CSMR 2007

11th European Conference on Software Maintenance and Reengineering

"Software Evolution in Complex Software Intensive Systems"

Editors:
René Krikhaar
Chris Verhoef
Giuseppe Di Lucca

21-23 March 2007
Amsterdam, the Netherlands

Sponsors:

IEEE Computer Society

IEEE

Reengineering Forum

In cooperation with:

PARASOFT
We make software work.

ICT NoviQ
Advanced Thinking

vrije Universiteit amsterdam

Rabobank

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 P2802
ISSN 1534-5351
ISBN 0-7695-2802-3

ISBN 0-7695-2802-3
90000
9 780769 528021

Proceedings

CSMR

2007

11th European Conference on Software Maintenance and Reengineering

Software Evolution in Complex Software Intensive Systems

Sponsored by:

Reengineering Forum

IEEE
COMPUTER
SOCIETY

In cooperation with:

Rabobank

PARASOFT®
We make software work.™

ICTNoviQ
ADVANCED THINKING

vrije Universiteit amsterdam

Proceedings

11th European Conference on Software Maintenance and Reengineering

Software Evolution in Complex Software Intensive Systems

21-23 March 2007
Amsterdam, the Netherlands

Editors
René Krikhaar
Chris Verhoef
Giuseppe Antonia Di Lucca

IEEE COMPUTER SOCIETY
IEEE

Los Alamitos, California
Washington • Tokyo

CPS Conference Publishing Services

The papers in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change. Their inclusion in this publication does not necessarily constitute endorsement by the editors, the IEEE Computer Society, or the Institute of Electrical and Electronics Engineers, Inc.

IEEE Computer Society Order Number P2802
ISBN 0-7695-2802-3
ISBN 978-0-7695-2802-1
ISSN Number 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 Lisa O'Conner
Cover art production by Alex Torres
Printed in the United States of America by The Printing House

Conference Publishing Services
http://www.computer.org/proceedings/

Table of Contents

Technical Papers

Technical Session 1: Maintenance Changes

Technical Session 2: Components and Objects

Technical Session 3: Dependency Analysis

Technical Session 4: Maintenance with Dynamics

Technical Session 5: Architecture

Technical Session 6: Web Applications

Technical Session 7: Quality

Technical Session 8: Static Analysis

Technical Session 9: Features

Technical Session 10: Recovery

Industry Track

Tool Track

Doctoral Symposium

Welcome from the Conference Chairs

Welcome to the 11th edition of the Conference on Software Maintenance and Reengineering (CSMR 2007), the leading European conference on the theory and practice of maintenance, reengineering, and the evolution of software systems.

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.

The main theme of this year's conference is "Software Evolution in Complex Software Intensive Systems". Usually, during its life-time, software grows and grows: Maintenance and Reengineering becomes a burden when they are not explicitly dealt with along the development of large complex and intensive software systems. Innovative methods and techniques are needed to face the problems raised in maintaining and reengineering the complex software systems developed today.

The conference will provide an atmosphere in which information can be easily shared between researchers and practitioners. Beside the technical aspects of maintenance and reengineering, other aspects play a role when enhancing complex intensive systems in industry. One of the key issues, in practice, concerns the cost and benefit of maintenance and reengineering activities. It is this reason, therefore, that the CSMR is organized together with the first IEEE Computer Society Conference on Exploring Quantifiable Information Technology Yields (IEEE EQUITY 2007). We are proud to have an interesting keynote which combines the topics of both conferences.

Dr. Jan Pieter Herweijer - former CIO of Ahold and AKZO - will provide a keynote for both CSMR and IEEE EQUITY on their shared day. The keynote will contain both aspects of IT-value and IT-portfolio management applied to large-scale reengineering work in practice.

The CSMR is happy to have in the program the Stevens Lecture. Nicholas Zvegintzov, well known in the software maintenance community, will receive the 2007 Wayne Stevens Award. Nicholas will talk on Software Development Methods.

In the technical track, eighty-five technical papers, from countries across the world, were submitted. Each paper has been reviewed at least by three members of the Program Committee and twenty-nine full papers were accepted for presentation at the Conference and inclusion in these proceedings.

The objective of the industrial track is to have software intensive companies' reports on real-world applications of techniques for software maintenance and reengineering. The industrial abstracts presented in this proceedings shed light on how software evolution challenges are perceived by key players in today's software industry and illustrating how these techniques are put in practice in large software reengineering projects.

The tool track highlights innovative approaches to issues highly relevant to controlling software evolution. Through a discussion of technical and practical aspects of current solutions to a wide-ranging set of challenges, this track illustrates how resourceful software evolution researchers are operationalizing theoretic contributions.

The Doctoral Symposium Committee was pleased to receive 11 submissions, out of which 6 were selected for presentation at the conference. Each submission was reviewed by two reviewers from the Doctoral Symposium Committee. During the Doctoral Symposium, the starting PhD students will present their initial results and ambitious plans, and the PhDs who are almost there will present their best results of four years of research. Furthermore, the session will feature advice on what (and not) to do when trying to complete a PhD thesis.

We are grateful to the many people and companies who are willing to support this conference. First of all, we want to thank Rabobank, ParaSoft, and ICT NoviQ for providing support and resources for the CSMR

2007. Collaboration with the ReEngineering Forum (REF) was great, we especially thank Elliot Chikofsky for his support in organizing this conference. Elliot's enthusiasm and all kind of initiatives made it possible to provide you the program. Of course, the Vrije Universiteit Amsterdam was willing to organize this conference in Amsterdam, and we thank the various employees who did some extra effort to make it happen.

We also want to thank the many people who helped to organize this conference: Elly Lammers for the local arrangements and in the role of financial chair and Erald Kulk for his tremendous job in handling all submissions and building the website for the CSMR 2007.

Welcome to Amsterdam and enjoy the Conference!

René Krikhaar – *General Chair, Editor*
Chris Verhoef - *Program Chair, Editor*
Giuseppe Antonio Di Lucca - *Program Co-Chair, Editor*
Bart du Bois - *Industrial and Tool Chair*
Arie van Deursen - *Doctoral Symposium Chair*
Harry Sneed – *Workshop and Special Session Chair*

Welcome from the Workshop and Special Session Chair

This years CSMR conference is offering four special sessions. They represent a cross section of the maintenance and reengineering community, both with their themes and their sponsors. The sponsors are from France, Italy, England and the USA, all countries which have always contributed to this conference. The themes are representative of current trends:

- Web Maintenance and Reengineering (WMR2007)
- Model-Driven Software Evolution (MDsE2007)
- System Quality and Maintainability (SQM2007)
- Research Agenda for Service-Oriented Architecture Maintenance (SOAM2007)

The session on web maintenance and reengineering is a workshop for evaluating, identifying, and discussing the maintenance, reengineering, and reverse engineering of web software, including web sites, web applications, and web services. Besides exploring how web software differs from traditional software, it also addresses the vital issue of how to combine rapid evolution while preserving quality.

The session on model-driven software evolution offers a forum for discussion of how model-driven software engineering (MDE) can contribute to the software evolution process. The evolution of existing software systems is inevitable if these systems are to survive. On the other hand, this evolution needs to be controlled and managed to restrict increasing complexity and decreasing quality. MDE provides the means of accomplishing just that in a systematic manner.

The session on system quality and maintainability deals with the contradictions between end user expectations, vendor business prospects, and software engineering capability, and strives to reconcile them. It raises the question of what software quality really is, especially in terms of maintainability, and how it can be measured. The problem is that of reconciling conflicting views of quality.

The session on a research agenda for maintaining service-oriented architectures copes with the new challenge of how SOAs should be maintained and evolved over time. It is for sure that such enterprise-wide infrastructures will not remain static. The will have to evolve to survive and their evolution must be planned and controlled. The leading researchers of the world are meeting here to discuss how to approach this increasingly acute problem.

Harry Sneed – *Workshop and Special Session Chair*

Organizing Committees

General Chair
René Krikhaar
Vrije Universiteit
Amsterdam, the Netherlands
and
ICT NoviQ
Eindhoven, the Netherlands

Program Chairs
Chris Verhoef
Vrije Universiteit
Amsterdam, the Netherlands

Giuseppe Antonio Di Lucca
University of Sannio, Research Centre On Software Technology
Benevento, Italy

Industrial and Tool Chair
Bart du Bois
University of Antwerp
Antwerp, Belgium

Doctoral Symposium Chair
Arie van Deursen
Delft University of Technology
Delft, the Netherlands

Workshop and Special Sessions Chair
Harry Sneed
ANECON GmbH
Vienna, Austria

Registration Chair
Elliot Chikofsky
Engineering, Management & Integration (EM&I)
Herndon, Virginia, USA

Local Arrangements
Elly Lammers
Vrije Universiteit
Amsterdam, the Netherlands

Website and Submission Chair
Erald Kulk
Vrije Universiteit
Amsterdam, the Netherlands

Program Committee

Nicolas Anquetil	*Universidade Católica de Brasília, Brazil*
Jose L Arciniegas	*Universidad del Cauca, Colombia*
Francoise Balmas	*Universite Paris 8, France*
Fernando Brito e Abreu	*New University of Lisbon, Portugal*
Liz Burd	*University of Durham, United Kingdom*
Gerardo Canfora	*University of Sannio, Italy*
Andrea De Lucia	*University of Salerno, Italy*
Serge Demeyer	*University of Antwerp, Belgium*
Massimiliano Di Penta	*University of Sannio, Italy*
Dirk Draheim	*Freie Universitaet Berlin, Germany*
Stephane Ducasse	*Universiti de Savoie, France*
Juan Carlos Duenas	*University Polytechnic Madrid, Spain*
Jürgen Ebert	*University of Koblenz-Landau, Germany*
Anna Rita Fasolino	*University of Naples Federico II, Italy*
Nicolas Gold	*Department of Computer Science King's College London, United Kingdom*
Tibor Gyimothy	*University of Szeged, Hungary*
Mira Kajko-Mattsson	*Stockholm University and Royal Institute of Technology, Sweden*
Jens Knodel	*Fraunhofer IESE, Germany*
Kostas Kontogiannis	*University of Waterloo, Canada*
Rainer Kosche	*University of Bremen Bremen, Germany*
Martin Krop	*Fachhochschule Solothurn / Nordwestschweiz, Switzerland*
Michele Lanza	*University of Lugano, Switzerland*
Spiros Mancoridis	*Drexel University, USA*
Andrian Marcus	*Wayne State University, USA*
Leon Moonen	*Technical University of Delft, the Netherlands*
Martin Pingzer	*University of Zürich, Switzerland*
Lori Pollock	*University of Delaware, USA*
Claudio Riva	*Nokia Research Centre, Finland*
Tobias Roetschke	*Technical University Darmstadt, Germany*
Harry Sneed	*Software Maintenance & Reengineering Consultant, Hungary*
Paolo Tonella	*ITC - IRST, Trento, Italy*
Fernando Usero	*Abengoa Telvent, Spain*
Mark van den Brand	*Technical University Eindhoven, the Netherlands*
Arie van Deursen	*Delft University of Technology, the Netherlands*
Niels Veerman	*Vrije Universiteit Amsterdam, the Netherlands*
Giuseppe Visaggio	*University of Bari, Italy*

Co-Reviewers

Pasquale Ardimento
Lerina Aversano
Alexandre Bergel
Arpad Beszedes
Nicola Boffoli
Magiel Bruntink
Marta Cimitile
Bas Cornelissen
Marco D'Ambros
Kris De Schutter
Vincenzo Deufemia
Maja D'Hondt
Eelco Dolstra
Bart Du Bois
Fausto Fasano
Rudolf Ferenc
Rita Francese
Lajos Fulop
Miguel Goulão
Carmine Gravino
Orla Greevy
Hans-Gerhard Gross
Connie Haoying Bao
Kees Hemerik
Patrick Knab
Jay Kothari
Adrian Lienhard
Marco Lormans
Mircea Lungu
Marcelo Mendes Marinho
Ali Mesbah
Rocco Oliveto
Hugo Parada
Damien Pollet
Denys Poshyvanyk
Matthias Rieger
Michele Risi
Romain Robbes
Giuseppe Scanniello
Istvan Siket
Tamas Gergely
Tom Tourwe
Porfirio Tramontana
Marcel van Amstel
Tijs van der Storm
Niels Van Eetvelde
Joris Van Geet
Filip Van Rysselberghe
Richard Wettel
Andy Zaidman

Committees

Industrial Track Committee

Paul Klint,	*CWI & University of Amsterdam, the Netherlands*
René Krikhaar,	*ICT NoviQ & Vrije Universiteit Amsterdam, the Netherlands*
Serge Demeyer	*University of Antwerp, Belgium*
William van der Sterren	*Philips Medical Systems, the Netherlands*

Tool Track Committee

Andy Zaidman	*Technical University of Delft, the Netherlands*
Eelco Visser	*Technical University of Delft, the Netherlands*
Martin Pinzger	*University of Zürich, Switzerland*
Matthias RIeger	*University of Antwerp, Belgium*
Steven Klusener	*Vrije Universiteit Amsterdam, the Netherlands*
Tibor Gyimothy	*University of Szeged, Hungary*
Yann-Gaël Guéhéneuc	*University of Montreal, Canada*

Doctoral Symposium Committee

Dirk Beyer	*Ecole Polytechnique Fédérale de Lausanne, Switzerland*
Andrew Malton	*University of Waterloo, Canada*
Martin Pinzger	*University of Zürich, Switzerland*
Tom Tourwe	*Eindhoven University of Technology, the Netherlands*

Steering Committee CSMR

Nicolas Gold (Chair)	*King's College London, UK*
Elliot Chikofsky	*Engineering Management & Integration, USA*
Jurgen Ebert	*University of Koblenz-Landau, Germany*
Harry Sneed	*Software Maintenance & Engineering Consultant, Hungary*
Chris Verhoef	*Vrije Universiteit Amsterdam, the Netherlands*
Giuseppe Visaggio	*University of Bari, Italy*
René Krikhaar	*Vrije Universiteit Amsterdam, the Netherlands*

An approach for mining services in database–oriented applications

Concettina Del Grosso[1], Massimiliano Di Penta[1], and Ignacio Garcia-Rodriguez de Guzman[2]

RCOST — Research Centre on Software Technology
Department of Engineering - University of Sannio
Viale Traiano - 82100 Benevento, Italy
University of Castilla-La Mancha, Spain
tina.delgrosso@unisannio.it, dipenta@unisannio.it, Ignacio.GRodriguez@uclm.es

Abstract

The diffusion of service oriented architectures is slowed down by the lack of enough services available for satisfying service integrator needs. Nevertheless, many features desired by service integrators have already been implemented in existing software systems. To this aim, approaches able to identify potential services into a legacy system source code are highly desirable.

This paper proposes an approach to identify, from database–oriented applications, pieces of functionality to be potentially exported as services. The identification is performed by clustering, through formal concept analysis, queries dynamically extracted by observing interactions between the application and the database. The approach has been assessed by identifying potential services in two Java software systems.

Keywords: service mining, migration towards SOA, Formal Concept Analysis

1 Introduction

Service–Oriented Architectures (SOA) are today becoming more and more popular and constitute a tangible industry trend. The idea of "software used not owned" [23] is radically changing the landscape of software engineering, pushing several useful mechanisms such as automatic service discovery [17], QoS–aware dynamic binding and re–configuration [25] and self–healing services [3].

The diffusion of SOA is going to increase the demand of pieces of functionality, available as services, that integrators can glue into their business processes. Many potential services are just out of there, hidden inside existing software systems, previously developed as stand–alone systems. These systems can easily become a valuable source for potential services to be made available for potential integrators/users.

Several approaches for supporting the migration of pieces of software towards services have been proposed in literature [7, 15, 18]. However, while the existing approaches focus towards migrating identified features or entire systems, there is a need for *componentizing* existing software systems by detecting features the system implements which can be potentially exported. To this aim, other authors proposed approaches based on clustering [16], and by analyzing the interactions between the application and its users [19, 20, 21].

This paper focuses on the identification of features to be exported as services in database–oriented applications, i.e., applications where the most relevant features consist in interactions with the database. The proposed approach analyzes, using Formal Concept Analysis (FCA), database queries dynamically extracted during the execution of the application over its main scenarios. FCA clusters related queries into concepts, suggesting possible services aiming to insert/retrieve information to/from the database. Once identified a feature, existing migration techniques can be used for exposing it as a service. The approach has been evaluated over two case studies, an industrial Java newspaper management application (E–PaperML) and a personal finance management system (JBooks).

The paper is organized as follows. Section 2 describes the proposed approach. Section 3 describes the empirical studies performed to assess the approach, defining it, formulating the research question and defining the variables to be measured, reporting and discussing the results. After a description of related work in Section 4, Section 5 concludes the paper and outlines future research directions.

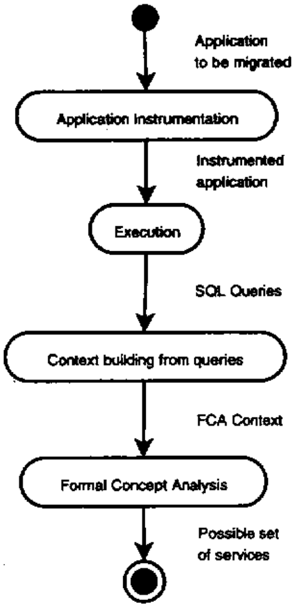

Figure 1. Service mining process

2 The Approach

As mentioned in the introduction, the proposed approach aims to identify application features implemented by querying a database. This is done by clustering queries extracted by observing the interactions between the software system and the database. This section details the whole identification process, also depicted in Figure 1.

To explain the approach, let us consider a very simple working example, consisting in a University book loan application (its database E–R model, represented in UML, is shown in Figure 2.

2.1 Dynamically collecting queries

As a first step, method entry and exit points are instrumented, as well as database interactions via JDBC APIs. The former will be mainly used to identify where a feature to be migrated as a service is located, while the latter (i.e., the queries) will be used to identify the features to be migrated. The instrumented application is executed, following usage scenarios extracted from the application user manual. This would permit the collection of traces related to all application features, although this approach does not account how frequently a feature is used. This is acceptable for this work purposes, since the objective is to extract information for all features – or at least for the most important ones – regardless of how frequently they are used (information that, instead, is useful for clustering purposes [2]).

In our example, let us consider that, after executing the instrumented application, we collect the set of queries reported in Table 1.

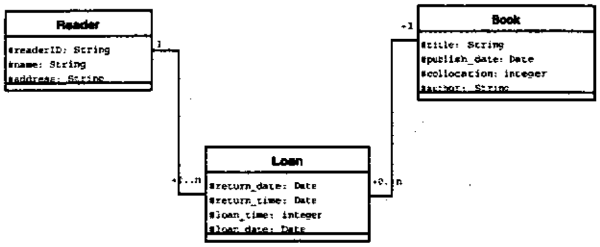

Figure 2. Working example E–R model

2.2 Identifying services using Formal Concept Analysis

Once queries have been extracted from database interaction, they are clustered using FCA. FCA is a mathematical tool that permits the identification of groups of objects sharing common attributes [4]. FCA can be thought of as the process of searching "rectangles" in a Boolean table representing a relation between *objects* and *attributes*. Thus, a concept is a maximal rectangle in a table where columns and rows permutations are allowed. More precisely [9], FCA starts with a *context*, a triple, $C = (O, A, R)$, where O is a finite set of *objects*, A is a finite set of *attributes*, and $P \subseteq O \times A$ is a relation between O and A. If the pair $(o, a) \in P$, it can be said that *object o* has *attribute a*. Given a set of objects $X \subseteq O$,

$$\sigma(X) := \{a \in A \mid \forall o \in X : (o, a) \in P\}$$

is the set of *common attributes* while, given $Y \subseteq A$,

$$\tau(Y) := \{o \in O \mid \forall a \in Y : (o, a) \in P\}$$

is the set of *common objects*.

A *concept* is a pair of sets (X, Y) where $X \subseteq O$ is called the *extent*, $Y \subseteq A$ is called the *intent*, and $Y = \sigma(X)$, $X = \tau(Y)$. That is, a *concept* is a maximal collection of *objects* sharing common *attributes*. The set of all *concepts* is denoted by $B(O, A, R)$. Furthermore, a *concept* (X_1, Y_1) is a subconcept of another *concept* (X_2, Y_2) if $X_1 \subseteq X_2$. This imposes a partial order relation on $B(O, A, R)$, and it can be written that $(X_1, Y_1) \preceq (X_2, Y_2)$. The partial order $\preceq$, can be used to build a lattice called *concept lattice*, where each node represents a *concept*. The *concept lattice* introduces a hierarchical clustering of *objects* and *attributes*, where upper concepts factor out common *attributes*, while lower concepts factor out common *objects*. More details can be found in the work of Ganter and Wille [9].

SQL queries extracted by dynamic analysis constitute the starting point for identifying features. Simple SQL queries are characterized by i) the information they recover,

Table 1. Queries from the working example

q0	SELECT author FROM book WHERE title='The mistery play' AND available=true
q1	SELECT collocation, loan_date, name FROM loan, book, reader WHERE book.title='The shooter' AND loan_date>='8-09-05' AND loan_time>='16:00'AND reader_ID=74 and reader.readerID=loan.readerID and book.title=loan.title
q2	SELECT author FROM book WHERE title='Blue ice' AND available=true
q3	SELECT name,address FROM reader, loan WHERE readerID=43 AND return_date >='13-11-06' and reader.readerID=loan.readerID
q4	SELECT author FROM book WHERE title='Da Vinci' AND available=true
q5	SELECT name,address FROM reader, loan WHERE readerID=534 AND return_date <='03-10-06' and reader.readerID=loan.readerID
q6	SELECT collocation, loan_date, name FROM loan, book, reader WHERE book.title='Jess in action' AND loan_date>='3-09-06' AND loan_time>='9:00'AND reader.readerID='56' and reader.readerID=loan.readerID and book.title=loan.title
q7	SELECT name, address FROM reader, loan WHERE reader.readerID=909 AND return_date<='24-11-06' and reader.readerID=loan.readerID

in terms of fields (i.e., the SELECT and the FROM clauses) and ii) how such an information extraction is constrained (i.e., the WHERE clause). The proposed approach first performs a separate analysis of fields and constraints. To make the use of fields referred in SQL queries possible, we need to full qualify them, i.e., to specify the table for each field. This is done by matching field names and tables accessed by the queries (i.e., the FROM clause) with the database structure as dumped from the DBMS.

After fields have been qualified, the FCA context is built. The context considers queries as *objects*, while fields selected from the database tables (i.e., the list of fields listed in the SELECT clause) plus the conjuncts of query conditions (extracted from the WHERE clause) as *attributes*. Conditions are transformed by replacing literals with symbols. For example, given the query *select author from book where title="Da Vinci Code"*, we extracted the attributes *book.author* (a field) and *book.title_EQ_VALUE*. Table 2 shows an example of a context extracted from simple queries (due to the lack of space, the context is represented as a list of attributes for each object).

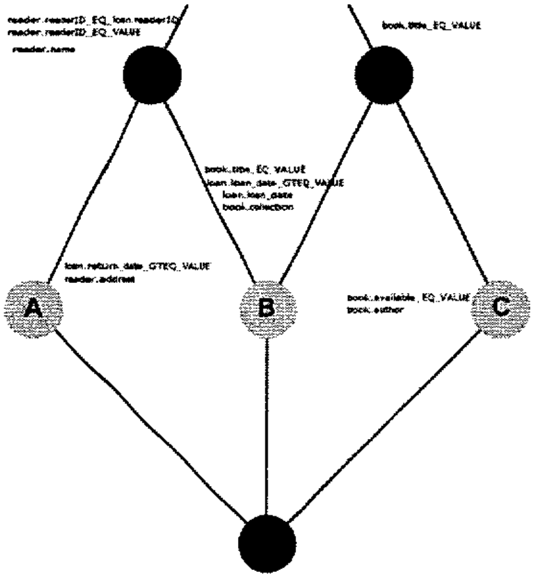

Figure 3. Working example concept lattice

Figure 3 shows the concept lattice obtained from the context of Table 2. Concepts A, B and C cluster queries sharing fields and conditions shown on nodes and on upper concept nodes. In particular, by looking at the three concepts, it can be noted that:

1. *Node A* clusters queries retrieving name and address of readers holding books where the loaning expiration date is above a given one. This suggests the creation of a service with an operation *getExpiredLoans(d:Date):AuthorData*, where *AuthorData* is a complex type composed of author name and address.

2. *Node B* clusters queries retrieving loans (i.e., loan date and book information) for a given book made after a given date. This suggests the creation of a service with an operation *getLoans(bookTitle: string, d:date):loanInfo* where *loanInfo* is a complex type composed of *loan_date* and *book.collection*.

3. *Node C* clusters queries searching for books where the title matches a given string and the availability is true. This suggests the creation of a service with an operation *checkBookAvailability(title:String):boolean*.

On larger case study, nodes clustering a number of queries below a given threshold – two in our case studies, although smaller thresholds would increase the approach's recall, while higher thresholds would increase the precision – will not be considered as candidate services. Although this could potentially exclude some good candidates, it also excludes database accesses that, during the applications' usage, are exercised only in a few cases.

2.3 Approach limitations

The current version of the approach only focuses on SELECT operations, whilst a similar strategy can easily applied on other operations as well. Also, nested queries are not supported, and the approach currently handles single query transactions, while a completely different approach is under development for transactions involving multiple SQL operations. For this reason the approach could also recover

Table 2. Working example context

q0	book.author book.title_EQ_VALUE book.available_EQ_VALUE
q1	book.collocation loan.loan_date reader.name book.title_EQ_VALUE loan.loan_date_GTEQ_VALUE reader.readerID_EQ_VALUE reader.readerID_EQ_loan.readerID book.title_EQ_loan.title
q2	book.author book.title_EQ_VALUE book.available_EQ_VALUE
q3	reader.name reader.address reader.readerID_EQ_VALUE loan.return_date_GTEQ_VALUE reader.readerID_EQ_loan.readerID
q4	book.author book.title_EQ_VALUE book.available_EQ_VALUE
q5	reader.name reader.address reader.readerID_EQ_VALUE loan.return_date_GTEQ_VALUE reader.readerID_EQ_loan.readerID
q6	book.collocation loan.loan_date reader.name book.title_EQ_VALUE loan.loan_date_GTEQ_VALUE reader.readerID_EQ_VALUE reader.readerID_EQ_loan.readerID book.title_EQ_loan.title
q7	reader.name reader.address reader.readerID_EQ_VALUE loan.return_date_GTEQ_VALUE reader.readerID_EQ_loan.readerID

features that are part of a more complex service, rather than the service as a whole.

Finally, scalability due to the use of FCA could potentially constitute an issue. However, the proposed approach mainly relies on query sets clustered by lattice nodes, rather than on the visualization of lattices, that is not necessary. Thus, provided that the FCA tool can build lattices over the query sets object of our study, the approach is able to scale up.

2.4 Tool Support

The source code instrumentor and the SQL query analyzer have been developed in Java on top of the JavaCC[1] parser generator. The FCA has been applied using the *Concepts* tool[2].

3 Empirical Study

This section reports the case studies aiming to assess the proposed approach for service mining in database applications. First, the section presents the empirical study definition and context, and defines the variables to be measured. Then, results are reported and discussed.

3.1 Empirical study definition and context

The goal of this study is to assess the effectiveness of the service mining techniques. The quality focus is the precision of the approach – i.e., the ability to filter out as many false positive services as possible – and its recall, i.e., the ability to recover the largest possible number of good candidate services. The study is carried out in the context of two Java database applications, JBooks and E–PaperML. JBooks [1] is a Java–based personal finance application. The application has a checkbox–based interface that allows the user to insert and visualize transactions, is interfaced with a relational database, and involves a double–entry system for all transactions, i.e., every transaction involves a transfer from one account to another. E–PaperML is a Java–based application intended to create and manipulate electronic newspapers. E–PaperML uses a SQLServer 2000 database to store newspapers and articles. The main features of this application consist in creating editions, creating and modifying news, manage the editorial board, etc. Table 3 reports characteristics of the two applications in terms of LOC, the number of classes and the number of the methods to accessing the database. Execution traces have been extracted by exercising the two applications over the following scenarios extracted from user manuals.

Table 3. Main characteristics of the case study applications

Case study	LOC	# of Classes	# of Method Accesses
E_PaperML	12655	84	4
JBooks	20108	20	7

- **E-PaperML**
  1. User authentication;
  2. Listing of previous editions;
  3. Visualization of an edition in the main screen;
  4. Addition of sections to the loaded edition;
  5. Removal of sections from the loaded edition;
  6. Loading of news, or their creation in the new edition;
  7. Removal of news from the current edition;
  8. Loading and change of the staff.
- **JBooks**
  1. Visualization of account information;

[1] *https://javacc.dev.java.net/*

[2] *http://www.st.cs.uni-sb.de/ lindig/src/concepts.html*

2. Choice of account type from a list;
3. Listing of transaction information;
4. Creation of a new account;
5. Creation of a new transaction;
6. Creation of a new account type;
7. Visualization of the amount moved between two accounts within a transaction.

3.2 Research Questions and Variables

The research questions this study aims to answer are:

1. What percentage of the retrieved query clusters are good candidates for possible services?
2. Are there services in the analyzed application that the approach was not able to identify?

To this aim, the performance of the approach is measured in terms of *precision* and *recall*. Given N the number of retrieved services, U the number of useful, retrieved services, T the number of services that could have been identified in the system, precision and recall are defined as:

$$Precision = \frac{U}{N} \cdot 100$$
$$Recall = \frac{U}{T} \cdot 100$$

The variable U is measured by inspecting the obtained query concepts, while T is the set of possible services as suggested by the software system developers (only available for E–PaperML).

3.3 Empirical Study Results

The analysis of execution traces permitted the extraction of 80 SELECT queries for E–PaperML, and 221 for JBooks. After queries have been processed – qualifying fields and replacing literals by symbols – the FCA was applied to cluster them. Results of the service identification are shown in Section 3.3.1 and Section 3.3.2.

3.3.1 E-PaperML

Figure 4 shows the concept lattice clustering together E–PaperML queries. The lattice contains 25 concepts, 15 of which -colored in grey- constitute candidate services. Each candidate service is described in Table 4, reporting for each service, (1) the input/output parameters, and (2) the service name and the FCA attributes (fields, conditions) characterizing the feature. Return parameter consisting of collections of primitive or complex types are named with the *Collection* suffix. Let us now discuss in detail a few examples of candidate services found:

- *Node* $qs2$ clusters together queries retrieving the IDs of news related to a given one. This suggests the creation of a service with an operation *getRelatedNews(newsID:int):Related_NewsCollection*, where *Related_NewsCollection* is a collection of news.
- *Node* $qs4$ clusters together queries retrieving first and last name of a journalist privided her/his ID. This suggests the creation of a service with an operation *getJournalistInformation(dni:String):JournalistData*, where *JournalistData* is a complex type composed of first and last name.
- *Node* $qs11$ clusters together queries retrieving the IDs of news, for a particular edition, appearing in the newspaper main page. This suggests the creation of a service with an operation *getNewsInMainPage(id_Edition:int):idNewsCollection.*
- *Node* $qs15$ clusters together queries retrieving name and description of all sections that can be created for an edition. This suggests the creation of a service with an operation *getSectionInformation(void):SectionDataCollection*, where *SectionDataCollection* is a collection of complex types composed of name and description for each section.
- *Node* $qs20$ clusters together queries retrieving title, subtitle, summary, body, section, date and ID for news beloging to a given section of an edition. This suggests the creation of a service with an operation like *getNewsBySectionAndEdition(edition:int, section:String):NewsDataCollection*, where *NewsDataCollection* is a collection of complex types composed of title, subtitle, summary, body, section, date and ID of each news.

3.3.2 JBooks

Figure 5 shows the concept lattice for the JBooks case study. The lattice contains 22 concepts, of which 11 – colored in grey – represent candidate services. Details about each candidate service are reported in Table 5.

In particular, by looking at the concepts, it can be noted that:

1. *Node* $qs1$ clusters together queries searching for the amount of the money of the last transaction made. This suggests the creation of a service with an operation *getLastTransactionValue():double*.
2. *Node* $qs3$ clusters together queries retrieving the total values transferred during transactions made before a given date and from a given account. This suggests the creation of a service with an operation

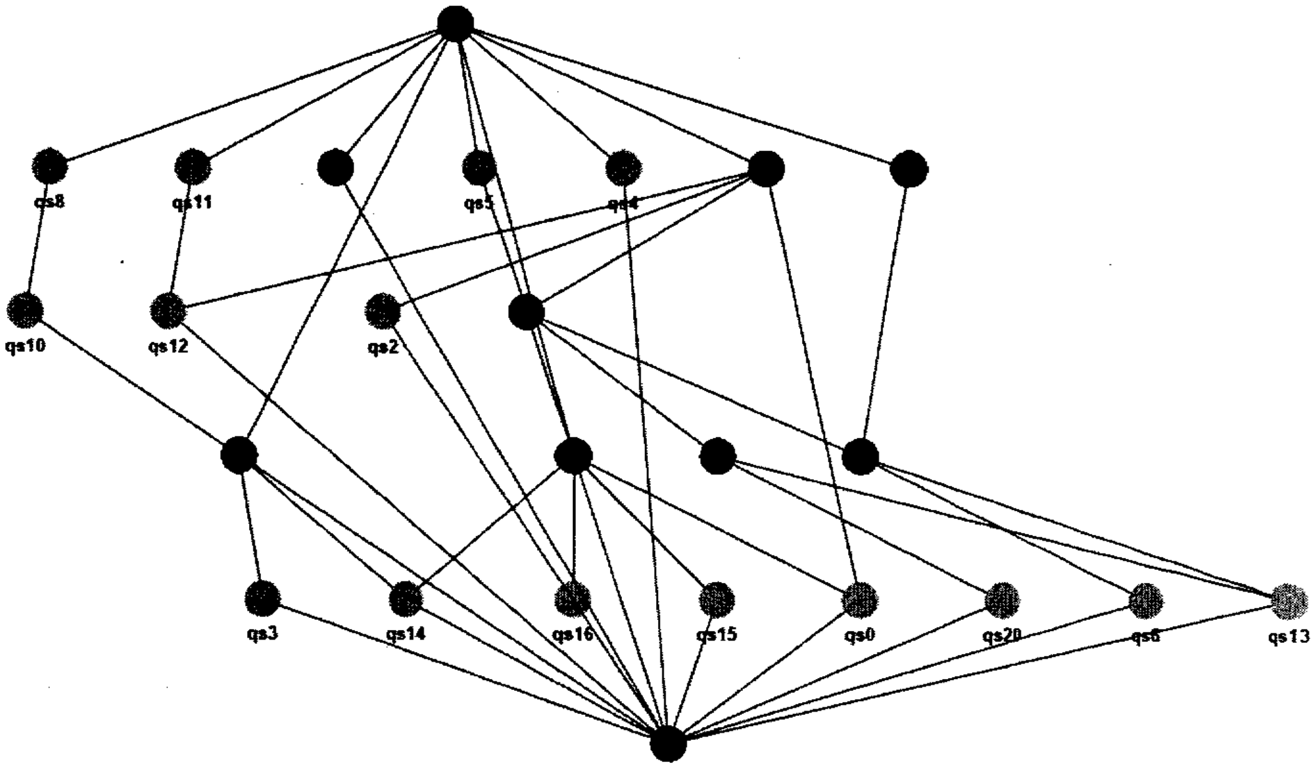

Figure 4. E-PaperML: FCA lattice

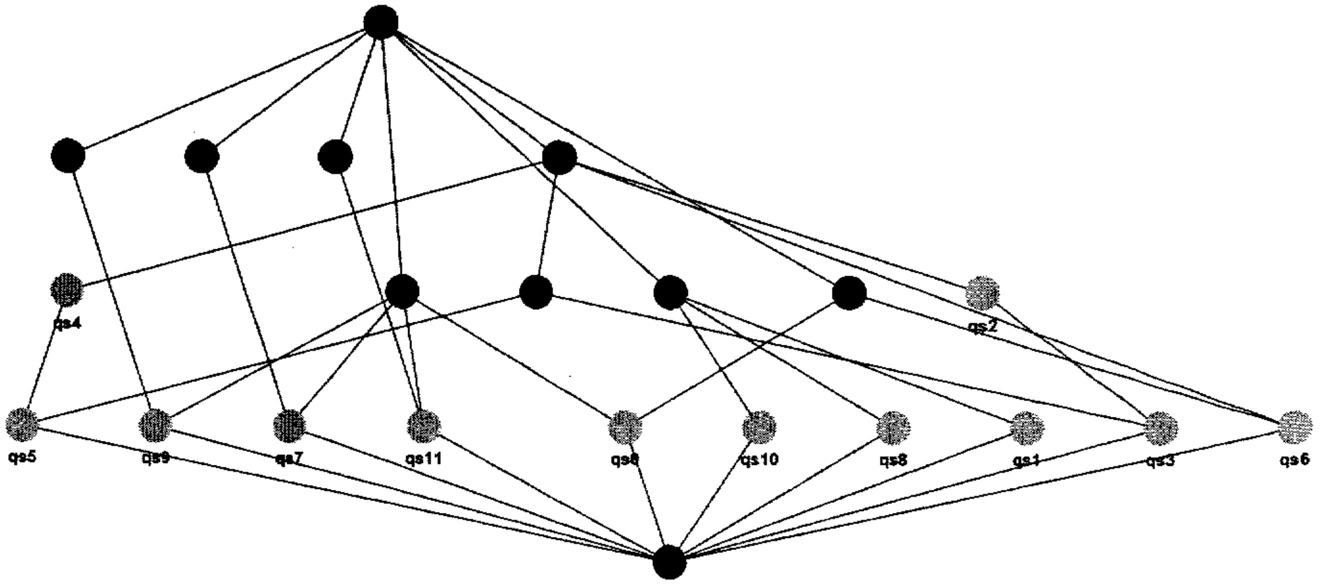

Figure 5. JBooks: FCA lattice

Table 4. E-PaperML candidate services

Concept ID	Input Params	Return Parameter	Service	Classifier Attributes
qs0	-	IdNewsCollection	getListOfNewsID	noticia.id_noticia empty
qs2	reference_News:int	Related_NewsCollection	getRelatedNews	not_relacionada.id_not_ppal_EQ_value not_relacionada.id_not_rel noticia.id_noticia_EQ_not_relacionada.id_not_rel
qs3	id_writer:String	AuthorName:String	getExternalWriterName	autor.id_EQ_otro_autor.dni autor.id_EQ_otro_autor.dni otro_autor.dni_EQ_value
qs4	dni:String	JournalistData	getJournalistInformation	autor.id_EQ_periodista.dni periodista.dni_EQ_value periodista.apellido2 periodista.apellido1 periodista.nombre
qs5	cif:String	PressAgencyName:String	getAgencyName	agencia_prensa.nombre autor.id_EQ_agencia_prensa.cif agencia_prensa.cif_EQ_value
qs6	-	-	**FALSE POSITIVE**	noticia.id_noticia_NOTEQ_value
qs8	role:String	StaffDataCollection	getPersonByRole	personal.cargo_EQ_value personal.dni personal.apellido2 personal.apellido1 personal.nombre
qs10	role:String edition:int	StaffDataCollection	getStaffInEdition	staff.id_edicion_EQ_value staff.dni_EQ_personal.dni
qs11	id_edition:int	IdNewsCollection	getNewsIdInMainPage	portada.id_edicion_EQ_value portada.id_noticia
qs12	id_edition:int	MainPageDataCollection	getOrderedNewsInMainPage	noticia.id_noticia_EQ_portada. id_noticia
qs13	id_edition:int	NewsDataCollection	getNewsFromEdition	edicion.id_EQ_noticia.id_edicion noticia.id_edicion_EQ_value
qs14	-	ExternalWriterDataCollection	getInformationOfExternalWriters	otro_autor.descripcion otro_autor.dni
qs15	-	SectionDataCollection	getSectionInformation	Seccion.descripcion Seccion.nombre
qs16	-	EditionIDCollection	getListOfEditionID	edicion.id
qs20	edition:int section:String	NewsDataCollection	getNewsBySectionAndEdition	noticia.seccion_EQ_value

getTotalMoney_Transfer_From_Account(date:Date, id_account:int, confirmed:int): MoneyAmountCollection.

3. *Node qs*5 clusters together queries retrieving the total values related to transactions made before a given date and to a given account. This suggests the creation of a service with an operation *getTotalMoney_Transfer_To_Account(date:Date, id_account:int, confirmed:int): MoneyAmountCollection.*

4. *Node qs*6 clusters together queries searching for the information related to transactions performed in a given date, from a specified account and to a specified account. This suggests the creation of a service with an operation *getTransactionInformation (date:Date, id_account_from:int, id_account_to:int): TransactionDataCollection* where *TransactionDataCollection* is a collection of complex types, each one containing information about the transaction, such as ID, category, source and target account, date, value, description, etc.

3.3.3 Precision and recall

Table 6 shows the precision for both case studies, and the recall for the E–PaperML. For JBooks, no false positives (100% precision) were found since, according to our knowledge of the application, no candidate service appeared to be meaningless nor useless. For E–PaperML we obtained a precision of 93%, i.e., one false positive, *qs6*. According to the developers, this query cluster cannot be brought back to a relevant feature. The *recall* for E–PaperML is 82%, thus also indicating that the proposed approach, other than being highly precise, was also able to achieve, at least for E–PaperML, a good recall. The developers suggested us that the following services, not identified by the proposed approach, would have been useful as well:

- *Obtain information from an edition:* this feature aims to retrieve information from an edition such as: the title of a newspaper, its publication date, its issue. The feature could be encapsulated in an operation *getEditionInformation(editionID:int):EditionData,* where *EditionData* is a complex type composed of the title of the newspaper, publication date and issue.

Table 5. JBooks candidate services

Concept ID	Input Parameters	Return Parameter	Service	Classifier Attributes
qs0	id:int	Amount:double	getInformationTransaction	transactions.id
qs1	-	Amount:double	getLastValueTransaction	transaction_id_seq.last_value
qs2	date:Date id_account:int	MoneyAmountCollection	getTotalMoney_Transfer_From_Account	account_from_EQ_VALUE
qs3	date:Date id_account:int confirmed:int	MoneyAmountCollection	getTotalMoney_Transfer_From_Account	confirmed_EQ_VALUE account_from_EQ_VALUE
qs4	date:Date id_account:int	MoneyAmountCollection	getTotalMoney_Transfer_to_Account	account_to_EQ_VALUE
qs5	date:Date id_account:int confirmed:int	MoneyAmountCollection	getTotalMoney_Transfer_To_Account	confirmed_EQ_VALUE account_to_EQ_VALUE
qs6	data:Date id_account_from:int id_account_to:int	TransactionDataCollection	getTransactionInformation	tran_date_GTEQ_VALUE transactions.description transactions.posted transactions.confirmed transactions.tax transactions.tran_date transactions.numberN transactions.account_to transactions.account_from transactions.category_id transactions.external_id
qs7	id_account:int	AccountDataCollection	getInformationAccount	accounts.id id_EQ_VALUE
qs8	-	MoneyValue:double	getLastAccountOfTransaction	account_id_seq.last_value
qs9	id:int	CategoryTransactionDataCollection	getInformationCategoriesTransaction	id_EQ_VALUE transaction_categories.id
qs10	-	MoneyValue:double	getLastValueCategoryTransaction	transaction_category_id_seq.last_value
qs11	id:int	TypeAccount:String	getIdAccountType	account_types.id id_EQ_VALUE

Table 6. Approach precision and recall for the two case studies

	EPaperML	JBooks
Precision	93%	100%
Recall	82%	N/A

- *Select sections in an edition:* this feature aims to retrieve a list of sections from an edition. The feature could be encapsulated in an operation like *getSectionListInEdition(editionID:int):SectionList*, where *SectionList* is a complex type composed of a list of strings representing newspaper sections.

- *Select identifier from the last inserted news:* this feature aims to retrieve the ID of the last inserted news. The feature could be encapsulated in an operation like *getLastInsertedNews(void):int.*

The main reason why the aforementioned features were not recovered was because either the features were not exercised when executing the application (i.e., the selected scenarios did not account for these features) or, in a few cases, because the features resulted in a number of queries below the selected threshold (two queries). In particular, for E–PaperML, 9 nodes clustering a single query were pruned out, one of these related to the missed service *getEditionInformation.* Clearly, lowering the threshold would have permitted the identification of such a service, while descreasing the precision, and thus increasing the amount of noise whoever is using the approach has to look into.

3.4 Threats to validity

Results presented in this paper might be affected from several threats to validity, that need to be discussed. Threats to *internal validity* are mainly due to how precision and recall were measured. For E–PaperML, precision and recall were computed with respect to indications gave us by system experts not aware of our service mining approach. They were able to indicate whether a service we extracted was or not a us a useful service, and to indicate us services that were not identified by the approach. Another, probably better, strategy would have been to ask them to identify a list of services and then to match this list against the list of candidate services we extracted. Such a study will be part of our future work. For JBooks the analysis was performed by one of the authors not involved in the development of the service mining tool. Threats to *external validity* can be due to the generalizability of results obtained by analyzing only two systems. E–PaperML and JBooks represent medium–size systems related to the publishing and financial domains. To further generalize the obtained results, further studies on larger software systems are necessary.

4 Related Work

The work described in this paper can be related to approaches for aspect mining, as well as to approaches for software system componentization and for database reengineering.

Commonalities can be found between the proposed approach and the aspect mining approach proposed by Tonella and Ceccato [22]. They build a concept lattice from execution traces extracting by exercising the application use cases, considering a context where the objects are the use cases and the attributes are methods exercised by each use case. Similarly, we extract queries by exercising the software system scenarios. However, our lattice is built upon fields and conditions extracted from SQL queries, rather than from exercised methods, since the focus in to cluster together related database interactions. Prieto *et al.* [13] propose a FCA–based approach to support the conceptual abstraction phase in a database reengineering process towards and object–oriented schema. In particular, Prieto *et al.* build a context where objects are tables and attributes are the fields. The obtained lattice highlights possible classes (as concepts) and hierarchies. Differently from them, we extract the lattice from queries rather than from the database structure, and consider conditions other than fields.

Other approaches for identifying potential services into existing software systems have been proposed by several authors. For example, Li and Tahvildari [16] presented a reuse–driven framework supporting the semi–automatic extraction of reusable components from an existing Java application, with the aim of transforming the existing system into a component–based system. Yang *et al.* [24] proposed an approach to transform pieces of functionality of a legacy system into web services. The approach uses business rules to extract services from specific Customer Relationship Management (CRM) components. Sneed [18] presented an approach for source code transformation with the aim of migrating towards a service–oriented architecture. This alternative is recommended when the component identified within source code are sufficiently decoupled from the environment, and the costs of reimplementation is too high. In the CELLEST project Stroulia *et al.* [20, 19, 21] proposed an approach to identify services in legacy application. The approach is based on the analysis of interactions between users and the legacy application, modeling the behavior of the legacy user interface as a state transition model. We share with them the idea of using information extracted from the user interaction with the application to detect candidate services.

The present work also related to previous work on reengineering of database applications. Database reverse engineering can be performed for the following purposes: redocumentation, model migration, restructuring, maintenance or improvement, tentative requirements, software assessment, integration, and conversion of legacy data [5, 8, 12]. Migrating database application towards SOA is a form of wrapping database access features. Wrappers make possible to transform queries from a data model to another [14]. Wrappers are used not only to adapt one data model to another, but also for other purposes, e.g., to adapt a relational database to a distributed environment [6]. Garcia *et al.* [11] describe a whole reengineering process for database applications, implemented using RelationalWeb. This tool takes a relational database as input and generates a full operational application to manage it. Also, they propose an approach, based on pattern–matching, to infer services from database applications [10]. Our approach differs from their one, in that it proposes a dynamic approach instead of a static one, and uses a FCA–based approach, inspired to a similar aspect–mining one [22], to cluster related queries.

5 Conclusions and Work–in–progress

Mining services from existing applications constitutes a valuable way to deal with the lack of available services, that is one of the factors limiting the development of service–oriented systems and the use of powerful features developed for them.

This paper proposed an approach to mine possible features that can be exported as services in database–oriented applications. The approach is based on Formal Concept Analysis (FCA), which clusters similar queries, dynamically extracted by observing the interaction between an application and a Database Management System. More exactly, FCA clusters queries retrieving the same fields using similar selection expressions. The approach has been applied on two Java software systems, E–PaperML and JBooks. Results showed that the approach is highly precise, i.e., over 90% of the obtained services are services that either make sense or, whenever possible – like in the case of E–PaperML – that the developer would have created as well. Also, the approach recall, where available (again E–PaperML) is high – around 80%.

Work–in–progress deals with more complex database applications, in particular aiming to mine services from database transactions, and also dealing with more complex queries, e.g., nested queries. Also, we plan to evaluate the approach on further, larger case studies from different domains.

References

[1] JBooks: a Java-based personal finance application. http://freshmeat.net/projects/jbooks/.

[2] G. Antoniol and M. Di Penta. Library miniaturization using static and dynamic information. In *Proceedings of IEEE International Conference on Software Maintenance*, pages 235–244, Amsterdam, The Netherlands, Sep 22-26 2003. IEEE Press.

[3] L. Baresi, C. Ghezzi, and S. Guinea. Smart Monitors for Composed Services. In *Proc. 2nd International Conference on Service Oriented Computing (ICSOC'04)*, pages 193–202, New York, USA, November 2004. ACM.

[4] G. Birkoff. *Lattice Theory*. American Mathematical Society, Providence R.I., 1940.

[5] M. Blaha. Dimensions of database reverse engineering. In *Proceedings of the Fourth Working Conference on Reverse Engineering (WCRE '97)*, pages 176–185. IEEE Computer Society, October 1997.

[6] Y. Bychkov and J. Jahnke. Wrapper development for legacy data reuse. In *Proceedings of the Eighth Working Conference on Reverse Engineering (WCRE'01)*, pages 328–334. IEEE Computer Society, 2001.

[7] G. Canfora, A. R. Fasolino, G. Frattolillo, and P. Tramontana. Migrating interactive legacy systems to web services. In *CSMR '06: Proceedings of the Conference on Software Maintenance and Reengineering*, pages 24–36. IEEE Computer Society, 2006.

[8] Y. Cohen and Y. Feldman. Automatic high-quality reengineering of database programs by abstraction, transformation and reimplementation. *ACM Transactions on Software Engineering and Methodology*, 12(3):285–316, 2003.

[9] B. Ganter and R. Wille. *Formal Concept Analysis*. Mathematical Foundations, Springer Verlag, 1999.

[10] I. Garcia-Rodriguez De Guzman, M. Polo, and M. Piattini. Obtaining web services from relational databases. In *Proceedings of the European Conference on Software Maintenance and Reengineering (CSMR 2006)*, volume 0, pages 304–310, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[11] I. Garcia-Rodrguez de Guzman, M. Polo, and M. Piattini. An integrated environment for reengineering. In *Proceedings of the 21st International Conference on Software Maintenance (ICSM 2005)*, pages 165–174. IEEE Computer Society, 2005.

[12] J. Henrard and J.-L. Hainaut. Data dependency elicitation in database reverse engineering. In *Proceedings of the Fifth European Conference on Software Maintenance and Reengineering (CSMR01)*, pages 11–19. IEEE Computer Society, 2001.

[13] C. Hernández, F. Prieto, M. A. Laguna, and Y. Crespo. Formal concept analysis support for conceptual abstraction in database reengineering. In *Database Maintenance and Reengineering Workshop (DBMR 2002)*, October 2002.

[14] J. Hick and J. Hainaut. Wrapper development for legacy data reuse. In *Proceedings of the Eighth Working Conference on Reverse Engineering (WCRE'01)*, pages 291, 306. IEEE Computer Society, 2001.

[15] G. Lewis, E. Morris, and D. Smith. Analyzing the reuse potential of migrating legacy components to a service-oriented architecture. In *CSMR '06: Proceedings of the European Conference on Software Maintenance and Reengineering*, pages 15–23. IEEE Computer Society, 2006.

[16] S. Li and L. Tahvildari. JComp: A reuse-driven componentization framework for Java applications. In *14th IEEE International Conference on Program Comprehension (ICPC'06)*, pages 264–267, 2006.

[17] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara. Semantic matching of web services capabilities. In *Proceedings of the first International Semantic Web Conference (ISWC 2002)*, volume 2348 of *Lecture Notes on Computer Science*, pages 333–347. Springer-Verlag, June 2002.

[18] H. M. Sneed. Integrating legacy software into a service oriented architecture. In *Proceedings of the Conference on Software Maintenance and Reengineering (CSMR06)*, 2006.

[19] E. Stroulia, M. El-Ramly, P. Iglinski, and P. Sorenson. User interface reverse engineering in support of interface migration to the web. *Automated Software Engg.*, 10(3):271–301, 2003.

[20] E. Stroulia, M. El-Ramly, L. Kong, P. Sorenson, and B. Matichuk. Reverse engineering legacy interfaces: An interaction-driven approach. In *WCRE '99: Proceedings of the Sixth Working Conference on Reverse Engineering*, page 292, Washington, DC, USA, 1999. IEEE Computer Society.

[21] E. Stroulia, M. El-Ramly, and P. Sorenson. From legacy to web through interaction modeling. In *ICSM '02: Proceedings of the International Conference on Software Maintenance (ICSM'02)*, page 320, Washington, DC, USA, 2002. IEEE Computer Society.

[22] P. Tonella and M. Ceccato. Aspect mining through the formal concept analysis of execution traces. In *Proceedings of the 11th Working Conference on Reverse Engineering (WCRE04)*. IEEE Computer Society, 2004.

[23] M. Turner, D. Budgen, and P. Brereton. Turning software into a service. *IEEE Computer*, 36(10):38–44, 2003.

[24] Y. Xu, Q. Duan, and H. Yang. Web-service-oriented customer relationship management system evolution. In *Proceedings of the 13th IEEE International Workshop on Software Technology and Engineering Practice (STEP'05)*, 2005.

[25] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. QoS-aware middleware for web services composition. *IEEE Transactions on Software Engineering*, 30(5):311–327, May 2004.