help of more types of functional dependences among objects. Concrete "path dependency" concerns composition of objects and navigability among objects, "local dependency" concerns relations of internal object components and "global dependency" concerns requirements on application. Objects structure is in ONF then, if users' requirements on applications are retrospectively deducible from the relations among objects. In agreement with our opinion it concerns very interesting contribution to problem of testing the accordance of suggested object oriented application model with its requirements. We think, that this problem is related to the other issues of databases development, but it not solves the problem we are concetrated on.

2.5 Three Ambler–Beck's Object Normal Forms

Three object Ambler–Beck's normal forms for object oriented applications are set up in internet (Ambler, 2004) and in the book (Beck, 2003). These normal forms are analogical with first, second and third relational normal form. The authors define these normal forms 1-ONF, 2-ONF and 3-ONF themselves and talk about them as a tool for objects classes' normalization complementary with technique of design patterns.

3 OUR INTERPRETATION OF THE PROBLEM

All works mentioned here are very important contribution into discussed area. When we resume, what probably the community of analysts and designers expect from the technique of object normalization, we can define the following conclusion:

1. It has to be very simple, clear, and understandable and it should work with minimum of concepts similarly as it is with "classical" normalization. We suppose that implementation of difficult definitions distinctively exceeding the range of classical normal forms, a lot of types of concepts and relations, is not the right way.
2. It should be focused only on database design, e.g. on the structure of objects, which will serve as data storage and manipulation in database systems. It does not need to work with objects, which are responsible for "operation" of applications. There are design patterns for them and there is no need to substitute this technique.
3. It is possible, that in the future the object approach will become universal approach to information system analysis, and relational technology will limit itself to be only one of possible implementation variant. So present conditions can be turned to the opposite. It would be smart to have the new theory analogical with entity-relation modelling concept and relational normalization. In the best case, the relational normalization (as a tool of the relational technology) should be deducible from new theory as its special kind.

We have to define first, what we understand by database object. Because it "only" serves for data saving and manipulation. It is not object, which ensures some behavioural aspects of the software application. This is why we propose to not work with data and with methods separately and define one concept of "attribute". We will not distinct, if the particular attribute is implemented into object by its data or if it is the result value of some method.

There is a question, if this simplification is not large. For example Ambler-Beck's approach works directly with data and methods and uses them in its definitions. But we think that we can afford this simplification for the data objects.

We think that the modified form of Ambler-Beck's approach fulfils best all above described requests. This version was already used in our lessons and practically in Czech companies, and we shall work on it further. We suppose that this method should precede during modelling all possible next thoughts about using of inheritance, composition and other relations among objects.

1-ONF

A class is in first object normal form (1-ONF) when its objects do not contain group of repetitive attributes. Repetitive attributes must be extracted into objects of the new class. The group of repetitive attributes is then replaced by the link at the collection of new objects. An object schema is in 1-ONF when all of its classes are in 1-ONF.

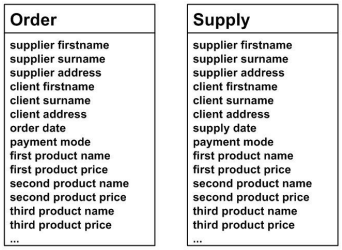

Figure 1: Example of non-normalized form.

In the figure 1 there is the example in non-normalized form and in the figure 2 there is the same example in 1-ONF. In contrast to the original approach, we do not assume, that designers recognize automatically groups of repetitive attributes and have them extracted out into independent classes. This rule is really necessary according to our experience from school and practice. The problem is not always trivial as in mentioned example. Repetitive attributes can exist under under various names and they are not always easy visible for the first sight.

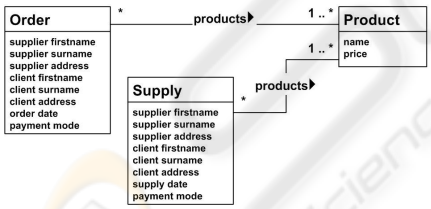

Figure 2: 1-ONF.

2-ONF

A class is in second object normal form (2-ONF) when it is in 1-ONF and when its objects do not contain attribute or group of attributes, which are shared with another object. Shared attributes must be extracted into the new objects of new class, and in all objects, where they appeared, must be replaced by the link at the shared object of new class. An object schema is in 2-ONF when all of its classes are in 2-ONF.

It concerns the attributes - supplier's first name, supplier's surname and his address and client's first name, client's surname, his address and method of payment in our example. These attributes were shared for concrete order and concrete supply. Hence it was necessary to establish new object class `Contract`.

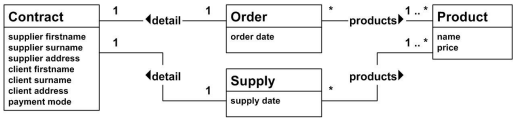

Figure 3: 2-ONF.

3-ONF

A class is in third object normal form (3-ONF) when it is in 2-ONF and when its objects do not contain attribute or group of attributes, which have the independent interpretation in the modelled system. These independent attributes must be extracted into objects of new class and in objects, where they appeared must be replaced by the link at this new object. An object schema is in 3-ONF when all of its classes are in 3-ONF.

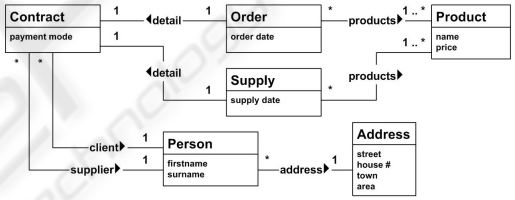

Figure 4: 3-ONF.

It concerns the data about suppliers and client in the objects of class Contract. These attributes have independent interpretation without contracts, because they represent particular persons. And we can declare the same about the addresses.

Our approach mentioned here brings many questions and themes to next thought. One of them is a question, how the inheritance and other used relations among objects can be included into this approach. The consideration offers, that right use of inheritance would be interested next fourth object normal form.

When we focus on the process of object normalization, we can see its practical analogy with technique of applying design patterns and primarily with refactoring technique. Our future research will focus on this direction, where we will try to define the rules of object normal forms as a formulation of refactoring requirements of object database scheme.

Moreover, usage of object database calculus would be the most suitable for exact description of required approach. Unfortunately we are still in start of this area, because one and the only accepted standard is ODMG-3.0 (Catell, 2000). This model is burdened by

an attempt of convergence with SQL-like object relational databases and extensively it is oriented on implementation. The authentic formal tool for description of object database scheme is still missing.

4 CONCLUSION

It is a pity, that so perspective and practical used database technology has not yet comprehensible and universally accepted theoretical foundation and formal techniques of design. It is unquestionable, that many research centres are interested in this theme, but coherent and universally accepted results were not yet published. Absence of common formal tool and technique makes the large incompatibility of present object database systems, even if they are very good practically used product themselves. Therefore we suppose that near future brings maybe a few alternative approaches.

Authors would like to acknowledge the support of the research project MSM6046070904 on knowledge management information system design of the Czech Ministry of Education.

REFERENCES

Ambler, S. (1997) *Building Object Applications That Work, Your Step-By-Step Handbook for Developing Robust Systems Using Object Technology.* Cambridge University Press/SIGS Books, 1997, ISBN 0521-64826-2

Ambler, S. (2004) *Object Orientation – Bringing data professionals and application developers together.* http://www.agiledata.org/essays/ objectOrientation101.html.

Barry D. (1996) *The Object Database Handbook: How to Select, Implement, and Use Object-Oriented Databases.* ISBN 0471147184

Beck K. (2003) *Agile Database Techniques – Effective Strategies for the Agile Software Developer.* John Wiley & Sons; ISBN 0471202835

Blaha M., Premerlani M. (1997) Object–Oriented Modeling and Design for Database Applications. Prentice Hall, ISBN 0-13-123829-9

Carda A., Merunka V., Polák J. (2003) Umění systémového návrhu – objektově orientovaná tvorba informačních systém pomocí pvodní metody BORM. Grada, Praha 2003. ISBN 80-247-0424-2.

Catell R. G. (2000) *The Object Database Standard: ODMG 3.0* ISBN 1558606475

Gemstone Systems *Gemstone Object Server – documentation & non-commercial version download.* http://www.gemstone.com, http://smalltalk.gemstone.com.

Hruška T. (1995) Objektově orientované databázové technologie. Proceedings of the conference Tvorba softwaru 1995, ISBN 80-901751-3-9

Khodorkovsky V. V. (2002) *On Normalization of Relations in Databases.* Programming and Computer Software 28 (1), 41-52, 2002, Nauka Interperiodica, ISBN: 5-02-002404-X.

Kroha P. (1995) *Objects and Databases.* McGraw Hill, London 1995, ISBN 0-07-707790-3.

Loomis M., Chaundri A. (1997) *Object Databases in Practice* ISBN 013899725X

Merunka V. (2003) *Object database system Gemstone* Proceedings of the conference OBJEKTY 2003. Ostrava 2003. ISBN 80-248-0274-0

Molhanec M.: *Some comments to the interpretation of the Object oriented paradigm* Proceedings of the conference OBJEKTY 2005. Praha 2005. ISBN 80-248-0672-X

Nootenboom H. J. (2002) *Nut's – a online column about software design.* http://www.sum-it.nl/en200239.html.

Tari Z., Stokes J., Spaccapietra S. (1997) *Object Normal Forms and Dependency Constraints for Object-Oriented Schemata.* ACM Transactions on Database Systems 513-569, Vol 22 Issue 4, December 1997.

Wai Y. M, Yiu-Kai N. Embley D. W. (1992) *An Improved Nested Normal Form for Use in Object–Oriented Software Systems.* Proceedings of the 2nd International Computer Science Conference: Data and Knowledge Engineering: Theory and Applications, pp. 446-452, Hong Kong, December 1992.

Yonghui W,, Zhou A. (2001) *Research on Normalization Design for Complex Object Schemes.* Info-Tech and Info-Net, vol 5. 101-106, Proceedings of ICII 2001, Beijing.