

Boston, Massachusetts
July 9-11, 2007

2007



PROCEEDINGS

SEKE 2007

The 19th International Conference on Software Engineering & Knowledge Engineering

Sponsored by

Knowledge Systems Institute Graduate School, USA

Technical Program

July 9-11, 2007

Hyatt Harborside Hotel, Boston, Massachusetts, USA

Organized by

Knowledge Systems Institute Graduate School

Copyright © 2007 by Knowledge Systems Institute Graduate School

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the publisher.

ISBN 1-891706-20-9 (paper)

Additional Copies can be ordered from:
Knowledge Systems Institute Graduate School
3420 Main Street
Skokie, IL 60076, USA
Tel:+1-847-679-3135
Fax:+1-847-679-3166
Email:office@ksi.edu
<http://www.ksi.edu>

Proceedings preparation, editing and printing are sponsored by
Knowledge Systems Institute Graduate School

Printed by Knowledge Systems Institute Graduate School

Table of Contents

Foreword	iii
Conference Organization	iv
Keynote	1
<i>Dr. Joe Urban</i>	
Software Engineering Methodology	
Constructing Self-Adaptive Systems with Polymorphic Software Architecture <i>Xiaoxing Ma, Yu Zhou, Jian Pan, Ping Yu, Jian Lu</i>	2
Odyssey-MDA: A transformational approach to component models <i>Natanael E. N. Maia, Ana Paula Terra Bacelo, Claudia M. Werner</i>	9
An Empirical Expository Study on Inferring Developers' Activities from Low-Level Data <i>Irina Coman, Alberto Sillitti (S)</i>	15
TRAP.NET: A Realization of Transparent Shaping in .NET <i>S. Masoud Sadjadi, Fernando Trigos</i>	19
A Framework for Selecting Agile Practices and Defining Agile Software Processes <i>Patricia Vilain, Priscila Basto Fagundes, Thiago Leao Machado (S)</i>	25
A Proposal to Delegate GUI Implementation using a Source Code based Model <i>Marco Monteiro, Paula Oliveira, Ramiro Goncalves (S)</i>	29
Cost-based Analysis of Multiple Counter-Examples <i>Flavian Vasile, Samik Basu</i>	33
Common Coupling as a Measure of Reuse Effort in Kernel-Based Software <i>Liguo Yu, Stephen R. Schach, Kai Chen</i>	39
An Approach to Validating Translation Correctness From SAM to Java <i>Yujian Fu, Zhijiang Dong, Gonzalo Argote-Garcia, Leyuan Shi, Xudong He (S)</i>	45

Table of Contents

Foreword	iii
Conference Organization	iv
Keynote	1
<i>Dr. Joe Urban</i>	
Software Engineering Methodology	
Constructing Self-Adaptive Systems with Polymorphic Software Architecture <i>Xiaoxing Ma, Yu Zhou, Jian Pan, Ping Yu, Jian Lu</i>	2
Odyssey-MDA: A transformational approach to component models <i>Natanael E. N. Maia, Ana Paula Terra Bacelo, Claudia M. Werner</i>	9
An Empirical Expository Study on Inferring Developers' Activities from Low-Level Data <i>Irina Coman, Alberto Sillitti (S)</i>	15
TRAP.NET: A Realization of Transparent Shaping in .NET <i>S. Masoud Sadjadi, Fernando Trigos</i>	19
A Framework for Selecting Agile Practices and Defining Agile Software Processes <i>Patricia Vilain, Priscila Basto Fagundes, Thiago Leao Machado (S)</i>	25
A Proposal to Delegate GUI Implementation using a Source Code based Model <i>Marco Monteiro, Paula Oliveira, Ramiro Goncalves (S)</i>	29
Cost-based Analysis of Multiple Counter-Examples <i>Flavian Vasile, Samik Basu</i>	33
Common Coupling as a Measure of Reuse Effort in Kernel-Based Software <i>Liguo Yu, Stephen R. Schach, Kai Chen</i>	39
An Approach to Validating Translation Correctness From SAM to Java <i>Yujian Fu, Zhijiang Dong, Gonzalo Argote-Garcia, Leyuan Shi, Xudong He (S)</i>	45

Software Processes and Engineering Practice

QSEE Project: An Experience in Outsourcing Software Development for Space Applications <i>Valdivino Santiago, Fatima Mattiello-Francisco, Ricardo Costa, Wendell Pereira da Silva, Ana Maria Ambrosio</i>	51
Broadening the Use of Process Patterns for Modeling Processes <i>Hanh Nhi Tran, Bernard Coulette, Bich Thuy Dong</i>	57
A Framework for Tailoring Software Process <i>Lisandra M. Fontoura, Roberto T. Price (S)</i>	63
Analyzing Configuration Management Repository Data for Software Process Improvement <i>Shihong Huang, Christopher Lo (S)</i>	67

Aspect-Based Software Development

Smooth Quality Oriented Component Integration through Product Line Based Aspect-Oriented Component Adaptation <i>Yankui Feng, Xiaodong Liu, Jon Kerridge</i>	71
Modular Specification of Aspect-oriented Systems and Aspect Conflicts Detection <i>Hui Liang, Jing Sun (S)</i>	77
Avoiding Bad Smells in Aspect-Oriented Software <i>Eduardo K. Piveta, Marcelo Hecht, Ana Moreira, Marcelo S. Pimenta, Joao Araujo, Pedro Guerreiro, R. Tom Price</i>	81

Software Testing and Quality Assurance

Metrics of Credibility and Interaction Quality: Design and Evaluation <i>Nilda Perez Otero, Marcelo Perez Ibarra, Sandra Mendez, Adelina Garcia, Ma. del Pilar Galvez Diaz, Viviana E. Quincoces, Hector Liberatori, Beatriz Fiorito, Cecilia Maria Lasserre</i>	87
Predicting Order of Likelihood of Defective Software Modules <i>Rattikorn Hewett, Phongphun Kijsanayothin, Alta van der Merwe</i>	93
Automated Test Code Generation from UML Protocol State Machines <i>Dianxiang Xu, Weifeng Xu, W. Eric Wong</i>	99

Validating A Layered Decision Framework for Cost-Effective Network Defense <i>Huaqiang Wei, Jim Alves-Foss, Du Zhang</i>	105
Toward Modeling and Analysis for Software Installation Testing <i>Jerry Gao, Sujana Tirumalasetti, Chien-Pin Hsu, Yip Cheong, Anne Colendich, Todd Fitch (S)</i>	111
Automatic Test Generation for Database-Driven Applications <i>Zhenyu Dai, Mei-Hwa Chen</i>	117
Fault-Based Testing of Data Schemas <i>Maria Claudia F. P. Emer, Sílvia Regina Vergilio, Mario Jino</i>	123
NLForSpec: Translating Natural Language Descriptions into Formal Test Case Specifications <i>Daniel Leitao, Dante Torres, Flavia Barros</i>	129
Enhanced Random Testing for Programs with High Dimensional Input Domains <i>F.-C. Kuo, K.-Y. Sim, Chang-ai Sun, S.-F. Tang, Zhi Quan Zhou</i>	135
On Test Case Distributions of Adaptive Random Testing <i>Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu (S)</i>	141
Reducing the Number of Test Cases for Performance Evaluation of Components <i>Joco W. Cangussu, Kendra Cooper, Eric Wong</i>	145
Combining Decorated Classification Trees with RCPS Stochastic Models to Gain New Valuable Insights into Software Project Management <i>Antonio Juarez Alencar, Gelson Guedes Rodrigues, Eber Assis Schmitz, Armando Leite Ferreira</i>	151
Towards a Reference Architecture for Software Testing Tools <i>Elisa Yumi Nakagawa, Adenilso da Silva Simao, Fabiano Ferrari, Jose Carlos Maldonado</i>	157
Controlling Restricted Random Testing: An Examination of the Exclusion Ratio Parameter <i>Kwok Ping Chan, T.Y. Chen, Dave Towey (S)</i>	163
An Approach to Software Testing of Machine Learning Applications <i>Christian Murphy, Gail Kaiser, Marta Arias (S)</i>	167

System Requirements Analysis, Modeling and Specification

Agile Methods and Quality Models: Towards an Integration in Requirements Engineering	173
<i>Alexandre Lazaretti Zanatta, Patricia Vilain</i>	
A Framework of Hierarchical Requirements Patterns for Specifying Systems of Interconnected Simulink/Stateflow Modules	179
<i>Changyan Zhou, Ratnesh Kumar, Devesh Bhatt, Kirk Schloegel, Darren Cofer</i>	
In the Requirements Lies the Power	185
<i>Rand Waltzman, Kristina Winbladh, Thomas A. Alspaugh, Debra J. Richardson</i>	
Data and Process Analyses of Data Warehouse Requirements	191
<i>Estella Annoni, Franck Ravat, Olivier Teste</i>	
Requirement Analysis Evolution through Patterns	197
<i>Luca Vetti Tagliati, Roger Johnson, George Rousso (S)</i>	
Automatic Generation of Use Case Diagrams from English Specifications Document	203
<i>Nathalie Rose T. Lim, Christobal T. Cayaba, Joseph Astrophel E. Rodil</i>	
REM4j - A framework for measuring the reverse engineering capability of UML CASE tools	209
<i>Steven Kearney, James F. Power</i>	
FlexUML: A UML Profile for Flexible Process Modeling	215
<i>Ricardo Martinho, Dulce Domingos, Joao Varajao</i>	
A Formal Specification for Product Configuration in Software Product Lines	221
<i>Huilin Ye, Yuqing Lin</i>	
Designing a Platform-Independent Use-Case for a Composite Application using a Reference Architecture	227
<i>Helge Hofmeister, Guido Wirtz (S)</i>	
Synchronization of UML Based Refactoring with Graph Transformation	232
<i>Y. Kosker, A. B. Bener</i>	
Using Formal Composition of Use Cases in Requirements Engineering	238
<i>Rabeb Mizouni, Aziz Salah, Rachida Dssouli</i>	

Web-Based Applications

Real-Time Trust Management in Agent Based Online Auction Systems <i>Rinkesh Patel, Haiping Xu, Ankit Goel</i>	244
Geometric Thumbnails For Web Searching <i>Chris Dunn, Beomjin Kim (S)</i>	251

Web Technology and Web Engineering

ADKwik: Web 2.0 Collaboration System for Architectural Decision Engineering <i>Nelly Schuster, Olaf Zimmermann, Cesare Pautasso</i>	255
Improving Usability of Web Systems with Similar Business Objectives <i>Rashid Ahmad, Zhang Li, Farooque Azam</i>	261
Processing Manipulations of Context Information on the Web <i>Roberto De Virgilio</i>	268
A Tag-Level Web-Caching Scheme for Reducing Redundant Data Transfers <i>Steven E. Cox, Du Zhang, Jinsong Ouyang</i>	274

Software Project and Resource Management

Methodology for Reusing Human Resources Management Standards <i>Asuncion Gomez-Perez, Jaime Ramirez, Boris Villazon-Terrazas</i>	280
An Adaptive Resource Management Approach for a Healthcare System <i>Claudia Raibulet, Luigi Ubezio, Stefano Mussino</i>	286
Study of the Relationships between Personality, Satisfaction and Product Quality in Software Development Teams <i>Marta Gomez, Silvia T. Acuna (S)</i>	292

Software Reuse and Component Technology

Towards Constructing High-available Decentralized Systems via Self-adaptive Components <i>Xi Sun, Li Zhou, Lei Zhuang, Wenpin Jiao, Hong Mei</i>	296
---	-----

SAFES: A Static Analysis for Field Security in Java Components <i>Aiyu Shi, Gleb Naumovich</i>	302
Reuse of Database Access Layer Components in JEE Product Lines: Limitations and a Possible Solution (Case Study) <i>Ding Peng, Stan Jarzabek, Damith C. Rajapakse, Hongyu Zhang</i>	308
Design of Wrapper for Self-Management of COTS Components <i>Michael E. Shin, Fernando Paniagua</i>	314
QoS-Optimized Integration of Embedded Software Components with Multiple Modes of Execution <i>Zonghua Gu, Qingxu Deng</i>	320
A C++ Framework for Developing Component Based Software Supporting Dynamic Unanticipated Evolution <i>Andre Rodrigues, Hyggo Almeida, Angelo Perkusich</i>	326
Representing Design Rationale to support Reuse <i>Adriana Pereira de Medeiros, Daniel Schwabe (S)</i>	332
Software System Maintenance	
Telling Stories about System Use: Capturing Collective Tacit Knowledge for System Maintenance <i>Adriana Cristina de Oliveira, Renata Mendes de Araujo, Marcos R.S. Borges</i>	337
Evolution and Runtime Monitoring of Software Systems <i>Hui Liang, Jin Song Dong, Jing Sun</i>	343
On Modern Debugging For Rule-Based Systems <i>Valentin Zacharias, Andreas Abecker</i>	349
Truth Eliciting Mechanisms for Trouble Ticket Allocation in Software Maintenance Services <i>Karthik Subbian, Y. Narahari</i>	355
Knowledge Engineering, Natural Language Processing, and AI	
Graphical Notation for Natural Language and Knowledge Representation <i>Magda G. Ilieva</i>	361

A Hybrid Approach for Natural Language Query Translation <i>Pornpimon Teekayuphun, Ohm Sornil</i>	368
Effective Fault Localization using BP Neural Networks <i>W. Eric Wong, Lei Zhao, Yu Qi, Kai-Yuan Cai, Jing Dong</i>	374
Temporal Software Change Prediction Using Neural Networks <i>Mehdi Amoui, Mazeriar Salehie, Ladan Tahvildari</i>	380
Do Neural-Network Question-Answering Systems Have a Role to Play in the Deployment of Real World Information Systems? <i>Antonio Juarez Alencar, Renata Chaomey Wo, Eber Assis Schmitz, Armando Leite Ferreira</i>	386
Knowledge Conversion in Software Development <i>Olivier Gendreau, Pierre N. Robillard</i>	392
A Language Facilitating Informal Reasoning about Programs <i>J. Nelson Rushton, Dwayne Towell</i>	396

Plenary Talk

Towards Seamless Business Process and Dialogue Specification <i>Dr. Dirk Draheim</i>	402
---	-----

Database Retrieval Methods

Evaluating the Efficiency of Retrieval Methods for Component Repositories <i>Oliver Hummel, Werner Janjic, Colin Atkinson</i>	404
Benchmarking the RDF(S) Interoperability of Ontology Tools <i>Raul Garcia-Castro, Asuncion Gomez-Perez, York Sure</i>	410
A Deep Classification of Temporal Versioned Integrity Constraints for Designing Database Applications <i>Robson Leonardo Ferreira Cordeiro, Renata de Matos Galante, Nina Edelweiss, Clesio Saraiva dos Santos (S)</i>	416
Generating Linear Temporal Logic Formulas for Pattern-Based Specifications <i>Salamah Salamah, Vladik Kreinovich, Ann Q. Gates (S)</i>	422

Ontology Based Classification Generating Method for Browsing-Based Component Retrieval	428
<i>Ge Li, Lu Zhang, Bing Xie, Weizhong Shao (S)</i>	

Data

A Context-Dependent Semantic Distance Measure	432
<i>Ahmad El Sayed, Hakim Hacid, Djamel Zighed</i>	
A Semantical Change Detection Algorithm for XML	438
<i>Rodrigo Cordeiro dos Santos, Carmem Hara</i>	
XML Schema Evolution by Context Free Grammar Inference	444
<i>Julio C. T. da Silva, Martin A. Musicante, Aurora T. R. Pozo, Silvia R. Vergilio</i>	

Software Development and Design Pattern

Software Tradeoff Assistant: An Integrated Framework for Analytical Decision Making and Tradeoffs in Software Development	450
<i>Ratikorn Hewett, Vikram Patankar</i>	
Improving Separation of Concerns in the Development of Scientific Applications	456
<i>S. M. Sadjadi, J. Martinez, T. Soldo, L. Atencio, R. M. Badia, J. Ejarque (S)</i>	
Pattern-based J2EE Application Deployment with Cost Analysis	462
<i>Nyun Zhang, Gang Huang, Ling Lan, Hong Mei (S)</i>	
Exploratory Design of Derivation Business Rules Using Query Rewriting	467
<i>Roman Krenicky, David Willmor, Suzanne M. Embury (S)</i>	
Classification of Design Pattern Traits	473
<i>Jing Dong, Yajing Zhao (S)</i>	

Data Warehouse

A Proposal for a Conceptual Data Warehouse Quality Model	477
<i>Manuel Serrano, Rafael Romero, Jose-Norberto Mazon, Juan Trujillo, Mario Piattini</i>	
Integrating Complex Data into a Data Warehouse	483
<i>F. Rava, O. Teste, R. Tournier, G. Zurfluh (S)</i>	

Data Mining and Machine Learning

- Learning from Software Quality Data with Class Imbalance and Noise
Andres Folleco, Taghi M. Khoshgoftaar, Jason Van Hulse, Chris Seiffert 487

System and Software Architecture

- Architectural Elements Recovery and Quality Evaluation to Assist in Reference Architectures Specification
Aline Pires Vieira de Vasconcelos, Claudia Maria Lima Werner 494
- EvoSpaces: 3D Visualization of Software Architecture
Sazzadul Alam, Philippe Dugerdil 500
- Ontobrowse: A Semantic Wiki for Sharing Knowledge about Software Architectures
Hans-Jorg Happel, Stefan Seedorf 506

Applications

- Building Business Considerations into Enterprise Application Designs
Rattikorn Hewett, Aashay Thipse 513
- Incremental effort prediction models in Agile Development using Radial Basis Functions
Raimund Moser, Witold Pedrycz, Giancarlo Succi (S) 519
- BASS: Business Application Support through Software Services
Mateus B. Costa, Rodolfo F. Resende, Marcelo V. Segatto, Eduardo F. Nakamura, Nahur Fonseca 523

Model-Driven Software Development

- Using Model-Driven Pattern Matching to derive functionalities in Models
Ignacio Garcia-Rodriguez de Guzman, Macario Polo, Mario Platini 529
- A Model-driven Approach to Architecting Secure Software
Ebenezer A. Oladimeji, Sam Supakkul, Lawrence Chung (S) 535

Agent-Based Technology and Intelligence

An Intelligent Agent of Automatically Notify Services <i>Shuo-Yan Hsu, William C. Chu</i>	541
A Proposal for a Decentralized Multi-Agent Architecture for Virtual Enterprises <i>Andreas Grunert, Sven Kaffille, Guido Wirtz</i>	546
Traceability for Agent-Oriented Design Models and Code <i>Gilberto Cysneiros, Andrea Zisman</i>	552
ONTOMADEM: An Ontology-driven Tool for Multi-Agent Domain Engineering <i>Rosario Girardi, Adriana Leite</i>	559
A Three Level Multi-agent Architecture to Foster Knowledge Exchange <i>Juan Pablo Soto, Aurora Vizzaino, Javier Portillo-Rodriguez, Mario Piattini (S)</i>	565
An Agent Based System for Search in Distributed Environments <i>Li Sa, Yong-Sheng Ding (S)</i>	570

DB Access and Query Processing

Tree Hash Under Concurrency Control <i>Kyosuke Yasuda, Takao Miura</i>	574
Query Processing in Paraconsistent Databases in the Presence of Integrity Constraints <i>Navin Viswanath, Rajshekhar Sunderraman</i>	580
An Object-Oriented Approach to Storage and Retrieval of RDF/XML Documents <i>Ching-Ming Chao</i>	586
CXPath: a Query Language for Conceptual Models of Integrated XML Data <i>Diego de Vargas Feijo, Claudio Naoto Fuzitaki, A Ivaro Moreira, Renata de Matos Galante, Carlos Alberto Heuser</i>	592

Service-Oriented Technology and Web Technology

OWLed: Extending Knowledge for Web Ontology Language <i>Hichem Zait, Aicha Mokhtari</i>	598
Using Ontologies to Represent Software Project Management Antipatterns <i>Dimitrios Settas, Ioannis Stamelos</i>	604

Service Composition Using Planning and Case-Based Reasoning <i>Kuan-Hsian Huang, Alan Liu</i>	610
MDA-based Ontology Development: A Study Case <i>Eluzai Souza dos Santos, Celia Ghedini Ralha, Hervaldo Sampaio Carvalho</i>	616
Towards Domain-Centric Ontology Development and Maintenance Frameworks <i>Faezeh Ensan, Weichang Du (S)</i>	622
Service Oriented Architecture Empirical Study <i>Mohammad Abu-Matar, Jeff Offutt (S)</i>	628
Semantic Support to Reformulate Public Services in Terms of Life Events <i>Luis Alvarez Sabucedo, Luis Anido Rifon (S)</i>	632
A Component-Based Solution and Architecture for Dynamic Service-Based Applications <i>Alessio Colzi, Tommaso Martini, Paolo Nesi, Davide Rogai (S)</i>	637

System Reliability and Verification

Adequacy of Composite Parametric Software Reliability Models <i>Lance Fiondella, Swapna S. Gokhale</i>	643
Evaluation of the OORT Techniques for Inspection of Requirements Specifications in UML: an empirical study <i>Tereza G. Kirner, Erik R. da Cruz (S)</i>	649

Agent Modeling/Methodology

Adjudicator: A Statistical Approach for Learning Ontology Concepts from Peer Agents <i>Behrouz Far, Abdel-Halim Hafez Elamy, Nora Houari, Mohsen Afsharchi</i>	654
---	-----

Agent Applications

DALICA: Intelligent Agents for User Profile Deduction <i>Stefania Costantini, Leonardo Mostarda, Arianna Tocchio, Panagiota Tsintza (S)</i>	660
--	-----

Human Interaction and GUI Development

- An Approach to Multimodal Input Interpretation in Human-Computer Interaction
Fernando Ferri, Patrizia Grifoni, Stefano Paolozzi 664
- Sketch Style Recognition in Human Computer Interaction
Daniilo Avola, Fernando Ferri, Patrizia Grifoni 670
- Human-Computer Interaction for a Novel Arm-wrestling Robot
Chul-goo Kang, Ho-yeon Kim (S) 676

Service

- Managing XML Versions and Replicas in a P2P Context
Deise de Brum Saccol, Nina Edelweiss, Renata de Matos Galante, Carlo Zaniolo 680
- Knowledge sharing through a simple release planning method for web application development
Sven Ziemer, Ilaria Canova Calori 686
- Distributed BPEL Processes
Luciano Baresi, Andrea Maurino, Stefano Modafferi 692
- SAM: Semantic Advanced Matchmaker
E.S. Ilhan, G.B. Akkus, A. B. Bener 698
- A development platform for distributed user interfaces
Anders Larsson, Magnus Ingmarsson, Bo Sun (S) 704

Security

- A Dynamical System Approach to Intrusion Detection Using System Call Analysis
Nitin Kanaskar, Remzi Seker, S. Ramaswamy 710
- Multi-level Anomaly Detection with Application-Level Data
Swapna S. Gokhale, Jijun Lu 718

Grid Technology

- A Four-layered Semantic Grid Architecture
Celia Ghedini Ralha, Jose Nelson C. Allemand, Alba C. M. Melo 724

Software Metrics, Measurement and Evaluation

Performance Analysis of the Active Object Pattern in Middleware <i>Paul J. Vandal, Swapna S. Gokhale, Aniruddha S. Gokhale</i>	730
Analyzing the Applicability of a Theoretical Model in the Evaluation of Functional Size Measurement Procedures <i>Nelly Condori-Fernandez, Oscar Pastor (S)</i>	736
Software Documents: Comparison and Measurement <i>Tom Arbuckle, Adam Balaban, Dennis K. Peters, Mark Lawford (S)</i>	740

Industrial Workshop

Workflow Management and Service Oriented Architecture <i>Theodorich Kopetzky, Dirk Draheim</i>	749
Implementing Agile Development - More than Changing Methodology <i>Chuck Fredrick</i>	751
Knowledge Modelling using UML <i>A. J. Rhem</i>	755
Reviewers' Index	758
Authors' Index	760

Note: (S) means short paper.

Using Model-Driven Pattern Matching to Derive Functionalities in Models

García-Rodríguez de Guzmán, I., Polo, M. and Piatini, M.

Abstract— Today, software engineering is evolving from source code to the model realm. As objects have been considered “atomic units” in software development and maintenance, models are becoming first order citizens in the well-known MDA paradigm. New languages and metamodel families are appearing to support the management of models and the new capabilities arising from this technological evolution. In this paper we present the MDPEM (Model-Driven Pattern Matching) technique and a possible application for models.

Index Terms— MDA, Model Instrumentation, Model-Driven Pattern Matching, OCL, QVT

I. INTRODUCTION

Software evolution is turning classic artefacts into models as a first order element in the development process. Current development paradigms are going through a change from object orientation to model-driven trends [1]. This evolution is partially supported by many proposals such as standard languages and metamodels. Perhaps one of the most popular initiative is OMG (*Object Management Group*), consisting of UML2 (*Unified Modeling Language*) [2], OCL2 (*Object Constraint Language*) [3], MOF2 (*Meta-Object Facility*) [4], CWM (*Common Warehouse Specification*) [5] and the recently appeared KDM (*Knowledge Discovery Metamodel*) [6] (intended to support the Architecture-Driven Modernization process [7]).

Since languages such as Java, SmallTalk and C++ met the new requirements when object orientation appeared, other kinds of languages are currently appearing to fulfil the new requirements of the MDA (*Model-Driven Architecture*) [8] paradigm. These languages are capable of managing models and metamodels as other languages, such as Java or C++, deal with objects and classes.

The MDA philosophy is creating new opportunities for software engineering. For example, software development can be supported by means of models [9, 10], reengineering for software modernization [11], system integration [12], etc.

Ignacio García-Rodríguez de Guzmán is with the Alarcos Research Group, Information Systems and Technologies Department, UCLM-SOLUZIONA Research and Development Institute, University of Castilla-La Mancha, Ciudad Real, Spain. (e-mail: ignacio.grodriguez@uclm.es)
Mariano Polo is with the Alarcos Research Group, Information Systems and Technologies Department, UCLM-SOLUZIONA Research and Development Institute, University of Castilla-La Mancha, Ciudad Real, Spain. (e-mail: Coral.Calero@uclm.es).

Mario Piatini is with the Alarcos Research Group, Information Systems and Technologies Department, UCLM-SOLUZIONA Research and Development Institute, University of Castilla-La Mancha, Ciudad Real, Spain. (e-mail: Mario.Piatini@uclm.es).

One of the most popular (and one of the most ambitious bets) to deal with models is QVT (Query/View/Transformations) [13]. This language, proposed by the OMG, is intended to perform operations with models. Using QVT, it is possible to throw queries against models, create views from models and of course, carry out transformations among models. One of the most common operations in QVT is pattern matching. For any operation, QVT expresses models as searching patterns. The QVT specification [13] establishes in which context pattern matching is used, but is does not make clear what to do when it is only necessary to look for occurrences of a given pattern.

Computer science has used pattern matching in many domains, such as lexical and syntactical analyzers, data mining, information retrieval, etc. But in most of them, pattern matching is carried out against text, files or other media. Our proposal is to use the latest advances in model technology to facilitate the application of the well-known technique of pattern matching.

This paper is organized as follows: Section II gives an overview of QVT and some related works; Section III introduces the concept of MDPEM; Section IV depicts what MDPEM could be useful for; Section V tackles a possible application of MDPEM in the aggregation of services to a system; and Section VI gives some conclusions and outlines some future lines of work.

II. BACKGROUND

A. About QVT

QVT is an initiative of the OMG. In 2002, it was published the QVT RFP. One year later in March 2003, the QVT-Partners published the “Initial submission for MOF 2.0 Query/Views/Transformations RFP” [14]. In November 2003, QVT-Partners published the “Revised submission for MOF 2.0 Query / Views / Transformations RFP” [15]. This specification showed a more complete specification along with the declarative and imperative QVT’s languages. In November 2005, OMG published the “MOF QVT Final Adopted Specification” [13].

In spite of the popularization of this language, there is no available any QVT engine implementing the declarative QVT language (up to now). Some projects in the academic world [16] are now underway. There are other initiatives such as the *MODELWARE QVT Tool* [17], based on the *QVT Operational language*, the other sublanguage of QVT.

B. Pattern Matching

Traditionally, patterns have been used for many purposes. For example, in some works [18, 19], patterns have constituted the cornerstone of the migration process from one kind of system to another. In other studies [20], the authors use a pattern-oriented language to integrate databases. Additionally, patterns have been used to reduce the inherent complexities that arise when applications must be connected to databases [20-22].

Patterns have been widely used to reengineer legacy systems [23], and particularly, patterns are very useful in the reverse engineering stage [24-26]. The well-known patterns of Gamma [27] have been used not only in the system design stage, but also with the aim of refactoring or restructuring legacy systems.

III. MODEL-DRIVEN PATTERN MATCHING

A. What does MDPEM consist of?

Model-Driven Pattern Matching is simply a traditional concept which has evolved together with software trends. The basic idea behind pattern matching is to find occurrences of a given pattern in a data set. Thus, MDPEM uses models to specify both the patterns and the target data set.

According to [28], "the essential idea behind pattern matching is to allow the succinct expression of complex constraints in an input data type; data which matches the pattern is then picked out and returned to the invoker". In the MDPEM context, the pattern model specifies the complex constraints and the target model represents the data.

B. A framework for MDPEM

As noted above, this paper focuses on the MDPEM concept in the QVT language; thus the developed framework to perform MDPEM is based on this model-oriented language.

The QVT language uses the concept of pattern matching in every action that its constructors perform intensively. But due to the novelty of QVT, there is neither foundation nor theory about how to use this language to perform pattern matching, and, additionally, the QVT specification [13] is confusing in this respect.

For this reason, a basic framework has been developed to support (at least on a theoretical level) the MDPEM process. Fig. 1 depicts the framework. For the sake of simplicity all the involved artifacts are located in an MOF-like pyramid. Level M2 of Fig. 1 represents the metamodels for patterns and target models, as well as the model transformations involved in the process; Level M1 represents all the models involved in the MDPEM process, namely the patterns, the target model, and the matchings (occurrences of the pattern in the target model). Since all the steps are developed in M2 and M1, M3 and M0 are not explained. Table 1 summarizes the elements represented in Fig. 1. For each element ("Level M2" column in TABLE I) on the M2 level, there exists a conforming model ("Level M1" column in TABLE I) on the M1 level.

The PatternToQVTPattern transformation, not included in Table 1, plays an important role in the MDPEM process. As

noted above, any pattern (PatternModel) must be transformed into a different representation that QVT imposes: the QVT Template (shown as QVTPattern Model in Fig. 1). The QVT Template is the structure that QVT uses to represent any pattern that needs to be matched. As a consequence, any pattern we want to match should be transformed to this representation. Therefore, a transformation (PatternToQVTPattern in Fig. 1, Level M2) involves both the pattern metamodel (PatternMMModel) and the QVTPattern MModel (QVT Template Metamodel).

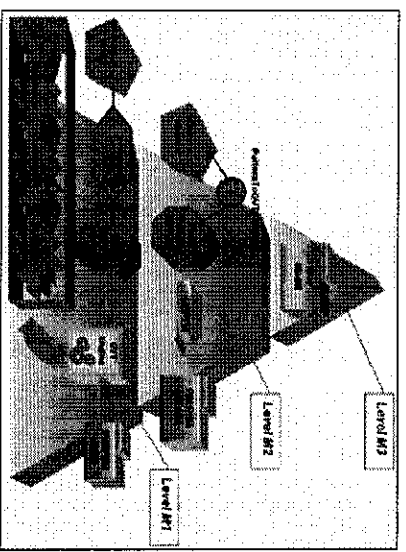


Fig. 1. MDPEM Framework

TABLE I
ELEMENTS INVOLVED IN THE MDPEM PROCESS

Level M2	Level M1	Description
PMM	PM	Patterns (models) are constrained by a given metamodel. This must be compatible with the target model to be able to find matchings.
QVTMM	QVTM	Patterns must be translated into a QVT pattern representation (QVT Template [13]), so the QVT Template metamodel (QVTMM) should be taken into account. As a consequence, each pattern model is transformed to a QVT Template (QVTM).
PIM/PSMM	PIM/PSM	Because in any MDA process models are called PIM or PSM (according to the abstraction level of the model), we suppose that the target model will be a PIM or a PSM.
-	Matching	After MDPEM is applied, matchings may be found. This matchings can be seen as fragments of the target model, so the metamodel for these elements can also be the PIM/PSMM.
PMM = Pattern Metamodel; PM = Pattern Model; QVTMM = QVT Pattern Metamodel; QVTM = QVT Pattern Model; PIM/PSMM = PIM/PSM Metamodel; PIM/PSM = PIM/PSM Model		

The MDPEM process can be divided into four steps:

1. Pattern model and target model are defined.
2. QVT Template model is obtained from Pattern Model.
3. MDPEM is carried out.
4. Matchings (sub-models) are returned to the invoker.

For a deeper explanation of MDPEM and examples of

patterns, please see [29].

IV. FINDING/DEFINING DERIVING PATTERN ON MODELS

The previous section outlined the MDPEM technique, but the main goal of this technique is not only to find occurrences for a given pattern. MDPEM can be also used to undertake more detailed operations over those models acting as target models.

This technique was conceived as a tool to support some parts of an MDA process to extract services from relational databases [30]. In this process, patterns are used to extract functionality from relational databases. That led us to the question: how to use MDPEM to do something more than find matchings?

A. Patterns plus actions to derive functionalities

In this paper our aim is to provide a mechanism to use MDPEM undertaking actions associated with the set of occurrences obtained in the matching process (Fig. 2).

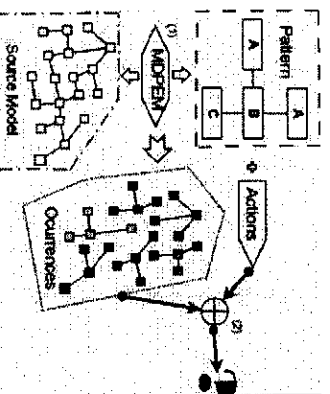


Fig. 2. Extended MDPEM

The basic MDPEM depicted in Fig. 1 can be expressed as a function f_{MDPEM} . This function is defined as follows: $f_{MDPEM}(M_s, M_T) = Set(M_G)$, where M_s represents the pattern, M_T represents the source model, and M_G represents an occurrence of M_p in M_T . Depending on both the pattern model and the target model, f_{MDPEM} may return a set of occurrences ($Set(M_G)$).

But according to Fig. 2, the MDPEM basic concept can be extended to provide an extra functionality. Patterns can be provided together with a description of an abstract action. The basic idea behind Fig. 2 is to have not only a pattern (M_p) but also an associated abstract action, which could be later transformed into a concrete transformation.

However, MDPEM can be specified in a more powerful way. First of all, instead of a pattern M_p , the process takes a pair $\langle \text{pattern, abstract action} \rangle$, algebraically represented as $K = (M_p, A_p)$, where M_p represents the aforementioned pattern model, and A_p represents the abstract action. Therefore, the previously defined function f_{MDPEM} must be redefined according to the new element K . This new function can be considered as follows: $f_{MDPEM}(K, M_T) = Set(A_A(M_G))$. A_A is then applied to each element in the resulting set of occurrences. Thus the abstract action, which is defined in

terms of the abstract elements of the pattern, is then "materialized" for each actual occurrence obtained in the MDPEM process.

In Fig. 2, K is represented by the pattern plus the action (top of the figure), and M_T is the source model depicted at the bottom of figure. After applying MDPEM and obtaining the occurrences, the specified action is then applied to each occurrence.

The term "abstract action" may be very vague, but in fact any action or structure can be specified to be applied to each occurrence. As a consequence, certain parts from models (MDPEM occurrences) can be updated or modified according to the instructions given in the abstract action (A_p).

In the following section, we will describe what MDPEM has been used for: to detect services in databases using patterns and abstract descriptions for the services attached to patterns.

V. USING MDPEM TO FIND AND CREATE SERVICES

This section explains the purpose of this technique. As noted, MDPEM was developed in the context of an MDA process to extract services from relational databases. The lack of tools and experience around languages such as QVT prompted the development of MDPEM to properly manage models and patterns.

In the context of the process proposed in [30], we can define a pattern and attach a particular behavior to this pattern. This behavior is specified in terms of operations, called services. Thus, for a given pattern involving a set of entities, a set of abstract operations involving these entities can be attached. Thus, when an occurrence is found, each abstract operation attached to the pattern is "instantiated" with the entities of the occurrence.

A. Problem definition

MDPEM can be used to discover services in relational databases. Thus, all the artifacts (patterns and target model) will be described in terms of tables, columns, foreign keys and so on.

The aim is to start from a database model, define a set of patterns, find their matches and obtain possible useful services. This case study fits the example depicted in Fig. 2, and:

- the database model plays the role of the source model,
- a pattern describing a particular database fragment structure plays the role of the pattern, and
- the abstract description of an operation involving the elements of the pattern.

B. Pattern model and target model definition

Fig. 3 depicts a fragment from a relational database extracted from an industrial application. This fragment is composed of 10 tables and will form the *target model* for the MDPEM process. This model has been used to carry out a simple case

occurrences of the pattern) can be annotated with operations or services (described as abstract operations related to the searching pattern). The instrumentation of models with these services facilitates the implementation of MDA-based development processes.

As well as development, MDPEM might be useful in other areas. Software maintenance in a model-level can be improved using this technique, e.g., looking for design patterns, as well as looking for bad-smells and correcting them (in a model – level). Every process described as a “*search pattern and apply rule*” sequence can be partially automated and then easily applied. Due to the lack of tools for implementing the QVT language, MDPEM has not been tested in an automated environment. As soon as a tool is available, this theoretical proposal will be migrated to a *runnable* one.

For each pattern metamodel, it is necessary to develop a suitable transformation to generate the equivalent *QVT Template* model. Therefore, it may be useful to automate the generation of the *QVT Template* from some pattern metamodels.

ACKNOWLEDGMENTS

This work was partially supported by the ENIGMAS project (PIB-05-058), the FAMOSO Project (2006: FIT-349000-2006-67), and the MECENAS project (PBI06-0024).

REFERENCES

- [11]. Bézhin, J. *Introduction to Model Engineering*. 2006. <<http://www.modelware-ist.org/>> [Accessed: 14-November-2006]
- [12]. OMG. *Unified Modeling Language: Superstructure, Version 2.0*. 2005. <<http://www.omg.org/docstore/05-07-04.pdf>> [Consultado el: 21 de Noviembre del 2005]
- [13]. OMG. *OCL 2.0 Specification, Version 2.0*. 2005. Object Management Group (OMG), p. 185.
- [14]. OMG. *Meta Object Facility (MOF) Specification*. 2002. <<http://www.omg.org/docstore/formal/02-04-03.pdf>> [Accessed: 11-05-2005]
- [15]. OMG. *Common Warehouse Metamodel (CWM) Specification*. 2003. <<http://www.omg.org/docstore/formal/03-03-02.pdf>> [Consultado el: 29-09-2005]
- [16]. OMG. *Architecture-driven Modernization (ADM): Knowledge Discovery Metamodel (KDM) Specification*. 2006. <<http://www.omg.org/docstore/06-06-07.pdf>> [Accessed in February, 2006]
- [17]. OMG. *ADM Task Force*. 2006. <<http://adm.omg.org/>> [Accessed in February, 2007]
- [18]. OMG. *MDA Guide Version 1.0.1*. 2003. Object Management Group, p. 62.
- [19]. Maamar, Z., et al. *Towards a contextual model-driven development approach for Web services*. in *Proceedings of the 8th International Conference on Enterprise Information Systems (ICEIS 2006)*. 2006. Pafos, Cyprus
- [10]. Xiaohong, Y., et al. *A Model Driven Development Framework for Enterprise Web Services*. in *Proceedings of the 8th International Conference on Enterprise Information Systems (ICEIS 2006)*. 2006. Hong Kong, China: IEEB Computer Society IEEB Computer Society. ISBN. 0-7695-2538-X
- [11]. Reus, T., H. Geens, and A. van Deursen. *Harvesting Software for MDA-Based Recovering*. in *European Conference on Model Driven Architecture - Foundations and Applications*. 2006. Bilbao (Spain): Springer-Verlag Berlin Heidelberg, LNCS 4066 Springer-Verlag Berlin Heidelberg, A. Reusink and J. Warner, Editors. ISBN. 3-540-35909-5
- [12]. van den Heuvel, W.-J. *Matching and Adaptation: Core Techniques for MDA-(ADM)-driven Integration of new Business Applications with Hyppped Legacy Systems*. in *Model-Driven Evolution of Legacy Systems (MAELS 2004)*. 2004. Monterey, California, USA
- [13]. OMG. *MOF QVT Final Adopted Specification*. 2005. <<http://www.omg.org/docstore/05-11-01.pdf>> [Accessed in February, 2005]
- [14]. OMG. *QVT-Partners: initial submission to qvt-rfp*. 2003. <<http://www.qvt.org/doc/initial1.0/qvtpartners1.0.pdf>> [Accessed in November, 2006]
- [15]. OMG. *Revised submission for MOF 2.0 QueryViews: Transformations RFP*. 2003. <<http://www.omg.org/docstore/04-04-01.pdf>> [Accessed in November, 2006]
- [16]. Queral, P., et al. *Un Motor de Transformación de Modelos con soporte para el Lenguaje QVT Relations*. in *Desarrollo del Software Dirigida por Modelos y Aplicaciones 2006 (DSDM'06)*. 2006. Siles, Barcelona (Spain)
- [17]. ModelWare. *The ModelWare QVT Tool*. 2007. <<http://www.modelware-ist.org/>> [Accessed: 15-02-2007]
- [18]. Bothain, T., E. Guardasacco, and M. Torrella. *Migrating COBOL Systems to the WEB by Using the MFC Design Pattern*. in *Ninth Working Conference on Reverse Engineering (WCRE'02)*. 2002. Richmond, Virginia: IEEB Computer Society IEEB Computer Society
- [19]. Goedele, M., and U. Zdun. *Preceneal legacy migrating with an architectural pattern language: a case study*. *Journal of Software Maintenance and Evolution: Research and Practice*, 2002. 14, p. 1-30.
- [20]. Brown, K., and B.G. Whitehead. *Crossing Chasms, A Pattern Language for Object-RDBMS Integration*, ed. K.S.C. Editor. 1995. K.S.C. Editor.
- [21]. Yoder, J. *Patterns for making business objects persistent in a relational database*. in *Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*. 2002. Tampa Bay (Florida)
- [22]. Yoder, J., R.E. Johnson, and Q.D. Wilson. *Connecting Business Objects to Relational Databases*. in *Proceedings of the 5th Conference on the Pattern Languages of Programs*. 1998. Mitchellville-IL (Estados Unidos)
- [23]. Cagnin, M.L., et al. *Reengineering Using Design Patterns*. in *Seventh Working Conference on Reverse Engineering (WCRE'00)*. 2000. Brisbane, Australia: IEEB IEEB
- [24]. Sarrigi, K. and K. Konogiannis. *On Modeling Software Architecture Recovery as Graph Matching*. in *International Conference on Software Maintenance (ICSM'03)*. 2003. Amsterdam, The Netherlands: IEEB Computer Society IEEB Computer Society
- [25]. Pinzer, M., et al. *Reveler: A Lectoral Pattern Matcher for Architecture Recovery*. in *Proceedings of the 9th Working Conference on Reverse Engineering*. 2002. Richmond, Virginia: IEEB Computer Society IEEB Computer Society
- [26]. Stoemper, C., L. O'Brien, and C. Verhoef. *Practice Patterns for Architecture Reconstruction*. in *Proceedings of the 9th Working Conference on Reverse Engineering (WCRE'02)*. 2002. Richmond, Virginia: IEEB Computer Society IEEB Computer Society
- [27]. Ganama, E., et al. *Design Patterns Elements of Resizable of Object-Oriented Software*. 1995: Addison-Wesley.
- [28]. QVTP. *Revised submission for MOF 2.0 Query / Views Transformations RFP (Version 1.1)*. 2003. <<http://tratl.ucslauticsresearch/publications/transforms/guarners1.1.pdf>> [Consultado el: 23-06-2005]
- [29]. Garcia-Rodriguez de Guzman, I., M. Polo, and M. Pardini. *A Framework for Model-Driven Pattern Matching*. in *Proceedings of the 9th International Conference on Enterprise Information Systems*. 2007. Funchal, Madeira - Portugal (In press)
- [30]. Garcia-Rodriguez de Guzman, I., M. Polo, and M. Pardini. *A Methodology for Database Reengineering in Web Services*. in *European Conference on Model Driven Architecture - Foundations and Applications (ECMDA-FA 2006)*. 2006. Bilbao (Spain): Springer-Verlag Berlin Heidelberg, LNCS 4066 Springer-Verlag Berlin Heidelberg, A. Reusink and J. Warner, Editors. ISBN. 3-540-35909-5
- [31]. Calero, C., et al. *An Ontological Approach To Describe the SQL:2003 Object-Relational Features*. Accepted in "Computer Standards and Interfaces", 2005: p. 28
- [32]. Rynosno, L., M. Genaro, and M. Pardini. *OCL2: Using OCL in the Formal Definition of OCL Expression Measures*. in *Quality in Modeling (QM'2006)*. 2006. Genova (Italy)