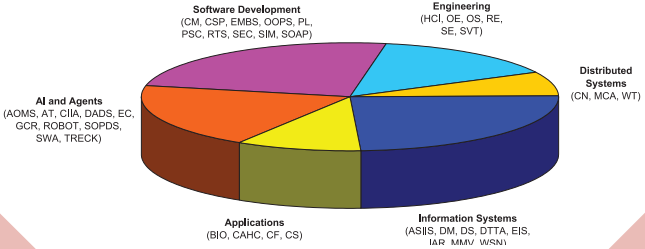
2009
Symposium on Applied Computing



Association for Computing Machinery

Advancing Computing as a Science & Profession

Main Menu

Hosted by
University of Hawaii at Mānoa and
Chaminade University of Honolulu

Honolulu, Hawaii, USA
March 8-12, 2009

The Association for Computing Machinery, Inc.
1515 Broadway
New York, New York 10036

ACM ISBN: 978-1-60558-166-8 **ACM Order Number: 104092**

Additional copies may be ordered prepaid from:

ACM Order Department
P.O. BOX 11405
Church Street Station
New York, NY 10286-1405
E-mail: acmhelp@acm.org

U.S.A. and Canada
Phone: 1-800-342-6626
All other countries
Phone: +1-212-626-0500
Fax: +1-212-944-1318

Printed in the U.S.A.

The 24th Annual ACM Symposium on Applied Computing

Honolulu, Hawaii, USA
March 8-12, 2009

SYMPOSIUM CHAIRS

Sung Y. Shin, South Dakota State University, United States
Sascha Ossowski, University Rey Juan Carlos, Spain

SYMPOSIUM VICE-CHAIR

Paulo Martins, Chaminade University, United States

PROGRAM CHAIRS

Ronaldo Menezes, Florida Institute of Technology, United States
Mirko Viroli, Università di Bologna, Italy

POSTERS CHAIR

Jiman Hong, Soongsil University, Korea

PUBLICATION CHAIR
(Proceedings Editor)

Dongwan Shin, New Mexico Institute of Mining and Technology, United States

eCONFERENCE MANAGEMENT CHAIR

Mathew J. Palakal, Indiana University Purdue University, United States

PUBLICITY CHAIR

Udo Fritzke, PUC-Minas, Brazil

LOCAL CHAIR

Martha Crosby, University of Hawaii at Mānoa, United States

TREASURER, WEBMASTER, & REGISTRAR

Hisham M. Haddad, Kennesaw State University, United States

About the Sponsoring SIG

ACM SIGAPP

The ACM Special Interest Group on Applied Computing is ACM's primary applications-oriented SIG. Its mission is to further the interests of the computing professionals engaged in the development of new computing applications and applications areas and the transfer of computing technology to new problem domains. SIGAPP offers practitioners and researchers the opportunity to share mutual interests in innovative application fields, technology transfer, experimental computing, strategic research, and the management of computing. SIGAPP also promotes widespread cooperation among business, government, and academic computing activities. Its annual Symposium on Applied Computing (SAC) provides an international forum for presentation of the results of strategic research and experimentation for this inter-disciplinary environment. SIGAPP membership fees are: $30.00 for ACM Non-members, $15.00 for ACM Professional Members, and $8.00 for ACM Student Members. For further information on SIGAPP, please contact Barrett Bryant at bryant@cis.uab.edu or visit the SIGAPP website at http://www.acm.org/sigapp.

Service Oriented Architectures and Programming Track

Track Co-Chairs: Claudio Guidi, University of Bologna, Italy
Ivan Lanese, University of Bologna, Italy
Manuel Mazzara, Newcastle University, United Kingdom

Poster Papers

Wireless Sensor Networks Track

Track Co-Chairs: Julie McCann, Imperial College London, United Kingdom
Richard Anthony, Grenwich University, United Kingdom

Poster Papers

Message from the Symposium Chairs

On behalf of the Organizing Committee, we welcome you to the 24th Annual ACM Symposium on Applied Computing (SAC 2009) hosted by Chaminade University in Hawaii. This international forum has been dedicated to computer scientists, engineers and practitioners for the purpose of presenting their findings and research results in various areas of computer applications. The organizing committee is grateful for your participation in this exiting international event. We hope that this conference proves interesting and beneficial.

The Symposium is sponsored by the ACM Special Interest Group on Applied Computing (SIGAPP), whose mission is to further the interests of computing professionals engaged in the design and development of new computing applications, interdisciplinary applications areas, and applied research. This conference is dedicated to the study of applied research of real-world problems. This event provides an avenue to discuss and exchange new ideas in the wide spectrum of application areas. We all recognize the importance of keeping up with the latest developments in our current areas of expedites.

SAC 2009 offers Technical Tracks and Posters. The success of the conference can be attributed to the substantial contribution of talented Track Chairs and Co-Chairs. Each track maintains a program committee and a set of highly qualified reviewers. We wish to thank the Track Chairs, Co-Chairs, Committee Members and participating reviewers for their hard work and effort to make the SAC 2009 conference a high quality conference. We also thank our invited keynote speakers, Dr. Vahid Tarokh, Harvard University and Dr. Rolf-Peter Kudritzki, University of Hawaii's Institute for Astronomy for sharing their knowledge with SAC attendees. Most of all, special thanks to the authors and presenters for sharing their experience with the rest of us and to all attendees for joining us in Honolulu, Hawaii this year.

The local organizing committee has always been a central contributor to the success of the SAC 2009 conference. Our gratitude goes to the Conference Vice-Chair Dr. Paulo Martins of Chaminade University and Local Chair Dr. Martha Crosby of University of Hawaii at Mānoa. We also extend our thanks to the Publication Chair, Dr. Dongwan Shin, New Mexico Tech for his tremendous effort in putting together the conference proceedings, Posters Chair Dr. Jiman Hong of Soongsil University for his hard work to make a successful Poster Program, Publicity Chair, Dr. Udo Fritzke, PUC-Minas for his hard work, and eConference Management Chair, Dr Mathew J. Palakal of Indiana University Purdue University for successfully maintaining the eCMS system. A special thanks goes to our Program Chairs Dr. Mirko Viroli, Università di Bologna and Dr. Ronaldo Menezes, Florida Institute of Technology for coordinating and bringing together an excellent Technical Program.

Again, we welcome you to SAC 2009 and the beautiful city of Honolulu, Hawaii. We hope you enjoy the SAC 2009 conference and your stay in Hawaii. Next year, we invite you to participate in SAC 2010 to be held in Crans Montana, Switzerland. The conference will be hosted by the University of Applied Sciences of Western Switzerland.

Sung Y. Shin and Sascha Ossowski
SAC 2009 Conference Chairs

PRECISO: A Reengineering Process and a Tool for Database Modernisation through Web Services

Ricardo P. del Castillo
Alarcos Research Group
University of Castilla-La Mancha
Paseo de la Universidad, nº4 13071
Ciudad Real, Spain
+34926295300
ricardo.pdelcastillo@uclm.es

Ignacio García-Rodríguez
Alarcos Research Group
University of Castilla-La Mancha
Paseo de la Universidad, nº4 13071
Ciudad Real, Spain
+34926295300
ignacio.grodriguez@uclm.es

Ismael Caballero
Alarcos Research Group
University of Castilla-La Mancha
Paseo de la Universidad, nº4 13071
Ciudad Real, Spain
+34926295300
Ismael.caballero@uclm.es

ABSTRACT

A common trend in *Service Oriented Architecture* (SOA) is to consider Information Systems exposing software as services. This current approach is not only applied to new software developments, but also it is related to the maintenance of legacy systems. Nowadays, a cornerstone of Information Systems are relational databases, which constitute meaningful sources of services. These services can provide database's information in SOA scenarios. This paper presents a reengineering process to recover and implement Web Services in automatic manner from relational databases. This process follows the ADM approach (*Architecture-Driven Modernization*). In this paper authors present a case study that has been carried out using a tool built to support the process. This tool is used to generate a set of Web Services which are integrated into a web development allowing to modernise the legacy database in a SOA context. This case study has been carried out in the context of software company Indra.

Categories and Subject Descriptors

D.2.2 [**Software Engineering**]: Design Tools and Techniques – *Computer-aided software engineering (CASE)*. D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement – *Restructuring, reverse engineering, and reengineering*.

General Terms

Algorithms, Design and Experimentation.

Keywords

Web Services, SOA, software modernisation, ADM, relational databases, MDA

1. INTRODUCTION

Nowadays, in the globalized world, organizations are increasingly forced to share more and more information as a basic activity in their daily operation [32]. Besides, the heterogeneity of Information Systems (IS) is growing every day due to the appearance of new technological environments, paradigms and standards [11, 12]. As a consequence of this fast technological evolution and high level of uncertainty in these markets (and in order to keep their competitiveness level through their Information System), organizations are involved in a process of continuous renewal [13].

Under these circumstances, developers of IS based on ICT (Information and Communication Technologies) are required to make shorter developments and maintenances [11]. This acceleration in the development process involves reusing components and software artefacts already existing in the organization [33]. The current Information Systems consist of several software artefacts. However, databases are possibly considered as one of the most fundamental assets since they contain all organization's information. Therefore, databases turn out to be the basis of decision-making at the operational, tactic and strategic levels.

Re-engineering has emerged as a powerful and accepted method to address the necessary evolution of IS in terms of migration and reuse of its artefacts (for example, to target environments such as the Web) [13].

Moreover, MDA (*Model-Driven Architecture*) is influencing the software development, rising it to higher abstraction level [26]. MDA considers each system or each piece of systems as models. Later on, transformations can be established among these models. Besides, ADM (*Architecture-Driven Modernisation*) appears to carry out re-engineering processes which follow the MDA approach [27]. This paper presents a re-engineering process based on ADM approach with the following performances: (1) recovery functionalities in data sources (these sources are typically relational databases); (2) functionalities are transferred towards services; and (3) these services are exposed through Web Services. Furthermore, a tool is implemented to validate the modernisation process. This tool performs the detection, implementation, setup and deployment of Web Services in an automatic manner.

The remainder of this paper is organized as follows. Section 2 summarizes related work. Section 3 focuses on proposed modernisation process. Section 4 presents the implemented tool and Section 5 shows its use in a Web development. Finally, section 6 addresses the conclusions of this paper as well as the future work.

SAC'09, March 8-12, 2009, Honolulu, Hawaii, U.S.A.

2. RELATED WORK

To improve the understanding of paper, the re-engineering concept should be remembered. The re-engineering is *"the examination and alteration of a subject system to reconstitute it in a new form and the subsequent implementation of the new form"* [1]. According to [9], the process has three stages: (1) reverse engineering in order to recover abstract representation of subject system; (2) restructuring to modify the system at same abstraction level; and (3) forward engineering to address the generation of new system version containing new features.

Existing research, about re-engineering on applications and database jointly, has usually been focused on certain aspects. One of them is SQL code embedded in applications or the extraction of functionalities from PL/SQL code [31]. The migration of database models [2]. The design recovery of database [16]. The integration of database based on different data models through wrapping techniques [22]. Building database-driven applications [30]; and so on. Nevertheless, there is little research on detection of services from databases.

Keeping re-engineering process in mind, the first stage typically focuses on achieving a set of abstract specifications in order to generate a new system with the new requirements. However, there are instances where the creation of a new database is not required, but to wrap database through an interface for access to it, with no need to restructure it [35]. These techniques are called *'wrapping techniques'* and them consist in building software components usually wrapping a DBMS. This is accomplished by transforming the requests to subject data model in another target model which is independent from DBMS [34]. In this way, databases could be integrated into new IS for which they were not initially designed [35]. The database life cycle is therefore extended.

Besides, *Model-Driven Architecture* (MDA) [26] advocates for IS developments based on models. MDA converts one model into another model of the same system through transformations. Moreover, MDA automatically generates source code from subject model. This is not a novel idea; according to [21], interest in *'Write Once, Run Anywhere'* approach has been shifted to *'Model Once, Generate Anywhere'* approach. MDA addresses some challenges arising from IS heterogeneity, since according to MDA, a system can be represented at different abstraction levels through different models [26]. Therefore, on the one hand, a system could be represented through a model at business level which depicts its functionalities. It is called *Platform-Independent Model* (PIM) [26]. And on the other hand, transformations are performed from PIM to achieve models at technological platform level which support specific details of each platform. It is called *Platform-Specific Model* (PSM) [26]. Finally, source code is generated from certain PSM models. In MDA approach each model represents one system and each model is depicted according to one meta-model. Meta-models are models which allow representing models.

Further, several works about re-engineering focused on MDA approach are frequently found in literature. This is known as *Architecture-Driven Modernisation* (ADM) [25, 27]. ADM is the evolution of MDA. ADM intends to carry out re-engineering processes which take into account different models as input and output artefacts of these processes. Some research in this direction is being carried out in both academic and industrial environments [17, 24].

Organizations feel increasingly compelled to adopt the new market viewpoint which is service-oriented. This new paradigm emerges in order to separate *possession and ownership concept* of *use concept* [36]; SOA defends just this approach [6]. A particular *implementation* of SOA is the Web Services technology [8, 18].

On this other side of the spectrum, in terms of re-engineering processes toward Web Services generation, relevant work can also be found in literature. Sneed in [33] proposes a re-engineering process obtaining Web Services from legacy COBOL applications. In [3] a MDA process is depicted transforming PIM models according to UML2 [28] meta-model toward several PSM models, one of them to support the generation of Web Services. In other works as [10], re-engineering processes are carried out on legacy systems taking Web Services as a major construction unit.

After presenting theoretical background of this work as well as related work, the next section will depict the proposed process in this paper.

3. PRECISO: THE ADM PROCESS

The process aims to establish guidelines to allow the generation of Web services from relational databases through re-engineering process on MDA artefacts, i.e. a modernization process. Figure 1 represents the proposed re-engineering process focused on ADM approach: First, a legacy relational database is the input of the process. A PSM model, according to SQL-92 meta-model [20], is afterward obtained from input database through reverse engineering. Then, the PSM is transformed to PIM model which raises the abstraction level of the system. This PIM model is represented in terms of UML2 meta-model [28]. The process generates a certain PSM model from PIM model through forward engineering. This PSM model depicts Web Services, and abstraction level is thus reduced again.

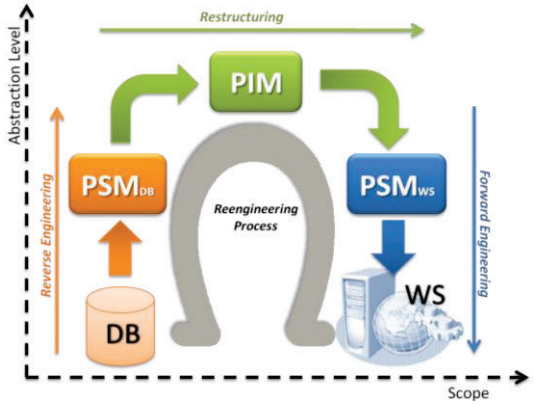


Figure 1. Re-engineering process according to ADM approach

The depicted scheme in Figure 1 is the structure of the proposed process. Figure 2 depicts the modernisation process which consists of three main activities. Each major activity is broken down into a set of tasks; these tasks are partially arranged. Each activity is detailed in the following subsections.

3.1 Database Model Recovery

The first activity aims to create a PSM model that represents the input database. In addition, the information extraction on database schema is used to discover potential services. Tasks involved in this activity are detailed in following paragraphs.

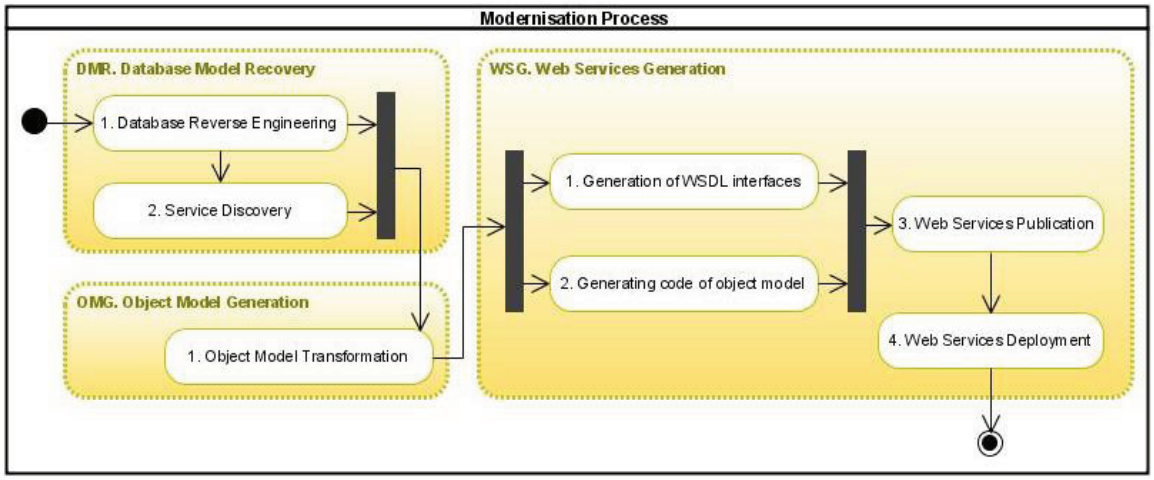


Figure 2. Proposed ADM process to generate Web Services from databases.

DMR 1. Database Reverse Engineering. The first task of the modernisation process is the recovery of relational database design through reverse engineering. Recovered metadata of database will make up a PSM model according to SQL-92 meta-model (see Figure 3), based on [7].

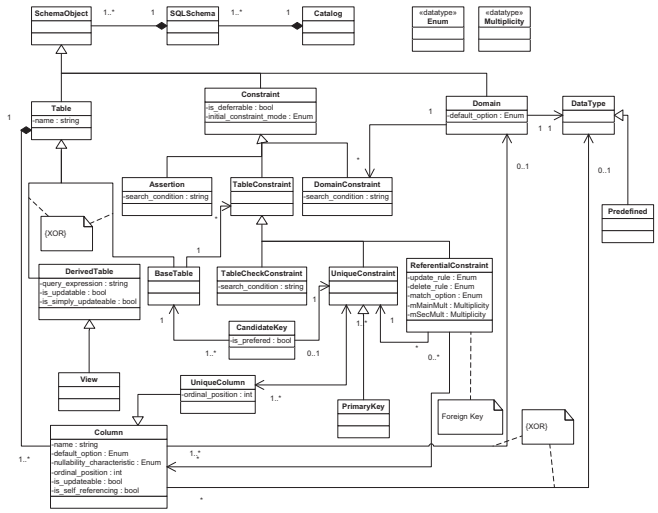


Figure 3. SQL-92 meta-model

DMR 2. Service Discovery. Potential services can be simultaneously discovered along the previous task. Certain patterns are sought in recovered database schema. Well known services are inferred from these patterns. According to [14] it is called *'Model Driven Pattern Matching'* (MDPEM).

3.2 Object Model Generation

The second activity generates an object model through obtained information from database recovery. This model of objects will be afterward the basis to generate Web Services in followings activities. This activity has a single task.

OMG 1. Object Model Transformation. This task carries out transformation *PSM* → *PIM* which involves respectively the model of relational database schema and the object model (see Figure 1). Object model is developed according to UML2 meta-model [28]. Transformations can be formally established through specific languages for defining transformations among models such as QVT (*Queries / Views / Transformations*) [29] o ATL (*ATLAS Transformation Language*) [19]. On the contrary, the transformations can also be described manually through source code when a tool is being implemented to support this modernization process.

3.3 Web Services Generation

The third activity of the process is considered as *'front-end'* activity which finally generates the Web Services to manage the initial input database.

WSG 1. Generation of WSDL interfaces. This task drops the abstraction level obtaining PSM model that supports Web Services (see Figure 1). This new PSM model is achieved through two input artefacts: the PIM representing the object model and discovered services through patterns. These patterns are summarized in Table 1. This task will generate the PSM model according to WSDL meta-model (*Web Services Description Language*) [37].

WSG 2. Generating code of object model. This task generates source code to support the object model obtained in previous tasks. This code will be the basis for implementing the infrastructure of Web Services.

WSG 3. Web Services Publication. Web Services are built through source code of object model and descriptions of WSDL interfaces. It also publishes a set of services belong to candidate services discovered from database schema through MDPEM [14].

WSG 4. Web Services Deployment. Web Services are finally deployed moving to production, thus these services become in fully operational services.

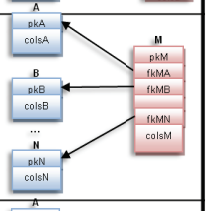
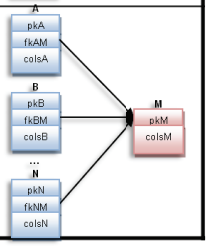
So far, general description of the proposed modernization process has been detailed. The implemented tool is presented as follows.

4. PRECISO: THE TOOL

4.1 Developed Tool

A tool has been implemented in order to support the mentioned modernisation process. This tool automates tasks of the process to carry out the Web Services generation from relational databases.

Table 1. Patterns and candidate services which are examined in relational database schema

Simple Services	Tables	CRUD operations		
		Getters & Setters Operations		
Advanced Services	Views	Queries		
	Patterns in database schema	*Referenced Table*	Select_A_of_B (pkB) Select_B_for_A (pkA)	
		Combined Table	Select_A_for_B (pkB) Select_A_for_B_filtrado (pkB, colsM) Exists_A_related_with_B (pkA, pkB) Select_A_for_B_and_C (pkB, pkC) Select_A_for_B_and_C_filtrado(pkB, pkC, colsM) Exists_A_related_with_B(pkA, pkB, pkC)	
		Observed Table	Select_A_for_B (pkB) Select_B_for_A (pkA)	

4.1.1 Database Model Recovery

Necessary functionality to recover database models is provided to the tool. Candidate services are also carried out in conjunction.

DMR 1. Database Reverse Engineering. Models derived from databases will be represented according to SQL-92 meta-model which is based on the meta-model presented in [7]. The tool has been developed using SQL-92 standard [20], since according to some studies these are the kind of databases more widely used in software industry [4, 5]. However, the PSM model may be built from other database management system whose data model are other standards such as SQL-86, hierarchical databases, databases based on COBOL, and so on. For this purpose, tool would only need using the appropriate meta-model to faithfully represent the model.

Metadata needed in SQL databases-92 to build the database schema model can be taken through *INFORMATION_SCHEMA* [23]. This is a standardized mechanism that identifies a set of views. These views return metadata on a standardized manner. In addition, the built model can be made persistent through XMI (*XML Metadata Interchange*) [15]. XMI facilitates their safe handling and integration within the overall process.

DMR 2. Service Discovery. The obtained database model is examined and candidate services are inferred through certain patterns based on MDPEM techniques [14]. Table 1 details both search patterns as well as services that can be derived from each pattern. On the one hand, there are simple services involving only a single table. These services are directly obtained from database scheme and matched with *CRUD* operations (*Create / Read / Update / Delete*) as well as *getters & setters* methods for handling various columns on each table. And on the other hand, advanced services involve several tables of schema (see Table 1). In this case, services may be directly obtained from views, or on the contrary, services could be obtained from the following patterns that are recognized in the relational database scheme: (1) *referenced table*, when there is a foreign key among two tables; (2) *combined table*, when there are two or more foreign keys from one table to other, likewise (3) *observed table*, unlike the previous one, this pattern searches two or more foreign keys to the same table.

4.1.2 Object Model Generation

The next activity is implemented in the tool allowing the object model generation.

OMG 1. Object Model Transformation. The tool at this stage must address the SQL-92 model transformation to the UML2 model, i.e. a *PSM→PIM* transformation. These transformations could be specified for example through QVT. These transformations could be afterward implemented in the tool. However, in this tool, an existing algorithm that is depicted in [30] has been used. Realized transformations are as follows: a table → a class, a column → an attribute, a foreign key → association, and so on. Moreover, SQL-92 data types are mapped on data types of generic programming language which is used to generate source code.

4.1.3 Web Services Generation

Finally, the tool allows publication and deployment of Web Services. The implementation details are presented in the following paragraphs.

WSG 1. Generation of WSDL interfaces. This task takes the obtained object model likewise the discovered services as its inputs. And it generates Web Service descriptions through WSDL interfaces as its output.

WSG 2. Generating code of object model. In order to get executable Web Services, tool must write source code of object model in hard disk through a programming language supporting Web Services technology. In this tool *C#* was used. Moreover, the tool offers the user option of modifying the source code of object.

WSG 3. Web Services Publication. The tool allows selecting services among discovered services. User selects services which

will compose the future generated Web Service. Due to security reasons, as well as others, it is not advisable to open the entire database through publishing certain services that will not be used. Moreover, tool can generate several Web Services with different services each of them. Each Web Service can show different points of view of the database.

WSG 4. Web Services Deployment. In this last stage the tool configures the built Web Services to enable them to execute requests on the Web. First, a certain application server should be selected to be used. The tool is ready to interact with *Microsoft Internet Information Server 6* (IIS6). According to the used server, the deployment may vary. Nevertheless there are two responsibilities that are always carried out: (1) source code files are copied in the correct location of the application server; and (2) this source code is stated as a Web directory accessible on the Web.

4.1.4 Other aspects to highlight

The tool semi-automates several tasks in the proposed modernization process (see Figure 2). But in addition, the tool must address other issues such as remote database connection, connections to databases from different manufacturers, project management, graphical display of involved models, testing, reporting, and so on. The proposed architecture, taking into account the previous challenges, is shown in Figure 4. Likewise the skin of the developed tool can be seen in Figure 5.

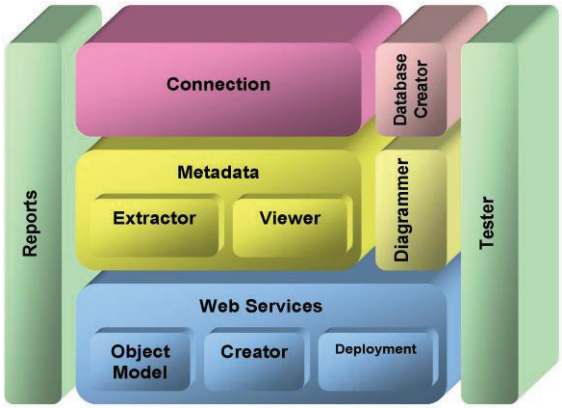


Figure 4. Architecture of the developed tool

5. CASE STUDY

The case study consists on a project which was jointly carried out by the University of Castilla-La Mancha and Indra Software Labs (a multinational software company) in the context of the *"CATEDRA INDRA"*, a R+D centre (located in Spain) devoted to carry out research projects in a close cooperation between industry and university. This research centre is supported by the *'University of Castilla-La Mancha'* (*UCLM*) and *'INDRA Software Labs at Ciudad Real'*.

There was a need for CATEDRA INDRA (CI hereinafter) to develop its corporate Web site in order to support all the information produced from the cooperation of industry & university. This site is addressed to academics, researchers, teachers, PhD candidates and students. The site contains information about conferences, lectures, courses, grants offered, events, awards, papers, journals and so on.

The website has been built using a standard Web architecture based on the *Microsoft .NET* platform. On the one hand, it has used *Microsoft Content Management Server 2002 (MCMS)* as a content management system (under the RDBMS *Microsoft SQL Server 2000*). And on the other hand, *Active Server Pages (ASP)* has been used for the presentation layer. Finally, the whole application (that is, the MCMS and the Web application) has been deployed through *Microsoft Internet Information Server 6 (IIS6).*

Due to size limitations, the case study focuses on a portal's sub-module. This module deals with the tasks to manage the research papers produced by CI. This module will search research articles according to different criteria. Furthermore, this module will add new paper data, modify or delete existing paper data.

5.1.1 Problem Specification

The development staff in charge of this project was interviewed to understand the information needs. They reported the information needs according to established user requirements. This valuable information would help in the usage of the tool to obtain the set of Web Services required to feed with information the Web layer.

In a nutshell, the problem is the following: implement a module to consult information about the publications of the researchers of the CI. That is, it will keep all the information from conferences and journals, data from the authors, R+D projects financed, and so on. Moreover, this module will contain a search engine to set up filters to carry out customized searches according to different criteria such as conference, journal, international/national, whether a conference is a LNCS, whether a journal is an indexed one, authors, and other criteria.

The system is based on an existing database provided by the CI. This database, which was created long time ago, keeps a lot of information from existing publications of people involved in the CI. This information is not managed by any application; therefore, in this context, it is possible to find the suitable conditions for implementing a modernisation process through the tool. Thus, the CI's database is considered as a legacy system. The tool will expose the required functionalities by means of Web Services starting from this legacy system and following the proposed process.

5.1.2 Execution of the case study

Firstly, the tool establishes a connection to the database of the CI. Subsequently, it generates a *xmi file* containing all the metadata about the studied database (see Figure 6). Second step is creating the object model for support the future Web Services. The tool generates necessary classes depending on the metadata obtained in the previous stage. Additionally, the tool writes executable class (in this case, *C#* language). Then, the Web Services are built through the tool. It will have achieved the executable Web Services, using the class model created in the previous step (see Figure 6). The tool allows selective publication of services. Therefore, it is sufficient to make some public services. These expose only the necessary parts of the database, for the development of the project. Finally, the tool converts Web Services in operational Web Services, carrying out the deployment of these in a Web application server. Therefore, it has been achieved suitable Web Services to provide the required functions for handling the database, according to the previously imposed information needs.

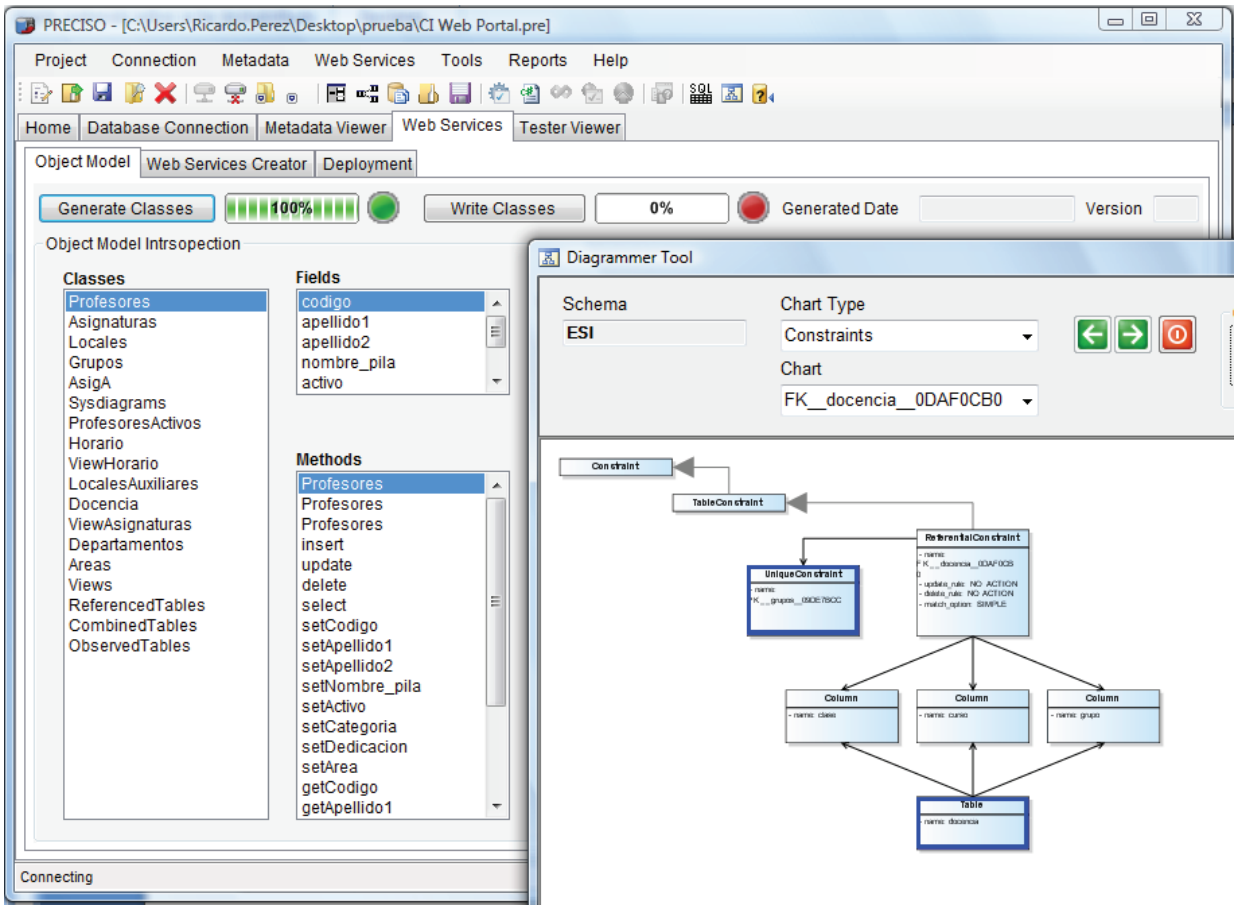


Figure 5. Graphical User Interface of the developed tool

5.1.3 Obtained results

The tool made a selective publication and deployment of the generated Web Services. The Web Services to provide with the required information by the development staff constitute a small set from the total of candidate services discovered from the database. Table 2 shows a summary of this performance. This table considers (1) the different types of services, (2) services generated for each type, (3) the number of services (for each type) included in the sub module of the CI and (4) the percentage of services included. The percentage of services that was published to support the functionalities of the CI portal was 30%. This percentage included 73 services on a total of 245 candidate services.

In addition, the development staff noticed that the non selected Web Services would be very useful for future developments. Since these non considered services were identified and collected, it would be easy to deploy and integrate them into the CI Web application for the implementation of additional features. The tool keeps information of the modernisation project, that is, the object model, configuration of the generated services, services deployed and services available. Thus, we only need to load the project of the current case study to deploy new Web Services to fulfil new information requirements.

The result is an operational Web Service which handles the legacy database. The Web Service supports the information needs in a SOA context such as CI portal. At this point, the CI portal can carry out the required functionalities by means of the new Web Services.

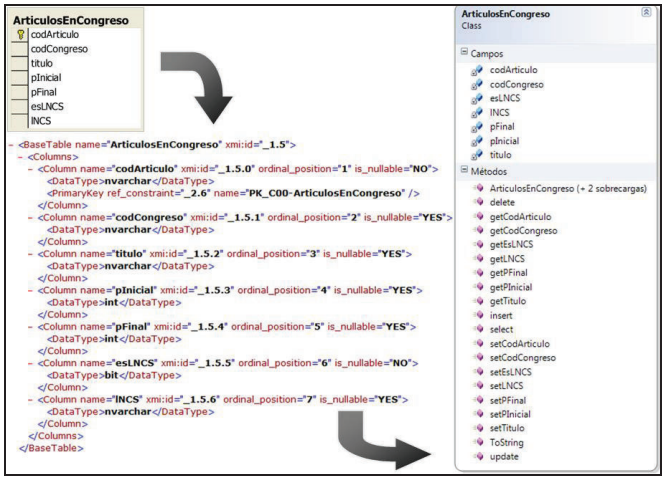


Figure 6. Generated artefacts throughout the process

The development staff took advantages from the availability of the required information to improve the development process: since all the required information was available as services, the staff

could work with real data when developing the Web application. The staff could put all their effort into the development of the Web interface. Furthermore, since the required information was available from the first, all the features of the CI Web application could be tested with the real information of the database. It allowed the staff to accelerate the testing process, because web developers could build the necessary web interfaces which in turn will use the aforementioned Web Services, and will properly display the information.

Table 2. Performance of published services in CI

Kind of Service	Candidate Services	Published Services	Performance
insert	13	11	85%
update	13	11	85%
delete	13	11	85%
select	13	12	92%
setters	56	0	0%
getters	56	0	0%
show	13	0	0%
views	4	4	100%
referenced tables	24	12	50%
combined tables	32	8	25%
observed tables	8	4	50%
TOTAL	245	73	30%

6. CONCLUSION AND FUTURE WORK

This paper has proposed an ADM process to automatically generate Web Services from relational databases. This ADM process has allowed modernizing legacy systems such as databases to deliver its functionality in services manner. Thus, databases can be integrated into SOA environments. In addition, it has built a tool supporting this process.

The modernization process includes a set of meta-models to represent models involved in every task. For example, SQL-92 meta-model has been used to represent the database model (or PSM model). Moreover, UML2 meta-model has been utilized to represent the system (or PIM model), among other meta-models.

In order to empirically validate the process, this paper has also presented a case study in an industrial context. The developed tool was used to carry out a modernization process within the development of a Web portal. This case study revealed a number of advantages of the proposed process such as development process acceleration, easy integration into SOA environments and certain improvements in testing.

The future extensions to this research focus on two key aspects. (1) It will carry out an in depth analysis in order to infer services based on searching of more patterns into database scheme. (2) It will formalize transformations among models through specific-purpose languages such as QVT or ATL languages. Indeed, new versions of the tool supporting new advances in this research will be further developed.

7. ACKNOWLEDGMENTS

This paper has been partially supported by the project PRALIN (PAC08-0121-1374), Junta de Comunidades de Castilla-La Mancha; the project ESFINGE (TIN2006-15175-C05-05/), Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica. Programa Nacional de Tecnologías de la Información; and the project MECENAS (PBI06-0024), Plan Regional de Investigación Científica, Desarrollo Tecnológico e Innovación.

7. ADDITIONAL AUTHORS

Macario Polo
Alarcos Research Group, University of Castilla-La Mancha
Pº de la Universidad, nº4 13071, Ciudad Real, Spain +34926295300
email: macario.polo@uclm.es
Mario Piattini
Alarcos Research Group, University of Castilla-La Mancha
Pº de la Universidad, nº4 13071, Ciudad Real, Spain +34926295300
email: mario.piattini@uclm.es

8. REFERENCES

[1] Arnold, R.S., Software Reengineering. 1994: IEEE Computer Society Press. 688 pages.

[2] Behm, A., A. Geppert and K. Dittrich, Algebraic Database Migration to Object Technology, in Lecture Notes in Computer Science. Springer Berlin / Heidelberg. 2000.

[3] Bezivin, J., S. Hammoudi, D. Lopes and J. Jouault, Applying MDA Approach for Web Service Platform, in Proceedings of the Enterprise Distributed Object Computing Conference, Eighth IEEE International. 2004, IEEE Computer Society. p. 58-70.

[4] 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. p. 136-147.

[5] Blaha, M. A Retrospective on Industrial Database Reverse Engineering Projects-Part 2. in Proceedings of the 8th Working Conference on Reverse Engineering (WCRE´01). 2001. Suttgart, Germany: IEEE Computer Society. p. 147-156.

[6] Booth, D., H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris and D. Orchard, Web Services Architecture. 2004: W3C - World Wide Web Consortiumpages.

[7] Calero, C., An Ontological Approach To Describe the SQL:2003 Object-Relational Features. Accepted in "Computer Standards and Interfaces". 2005: p. 28.

[8] Channabasavaiah, K., K. Holley and E. Tuggle, Migrating to a service-oriented architecture, Part 1. http://www.ibm.com/developerworks/library/ws-migratesoa/. 2003.

[9] Chikofsky, E.J.a.J.H.C., Reverse Engineering and Design Recovery: ATaxonomy. IEEE Software(January), 1990: p. 13-17.

[10] Chung, S., P.S. Young and J. Nelson, Service-Oriented Software Reengineering: Bertie3 as Web Services, in Proceedings of the IEEE International Conference on Web Services. 2005, IEEE Computer Society.

[11] Di Lucca, G.A., A.R. Fasolino and P. Tramontana, Reverse engineering Web applications: the WARE approach. Journal of Software Maintenance and Evolution: Research and Practice 16, 2004.

[12] Endrel, M., J. Ang, A. Arsanjani, S. Chua, P. Comte, P. Krogdahi, M. Luo and T. Newling, Patterns: Service-

Oriented Architecture and Web Services. 2004: IBM - WebSphere Software.pages.

[13] García, I., PRESSWEB: Un Proceso para REingeniería de Sistemas heredados hacia Servicios Web. 2007, Universidad de Castilla-La Mancha. p. 344.

[14] García, I., M. Polo and M. Piattini, Using Model-Driven Pattern Matching to derive functionalities in Models., in Proceedings of the Nineteenth International Conference on Software Engineering and Knowledge Engineering. 2007: Boston, USA.

[15] Grose, T.J., G.C. Doney and S.A. Brodsky, Mastering XMI: Java Programming with XMI, XML, and UML, ed. O. Press. 2001: John Wiley & Sons. 480 pages.

[16] Hainaut, J.-L., V. Englebert, J. Henrard, J.-M. Hick and D. Roland1, Database reverse engineering: From requirements to CARE tools, in Applied Categorical Structures. SpringerLink. 2004.

[17] Heuvel, W.-J.v.d., Matching and Adaptation: Core Techniques for MDA-(ADM)-driven Integration of new Business Applications with Wrapped Legacy Systems, in Model-Driven Evolution of Legacy Systems (MELS 2004). 2004: Monterey, California, USA.

[18] IBM, New to SOA and Web services. http://www.ibm.com/developerworks/webservices/newto/. 2006: p. 6.

[19] INRIA, ATL Transformation Description Template version 0.1. http://www.eclipse.org/m2m/atl/doc/ATL_Transformation_Template%5Bv00.01%5D.pdf. 2005, ATLAS group.

[20] ISO/IEC, ISO/IEC 9075:1992, Database Language SQL. 1992.

[21] Jean Bezivin and O. Gerb, Towards a Precise Definition of the OMG/MDA Framework, in Proceedings of the 16th IEEE international conference on Automated software engineering. 2001, IEEE Computer Society.

[22] McBrien, P. and A. Poulovassilis, Automatic Migration and Wrapping of Database Applications - A Schema Transformation Approach, in Proceedings of the 18th International Conference on Conceptual Modeling. 1999, Springer-Verlag.

[23] Melton, J. and A.R. Simon, Understanding the new SQL: A Complete Guide. 1993, United States of America: Morgan Kaufmann Publishers, Inc.pages.

[24] MOMOCS. MOdel driven MOdernisation of Complex Systems Is an EU-Project. 2008 [accessed: 2008; Disponible en: http://www.momocs.org/.

[25] Newcomb, P., Architecture-Driven Modernization (ADM), in Proceedings of the 12th Working Conference on Reverse Engineering. 2005, IEEE Computer Society.

[26] OMG, MDA Guide Version 1.0.1. www.omg.org/docs/omg/03-06-01.pdf, OMG, Editor. 2003. p. 62.

[27] OMG, Architecture-Driven Modernization Roadmap. 2006, Object Management Group.

[28] OMG. Unified Modeling Language: Superstructure. Version 2.0. http://www.omg.org/docs/formal/05-07-04.pdf. 2007 [accessed: 16-08-2007]; Disponible en: http://www.omg.org/docs/formal/05-07-04.pdf.

[29] OMG, QVT. Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification. http://www.omg.org/spec/QVT/1.0/PDF. 2008, OMG.

[30] Polo, M., J.Á. Gómez, M. Piattini and F. Ruiz, Generating three-tier applications from relational databases: a formal and practical approach. Information and Software Technology, 2002. 44.

[31] Reus, T., H. Geers and A.v. Deursen. Harvesting Software for MDA-Based Recovering. in European Conference on Model Driven Architecture - Foundations and Applications. 2006. Bilbao (Spain): Springer-Verlag Berlin Heidelberg.

[32] Shankaranarayanan, G. and Y. Cai, A Web Services Application for the Data Quality Management in the B2B Networked Environment., in Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 7 - Volume 07. 2005, IEEE Computer Society.

[33] Sneed, H.M., Migrating to Web Services, in Emerging Methods, Technologies and Process Management in Software Engineering. 2008, Wiley-IEEE Computer Society Pr. p. 151-176.

[34] Thiran, P. and J.-L. Hainaut, Wrapper Development for Legacy Data Reuse in Proceedings of the Eighth Working Conference on Reverse Engineering (WCRE'01) 2001 IEEE Computer Society. p. 198

[35] Thiran, P., G.-J. Houben, J.-L. Hainaut and D. Benslimane, Updating Legacy Databases through Wrappers: Data Consistency Management in Proceedings of the 11th Working Conference on Reverse Engineering (WCRE'04) - Volume 00 2004 IEEE Computer Society. p. 58-67

[36] Turner, M., D. Budgen and P. Brereton, Turning Software into a Service, in IEEE Computer Society. 2003pages.

[37] W3C. WSDL in Web Services Description Working Group. 2007 [accessed: 2008 08/01/2008]; Disponible en: http://www.w3.org/2002/ws/desc/.