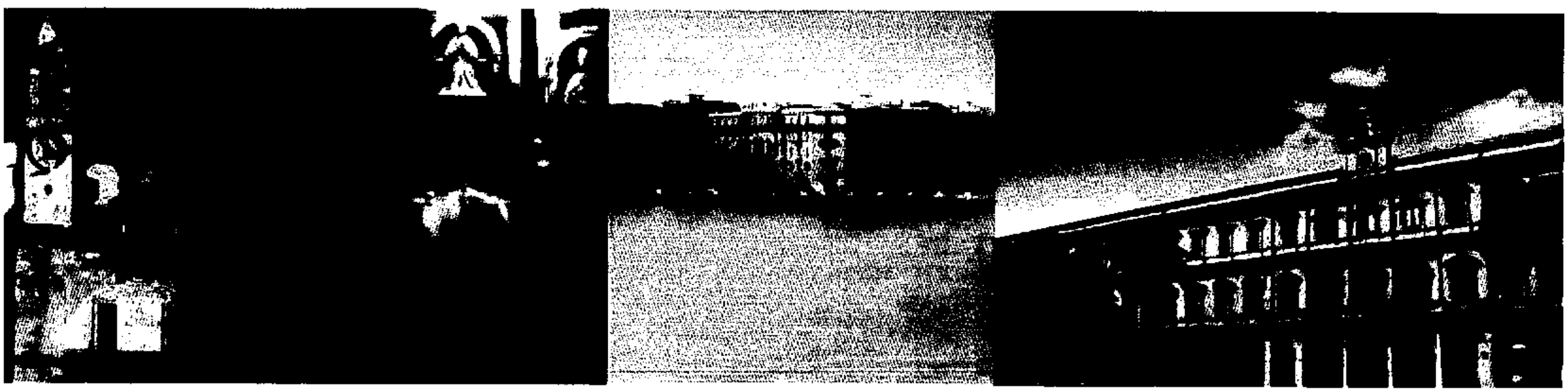
Proceedings of the 10th European Conference on

Software Maintenance and Reengineering

Bari, Italy
22-24 March 2006



2006

Sponsored by

IEEE computer society 60th anniversary

Research Centre On Software Technology

CONFCOOPERATIVE Puglia

dib Department of Informatics Bari

CeRICT

CERIT centro ricerche per l'innovazione nell'information technology

Reengineering Forum

IEEE

EDS

SAP

serlab SOFTWARE ENGINEERING RESEARCH

exprivia

UNIVERSITY OF BARI

Published by the IEEE Computer Society Press
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314

IEEE Computer Society Order Number P2536
ISSN 1534-5351
ISBN 0-7695-2536-9

ISBN 0-7695-2536-9

Proceedings

10th European

Conference on Software Maintenance and Reengineering

22-24 March 2006
Bari, Italy

Proceedings

10th European Conference on Software Maintenance and Reengineering

Sponsored by

IEEE

IEEE computer society

Esprivia

Reengineering Forum

Department of Informatics Bari

University of Bari

Research Centre On Software Technology

CeRICT

CRIAT

CONFCOOPERATIVE Puglia

EDS

SAP

serlab

IEEE COMPUTER SOCIETY

IEEE

Los Alamitos, California
Washington • Tokyo

IEEE Computer Society Order Number P2536
ISBN 0-7695-2536-9
ISBN-13 978-0-7695-2536-5
ISSN 1534-5351

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Individual paper REPRINTS may be ordered at: reprints@computer.org

Editorial production by Silvia Ceballos

Cover art production by Alex Torres

Printed in the United States of America by Odyssey Press Inc.

IEEE COMPUTER SOCIETY

IEEE

IEEE Computer Society
Conference Publishing Services
http://www.computer.org/proceedings/

Conference on Software Maintenance and Reengineering

Table of Contents

Technical Session 4: Tools and Frameworks

Technical Session 5: System Refactoring and Migration

Technical Session 6: Design Recovery

Technical Session 7: Experience Reports and Empirical Studies

Technical Session 8: Program Analysis

Technical Session 9: System Assessment

Short Papers Session

Keynotes

SPECIAL SESSIONS

Doctoral Symposium

Tool Demonstration

Industrial Track

MESSAGE FROM THE CHAIRS

Welcome to the 10th edition of the Conference on Software Maintenance and Reengineering (CSMR 2006), the leading European conference on the theory and practice of maintenance, reengineering and the evolution of software systems. In its first ten years, CSMR has grown more and more, becoming an important European forum for discussion and exchange of research and experience among researchers and practitioners both from academia and the industrial world in the field of Software Engineering. It attracts contributions from many countries all over the world.

The unceasing evolution of methods and technologies in software systems development speeds up the need to evolve existing systems that have to be continuously maintained to preserve their business value, allowing them to continue to be used effectively in new and modern modified environments. Moreover, new technologies themselves introduce new problems in the maintenance and evolution of the systems developed by them, requiring further new solutions.

The CSMR aims to be a reference point where new advances in the research of methods and techniques to efficiently and effectively maintain and evolve software systems are presented and discussed, as well as to promote cooperation among researchers to face problems and to provide new solutions to overcome them.

With this in the mind, the main theme of this year's conference is "Developing maintainable and maintaining Service Oriented Software Systems (SOSS)". The development of service-based systems (e.g. using web services) is increasing, thus we have to face new issues and problems to develop, maintain and evolve SOSS. The definition and adoption of methods and techniques to do this effectively, as well as to migrate or reengineer existing systems to SOSS is one of the grand challenges for the next few years.

Sixty-five technical papers, from countries across the world, were submitted to CSMR 2006. Each paper has been reviewed at least by three members of the Program Committee and twenty seven full papers and four short papers were accepted for presentation at the Conference and inclusion in the Proceedings. Moreover, the technical program includes three Special Sessions: a Doctoral Symposium, with 6 presentations, a Tool Demonstration, with 6 presentations, and an Industrial Track, with 3 presentations.

Two keynote speeches enrich the CSMR 2006 Program: the first, by Professor Mark Harman from King's College London, is on the application of the Search-Based Software Engineering to the field of systems maintenance and reengineering; the second, by Dr. Werner Sommer, from SAP Labs France, is on the management of product innovation. The presence of a keynote speaker from the academia and a keynote speaker from the industry, as well as the Industrial Track special session, would highlight and enforce the role of the CSMR as an important meeting point between the academia and the industrial world.

The program is completed by a half day pre-conference tutorial about the migration of legacy components to a service oriented architecture, and a half day post conference workshop on the maintenance and reengineering of web based applications.

We want thank all of those who have contributed to set up and hold this year's conference: the authors for submitting their papers, the Program Committee members for their valuable work in reviewing and selecting the papers and in promoting the conference, the chairs of the special sessions, the chairs of the organizing committee together with all the people that helped in arranging the conference, and the members of the Steering Committee for their encouragement and advice. We thank all the organizations that are financially sponsoring this event and the University of Bari for the local arrangements. We also thank the IEEE Computer Society for producing the final proceedings.

Finally, we want thank all the attendees of CSMR 2006 who will make it a successful event, and we hope you will find the program interesting and enjoy your stay in Bari, a town rich in history and tradition.

Giuseppe Visaggio, General Chair
Giuseppe Antonio Di Lucca, Program Chair
Nicolas Gold, Program Co-Chair

COMMITTEES

Conference Organizing Committee

General Chair: *Giuseppe Visaggio*
University of Bari
Dipartimento di Informatica, Bari, Italy

Program Chair: *Giuseppe Antonio Di Lucca*
University of Sannio
RCOST -Research Center on Software Technology, Benevento, Italy

Program Co-Chair: *Nicolas Gold*
Department of Computer Science
King's College London UK

Tool Demonstration Chair: *Massimiliano Di Penta*
University of Sannio
RCOST -Research Center on Software Technology, Benevento, Italy

Doctoral Symposium Chair: *Rainer Koschke*
University of Bremen
Bremen, Germany

Industrial Track Chair: *Sergio Betti*
Centro Regionale di Competenza sulle ICT CRdC-ICT, Benevento, Italy

Publicity Chair: *Alessandro Bianchi*
University of Bari
Dipartimento di Informatica, Bari, Italy

Local Organization and Treasure Chair: *Danilo Caivano*
University of Bari
Dipartimento di Informatica, Bari, Italy

Web & submission Chair: *Pasquale Ardimento*
University of Bari
Dipartimento di Informatica, Bari, Italy

Program Committee

Nicolas Anquetil, Universidade Católica de Brasília
Giulio Antoniol, University of Sannio
Haroon Babri, Lahore University of Management and Technology
Teresa Baldassarre, University of Bari
Francoise Balmas, Universite Paris 8
Ira Baxter, Semantic Designs, Inc.
Keith Bennett, University of Durham
Fernando Brito e Abreu, New University of Lisbon
Liz Burd, University of Durham
Gerardo Canfora, University of Sannio
Elliot Chikofsky, Engineering Management & Integration
Aniello Cimitile, University of Sannio
Andrea De Lucia, University of Salerno
Dirk Draheim, Freie Universität Berlin
Stephane Ducasse, Université de Savoie
Jürgen Ebert, University of Koblenz-Landau
Anna Rita Fasolino, University of Naples Federico II
Juan Fernandez-Ramil, The Open University
Harald Gall, University of Zürich
Tibor Gyimothy, University of Szeged
Mark Harman, King's College London
Mira Kajko-Mattsson, Stockholm University and Royal Institute of Technology
Kostas Kontogiannis, University of Waterloo
Michele Lanza, University of Lugano
Spiros Mancoridis, Drexel University
Andrian Marcus, Wayne State University
Hausi Muller, University of Victoria
Lori Pollock, University of Delaware
Vaclav Rajlich, Wayne State University
Claudio Riva, Nokia Research Center
Harry Sneed, Software Maintenance & Reengineering Consultant
Eleni Stroulia, University of Alberta
Tarja Systä, Tampere University of Technology
Paolo Tonella, ITC - IRST, Trento
Maria Tortorella, University of Sannio
Porfirio Tramontana, University of Naples Federico II
Arie van Deursen, Delft University of Technology
Chris Verhoef, Free University of Amsterdam
Hongji Yang, De Montfort University
Andy Zaidman, University of Antwerp

Additional Reviewers

Ilham Alloui
Lerina Aversano
Mario Luca Bernardi
Thierry Bodhuin
Andrea Capiluppi
Luigi Cerulo
Marco D'Ambros
Vincenzo Deufemia
Massimiliano Di Penta
Damiano Distante
Fausto Fasano
Tudor Girba
Bas Graaf
Carmine Gravino
Orla Greevy
Steven Klusener
Adrian Kuhn
Patricia Lago
Mircea Lungu
Onaiza Maqbool
Ali Mesbah
Atmaram Mulliah
Rocco Oliveto
Martin Pinzger
Denys Poshyvanyk
Derek Rayside
Romain Robbes
Giuseppe Scanniello
Silvio Stefanucci
Christoph Stoermer
Marco Torchiano
Tom Tourwe
Niels Veerman
Herve Verjus
Corrado Aaron Visaggio
Xinrong Xie
Zhenchang Xing

Obtaining Web Services from Relational Databases

García-Rodríguez de Guzmán, I. [1], Polo, M. 2, Piattini, M. [2]
[1] *University Rey Juan Carlos*
Kybele Research Group
Ignacio.Garcia@urjc.es
[2] *ALARCOS Research Group*
Information Systems and Technologies Department
UCLM-Soluziona Research and Development Institute
University of Castilla-La Mancha
Paseo de la Universidad, 4 – 13071 Ciudad Real, Spain
{Macario.Polo,Mario.Piattini}@uclm.es

Abstract

Database reverse engineering in general has only been used to recover the conceptual schema employed during its construction. To the best of our knowledge, none of this work has made use of that static structure to perform a deep analysis to infer functionality, or in other words, services over data that could be offered via Web Services.

Our work goes beyond the discovery of functionality, generating Web Service-based applications to allow the remote access and management of legacy databases.

1. Introduction

When we speak about reengineering, we imagine a legacy application from which (and by means of a reverse engineering technique) we extract information about its most relevant entities, with the possible goal of generating an improved version. Reengineering has many purposes, namely improving the quality of an application, obtaining a new version of the software system, adding new features, etc.[1].

Not only old applications but also relational databases should be considered as legacy systems, because relational databases are subjects of reengineering processes.

Having a multi-tier application [2], we have to realize that the domain tier is chiefly responsible for implementing the required operations to achieve the objective for which the system was developed; moreover, this tier is usually considered as the conceptual model for building the relational database. Because of that, the database can be used not only to extract the static structure in a reengineering process but also to infer many of the original system functionalities.

Our research area is *reengineering relational databases to Web Service-based applications*, and related to this, we are developing a reengineering tool, in which we are implementing all our research results [3].

If we try to develop a general and partially-automated reengineering process for relational databases, we generally need to specify which kind of relational databases are involved. As far as we know, despite SQL-99 (object-relational databases) being the current standard, most legacy databases are still supported by SQL-92. For this reason, we are now mainly focusing on SQL-92- based databases, to exploit all its features, in order to develop a set of patterns which can help us find potential services in the schema of the database, using a model-driven pattern matching process as a sub-step of our general reengineering process.

The paper is organized as follows: Section 2 provides a brief description of the related work; Section 3 overviews Relational Web, our reengineering environment; Section 4 summarizes a part of our work; Section 5 depicts the role of Web Services in our context; Section 6 gives some conclusions and future work.

2. Related Work

Until recently, data reengineering (and more specifically reverse engineering) had not been one of the most important topics in reengineering for two straightforward reasons: (1) the traditional partition of software engineering and database systems, and (2) source code reverse engineering seems more interesting in many aspects in the academic environment [4]. Even though we can find significant research work in the context of database reengineering, almost all of it focuses on recovering the conceptual schema used during initial development to migrate the database from an old data model to a new one, to a new DBMS in the same model, etc.

As well as tools, methodologies have also been designed. [5, 6] present a reengineering database project named DB-MAIN. DB-MAIN is a generic methodology with the following steps: (1) database structure extraction and (2) data structure conceptualization. This methodology is implemented in a tool also called DB-MAIN.

In [7], the MIDAS framework is described. MIDAS tackles the task of migrating net databases into relational databases. MIDAS implements can also replace database access subroutines for SQL statements by means of a set of graph-based transformations.

Model management, one of the most important drivers in current reengineering theory, is used, of course, to join traditional techniques with current advantages in reengineering. In this respect, we can mention [8, 9], where the MOMENT framework for model management is explained. Model management is a new discipline in model engineering that considers models (and metamodels) as first order citizens and also provides a set of generic operators in order to manipulate them. In MOMENT, models are specified as sets of elements different from the metamodel, so the algebraically specified operators by means of Maude (see below) can access these elements with no knowledge of the model representation [10]. MOMENT also deals with database reengineering. MOMENT applies model transformation only to recover the structure and to generate applications, taking relational databases as the starting point.

3. Relational Web: a brief description

Relational Web is a tool which implements a whole reengineering process for relational databases. The different database vendors supported are Microsoft Acces, SQL Server, Caché Intersystems and Oracle.

Figure 1 summarizes the process, setting aside many technical and algorithmic details (see [3, 11]). As can be seen in Figure 1, the starting point is a relational database from which we obtain the logical schema corresponding to the physical database, which is represented using a *"database metamodel"*. To take advantage of the object-oriented paradigm, the instance of the current database metamodel is transformed into an instance of the ***"Object Oriented System"*** metamodel, which is itself nearly a representation of the end-application. Having this instance of the OOS metamodel, we use different code-factories depending on the sort of application we want/need to generate.

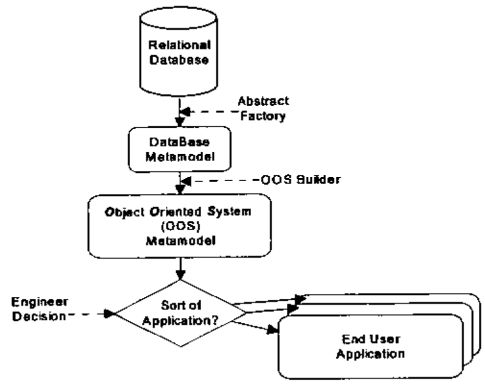


Figure 1. Relational Web abstract process representation

As previously mentioned, it is difficult to infer behaviour from a relational database, even more so when we are dealing with SQL-92 databases. But *Relational Web* implements two mechanisms to give behaviour to final applications: implementing the well known *CRUD operations* and building state machines.

By means of a state machine, we can establish a set of different states where an instance representing a record of a table could be found in any moment of its life-cycle. For example, imagine an object representing an instance of the table *Account*, representing a banking account. Its possible set of states could be: *BalanceZero*, *BalancePositive* and *BalanceNegative*.

In the current version, the decision about making one set of states or another falls to the engineer, that is to say, the state machine generation process is not an automatic process, but a manual one. For further information, please see [3].

4. Eliciting information from SQL-92 databases

As far as we know, there are no studies that deal with the task of inferring functionality from the static representation of a relational database. We are now working in this area, using the MDA approach [12] to solve it.

The current standard in relational (object-relational in fact) databases is SQL-99, but the reason we are starting our research work in the SQL-92 standard is because SQL-92 is one of the most common set-up databases (as for example can be seen in [13], where most reversed databases were relational), and thus, the most suitable candidate for a reengineering process (to expose the database to the web by means of Web Services, in our context).

The full structure of SQL-92 databases is represented in the metamodel presented in [14]. This metamodel has been developed for SQL-2003, but since SQL-92 is a subset of SQL-99, and the latter is also a subset of SQL-2003, we can customize the SQL-2003 metamodel by means of a projection (which can be seen as a model transformation in the MDA approach).

Taking these assumptions, we can extend and improve all the results obtained from the research done on SQL-92 to the following versions of the SQL standard.

4.1. Model-Driven Pattern Matching

As noted above, the database schema is considered as the reflection of the domain layer, so it is very possible that many of the functionalities of the application are reflected in the schema of the relational database.

Although it is a difficult task, we are trying to discover patterns to identify functionality in the recovered schema of the relational database.

Considering the relational database schema and the patterns as models, we can make pattern matching using a transformational language, such as QVT [15], which is the OMG proposal to make queries, create views and perform transformations over models.

4.2. An example of our database patterns

In this section, we are going to briefly present another kind of pattern we have discovered (because of the available space, this is only a very short summary and serves as an example of our work):

- Pattern 1: If more than one foreign key exists between two tables, we generate for the secondary table (the table which is the origin of the foreign key) a method into which an SQL sentence is built automatically. This SQL sentence joins, using "ORs", different combinations of the pair "foreign key columns-referenced columns". This SQL acts as a more flexible filter which could give us more information.
- Pattern 2: In addition to the inherent security in SQL-92 databases (which acts at table level), we can generate source code in order to allow manipulation of a database record only by the user who added this record. For this pattern we suppose that a "users table" exist and the constrained table is related to the users table by means of a foreign key. Having that, we know the user who (perhaps) inserted a given record, and thus, we can generate code to allow constrained access.

4.3. Automatic generation of state machines

As noted in section 3, Relational Web has the possibility of adding state machines to customize the behaviour of the classes representing the tables in the relational database.

We think that a deep analysis of the structure of the database (SQL-92 at this time) could reveal enough information to say whether a class representing a record of a table is in one state or another.

5. Exposing functionality with Web Services

Up to now, we have spoken about reengineering SQL-92 databases to infer functionality, but we have not mentioned how we are going to offer this functionality.

In the end, all the functionality extracted can be viewed as operations over the database and our intention is to use Web Services to expose these operations (**Figure 2**). The current version of *Relational Web* makes it possible to generate EJB-based applications (EJB artefacts [16], the distributed components of Java); in the same way, Web Services (which are more flexible and heterogeneous) could perform the same goal. The reason for using Web Services instead of other types of component or software artefacts, is that Web Services have been created for system integration and to wrap legacy systems among other things [17].

The generated Web Services will expose the following operations: *CRUD operations*, Business operations defined in the state machines and Operations due to patterns.

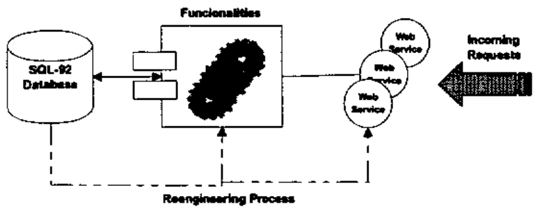


Figure 2. Web Service role inside the process

6. Conclusions and future work

Up to now, we have developed a complex but full functional environment for database reengineering, Relational Web, whose efficiency has been proven in many projects.

Despite the fact that SQL-92 databases are obsolete, they are the real legacy systems, and most companies are working over this kind of database. For this reason, we are designing an effective process for reengineering this kind of database towards the web, exposing functionalities and operations.

Our research work is mainly focused on discovering hidden functionalities by means of pattern matching and state machines. These functionalities are exposed by means of Web Services, which are also automatically generated inside the reengineering process.

7. Acknowledgements

This work is partially supported by the MÁS project (Mantenimiento Ágil del Software), Ministerio de Ciencia y Tecnología/FEDER, TIC2003-02737-C02-02, and the ENIGMAS project, Plan Regional de Investigación Científica, Desarrollo Tecnológico e Innovación, PBI-05-058.

8. References

[1]. Arnold, R.S., *Software Reengineering*, ed. 0-8186-3272-0. 1992: IEEE Press. pp. 675.

[2]. Larman, C., *Applying UML and Patterns*. 1998, New York: Prentice Hall, Upper Saddle River.

[3]. García-Rodríguez de Guzmán, I., M. Polo, and M. Piattini. *An Integrated Environment for Reengineering*. in *International Conference on Software Maintenance*. 2005. Hungary, Budapest: IEEE Computer Society.pp. 165-174

[4]. Müller, H.A., J.H. Jahnke, D.B. Smith, M.-A. Storey, S.R. Tilley, and K. Wong. *Reverse Engineering: A Roadmap*. in *International Conference on Software Engineering - Proceedings of the Conference on The Future of Software Engineering*. 2000. Limerick, Ireland.June 04 -11.pp. 47-60

[5]. Henrard, J., V. Englebert, J.M. Hick, D. Roland, and J.L. Hainaut. *Program understanding in database reverse engineering*. 2002.pp. 11

[6]. Hick, J.-M. and J.-L. Hainaut, *Strategy for Database Application Evolution: The DB-MAIN Approach*. LNCS 2813, 2003: p. pp. 291-306.

[7]. Cohen, Y. and Y.A. Feldman, *Automatic High-Quality Reengineering of Database Programs by Abstraction, Transformation and Reimplementation*. ACM Transactions on Software Engineering and Methodology, 2003. **12**((3)): p. pp. 285-316.

[8]. Boronat, A., J.Á. Carsi, and I. Ramos. *An Algebraic Baseline for Automatic Transformations in MDA*. in *Software Evolution through Transformations: Model-based vs. Implementation-level Solutions (SETra 2004)*. 2004. Rome, Italy.30-44 pp.

[9]. Boronat, A., J.Á. Carsí, and I. Ramos. *Automatic Reengineering in MDA Using Rewriting Logic as Transformation Engine*. in *Proceedings of the Ninth European Conference on Software Maintenance and Reengineering (CSMR'05)*. 2005. Manchester, United Kingdom: IEEE Computer Society.21–23 March.pp. 228-231

[10]. Boronat, A., J. Iborra, J.Á. Carsí, I. Ramos, and A. Gómez. *Del método formal a la aplicación industrial en Gestión de Modelos: Maude aplicado a Eclipse Modeling Framework*. in *Técnicas formales para el diseño del software*. 2005. Madrid, Spain

[11]. García-Rodríguez de Guzmán, I., M. Polo, and M. Piattini. *Metamodels and architecture of an automatic code generator*. in *Proceedings of 2nd Nordic Workshop on the Unified Modeling Language (NWUML'2004)*. 2004. Turku (Finland): TUCS General Publication.19-20 de Agosto.K. Koskimies, et al.pp. 115-126

[12]. OMG, *MDA Guide Version 1.0.1*. 2003, Object Management Group. p. 62.

[13]. Blaha, M. *A Retrospective on Industrial Database Reverse Engineering Projects-Part 1*. in *Proceedings of the 8th Working Conference on Reverse Engineering (WCRE'01)*. 2001. Suttgart, Germany: IEEE Computer Society.October 02 - 05, 2001.pp. 136-147

[14]. Calero, C., F. Ruiz, A. Baroni, F. Brito e Abreu, and M. Piattini, *An Ontological Approach To Describe the SQL:2003 Object-Relational Features*. Accepted in "Computer Standards and Interfaces", 2005: p. 28.

[15]. QVTP, *Revised submission for MOF 2.0 Query / Views /Transformations RFP (Version 1.1)*. 2003, QVT-Partners (http://qvtp.org/).

[16]. EJB, *Enterprise JavaBeans Technology*. 2005, Sun MicroSystems.

[17]. Alonso, G., F. Casati, H. Kuno, and V. Machiraju, *Web Services. Concepts, Architectures and Applications*, ed. M.J. Carey and S. Ceri. 2004, Berlin: Springer. pp. 354.