

**July 1-3, 2003**

**San Francisco, California, USA**

# SEKE

**Proceedings of the Fifteenth International  
Conference on Software Engineering &  
Knowledge Engineering**

**Knowledge Systems Institute**





**Knowledge Systems Institute  
Graduate School of Computer & Information Sciences  
3420 Main Street  
Skokie, Illinois 60076**

**July 1-3, 2003  
San Francisco, California, USA**

# SEKE

**Printed In USA, June 2003  
ISBN 1-891706-12-8**

**PROCEEDINGS**

**SEKE 2003**

**The 15<sup>th</sup> International Conference on  
Software Engineering & Knowledge Engineering**

**Sponsored by**

**Knowledge Systems Institute, USA**

**Technical Program**

**July 1-3, 2003**

**San Francisco, California, USA**

**Organized by**

**Knowledge Systems Institute**

**SEKE 2003**  
Fifteenth International Conference on  
Software Engineering and Knowledge Engineering

**Conference Organization**

**Conference Co-Chairs**

C. V. Ramamoorthy, University of California, Berkeley  
Shi Kuo Chang, Knowledge Systems Institute, Chicago

**Program Co-Chairs**

Kang Zhang, University of Texas at Dallas, USA  
John Debenham, University of Technology, Sydney, Australia

**Steering Committee**

Vic Basili, University of Maryland, USA  
Bruce Buchanan, University of Pittsburgh, USA  
Shi-Kuo Chang, University of Pittsburgh and Knowledge Systems Institute, USA  
C. V. Ramamoorthy, University of California, USA

Copyright © 2003 by Knowledge Systems Institute

All right 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 permission of the publisher.

ISBN 1-891706-12-8 (paper)

Additional Copies can be ordered from:  
Knowledge Systems Institute  
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

Printed in the United States of America

**Program Committee**

Silvia Teresita Acuña, Argentina  
Vincenzo Ambriola, Italy  
Anneliese Amsler Andrews, USA  
Tim Arndt, USA  
Egidio Astesiano, Italy  
Juan Carlos Augusto, UK  
Antonia Bertolino, Italy  
Leopoldo Bertossi, Canada  
Alf Bertz, USA  
Lionel Briand, Canada  
Augusto Celentano, Italy  
William Chu, Taiwan  
Paolo Ciancarini, Italy  
Gennaro Costagliola, Italy  
Honghua Dai, Australia  
John Debenham, Australia  
Yi Deng, USA  
Gregor Engels, Germany  
Filomena Ferrucci, Italy  
Antonio Fresa, Italy  
Alfonso Fuggetta, Italy

Carlo Ghezzi, Italy  
Athula Ginge, Australia  
Angela Guercio, USA  
Qinming He, China  
Maolin Huang, Australia  
Letizia Jaccheri, Norway  
Erland Jungert, Sweden  
Natalia Juristo,  
Latifur Khan, USA  
Robert Laurini, France  
Stefano Levialdi, Italy  
Jay Liebowitz, USA  
Huimin Lin, China  
Mikael Lindwall, USA  
Jiming Liu, Hong Kong  
Vincenzo Loia, Italy  
Andrea De Lucia, Italy  
Maurizio Martelli, Italy  
Deependra Moitra, India  
Sandro Morasca, Italy  
Lakshmi Narasimhan, USA  
Paolo Nesti, Italy  
Mehmet A. Orgun, Australia

David Robertson, UK  
Guenther Ruhe, Canada  
Ioana Rus, USA  
Walt Scacchi, USA  
Phil Sheu, USA

Margaret-Anne Storey, Canada  
Scott Tilley, USA  
Genny Tortora, Italy  
Jeffrey Tsai, USA

M. Vazirgiannis, Greece

Sira Vegas, Spain

Giuseppe Visaggio, Italy

Giuliana Vitiello, Italy

Christiane Gresse von Wangenheim, Brazil

Stefan Wermter, UK

Xindong Wu, USA

Yiyu Yao, Canada

Marvin Zelkowitz, USA

### Workshop on Data Mining for Software Engineering and Knowledge Engineering

#### Workshop Chair/Co-chairs

Honghua Dai, Deakin University, Australia  
Geoff Webb, Monash University, Australia

#### Workshop Publicity Chair

Gang Li, Deakin University, Australia

#### Workshop Program Committee

Keith Chan, Hong Kong  
Lai, Chih, USA

Honghua Dai, Australia

Gang Li, Australia

James Liu, Hong Kong

Zongtian Liu, China

Walcelio L. Melo, USA

Hiroshi Motoda, Japan

Claudio Nehme, Brazil

Chinya Ravishanker, USA

Javier Segovia, Spain

Simoon J. Simoff, Australia

Zhihai Wang, Australia

Geoff Webb, Australia

Xindong Wu, USA

Min Yao, China

Aoying Zhou, China

Zhi Hua Zhou, China

### Workshop on Software Engineering Decision Support

#### Workshop Chair

Gtntner Ruhe, University of Calgary, Canada

#### Workshop Program Committee

Stephan Biffl, Austria

Lionel Briand, Canada

Khaled El-Emam, Canada

Ross Jeffery, Australia

Sandro Morasca, Italy

David Raffo, USA

Gtntner Ruhe, University of Calgary, Canada

Ioana Rus, USA

Claes Wohlin, Sweden

### Workshop on Web Engineering

#### Workshop Chair

Athula Ginige, University of Western Sydney, Australia

#### Workshop Program Committee

Athula Ginige, University of Western Sydney, Australia

Martin Gaedke, University of Karlsruhe, Germany  
Bebo White, Stanford University, USA

Paul Dantzig, IBM T J Watson Research Center, USA

Steve Hansen, University of Western Sydney, Australia

Yogesh Deshpande, University of Western Sydney, Australia

San Murugesan, Southern Cross University, Australia

Filomena Ferrucci, Università di Salerno, Italy

#### Conference Publicity Chair

Eric Wong, University of Texas at Dallas, USA

#### Conference Finance Chair

Judy Pan, KSI, USA

#### Conference Secretariat

C. C. Huang, KSI, USA

Rex Lee, KSI, USA

### SEKE'2003 Preface

On behalf of the Program Committee of the Fifteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'2003). We would like to welcome you to San Francisco, USA. The conference aims at bringing together experts in software engineering and knowledge engineering to discuss on relevant results in either software engineering or knowledge engineering or both. Special emphasis is on the transference of methods between both domains.

It is our pleasure to announce that by the deadline of 10 March 2003, the conference and workshops received 159 submissions from 26 countries. All the papers have been rigorously reviewed by at least 2-3 members of the international Program Committee. Based on the review result, 71 papers have been accepted as regular papers, with an acceptance rate of 44.7%, and 35 accepted as short papers. We would like to thank all the authors for their contributions.

This year, we have a rich collection of activities in the technical program, including two keynote speeches, one tutorial, three workshops, and 20 technical sessions. The keynotes, tutorial, workshops, and technical sessions cover a wide range of topics in software engineering and knowledge engineering, including:

- Data Mining for Smarter Project Management,
- Web Engineering,
- Software Engineering Decision Support,
- Data Mining for Software Engineering and Knowledge Engineering,
- Software Maintenance, Software Measurements,
- Software Process and Architecture,
- Software Visualization and Comprehension,
- Agent Systems,
- Formal Methods,
- Software and Knowledge Reuse,
- UML,
- Knowledge Acquisition,
- Knowledge Retrieval and Management,
- Integrity, Security and Fault-Tolerance,
- Distributed Software Development, and
- Spatial Reasoning and Search.

We are very grateful to the two keynote speakers, Athula Ginige and Gio Wiederhold, the tutorial and workshop organizers, Tim Arndt, Juan Carlos Augusto, Gary D. Boetticher, Honghua Dai, Erland Jungert, Tim Menzies, Guenther Ruhe, and M. Vazirgiannis. The Publicity Chair, Eric Wong, and members of the Program Committee should be congratulated and specially thanked for their publicity effort and timely reviews of the submitted papers.

Finally, we would like to thank Shi-Kuo Chang for his guidance and leadership throughout organization of this conference. The assistance of the staff at KSI is also greatly appreciated. C. C. Huang and Rex Lee at KSI handle the registration and organization of the proceedings. Special thanks go to Yu Qian, a PhD student at UT-Dallas, for his effective and efficient assistance in working with the paper submission and review system, which has made the review process smooth and timely. Finally, we would like to thank Judy Pan at KSI, who carefully arranged the details of the reception, the boat tour and dinner that have greatly enhanced this conference.

John Debenham  
University of Technology, Sydney, Australia

Kang Zhang  
University of Texas at Dallas, USA

## Table of Contents

Conference Organization.....	iii
Preface.....	v
 <b>Keynote 1:</b>	
Re-engineering Software Development Process for eBusiness Application Development .. <i>Athula Ginige, University of Western Sydney, Australia</i>	1
 <b>Session 1A: Web Engineering</b>	
Revealing Web User Requirements through e-Prototyping..... <i>Martin Jeenicke, Wolf Gideo-Bleek, Ralf Klischewski, University of Hamburg, Germany</i>	9
WebML+ : a Web modeling language for forming a bridge between business modeling and information modeling..... <i>Rachatrin Tongrungrrojana, David Lowe, University of Technology, Sydney, Australia</i>	17
An Evaluation of Web3d Technologies from Developer's and End-User's Point of View..... <i>Gennario Costagliola, Sergio di Martino, Filomena Ferrucci, Università degli Studi di Salerno, Italy</i>	25
Representing and Querying the Evolution of Databases and their Schemas in XML..... <i>Fusheng Wang and Carlo Zaniolo, University of California, Los Angeles</i>	33
 <b>Session 1B: Software Maintenance</b>	
Corrective Maintenance Process: a Case Study from a Telecom Software Development and Maintenance Organization..... <i>Antonio Fresa, M.C. Annosi, F. Forte, Ericsson Lab Italy</i>	39
Revision Recognition for Scientific Computing: Theory and Application..... <i>Andrew P. Strelzoff, University of California - Santa Barbara, USA</i>	46
Adding Verification Property of Inter-Processes Using Aspect-Oriented Approach..... <i>Pantti Netiniantti, Bangkok University, Thailand</i> <i>Tzilla Elrad, Illinois Institute of Technology, U.S.A.</i>	54

Knowledge for Software Maintenance..... <i>Nicolas Anquetil, Kátia Oliveira, Márcio Greyck Dias, Marcelo Ramal, Ricardo Meneses, Universidade Católica de Brasília, Brazil</i>	61
 <b>Session 2A: Software Measurements</b>	
Application of Neural Networks for Estimating Software Maintainability Using Object- Oriented Metrics..... <i>Mie Mie Thei Thwin, Tong-Seng Quah, Nanyang Technological University, Singapore</i>	69
Spatial Data Analysis as a Software Quality Modeling Technique..... <i>Justin M. Beaver, NASA, USA, Guy A. Schiavone, University of Central Florida, USA</i>	74
Towards an Ontology for Software Measurement..... <i>Marcela Genero, Francisco Ruiz, Mario Piattini and Coral Calero, University of Castilla - La Mancha, Spain</i>	78
 <b>Session 2B: Agent Systems 1</b>	
Open Giant Intelligent Information Systems and Its Agent-Oriented Analysis and Design..... <i>Longbing Cao, Chunsheng Li, Chengqi Zhang, University of Technology, Sydney, Australia, Ruwei Dai, Institute of Automation, Chinese Academy of Sciences, China</i>	85
An Ontology Driven Design Method for inter-Agent Communication..... <i>P. Ciancarini, V. Presutti, L. Abeti, University of Bologna, Italy</i>	90
From Actors to Applications: Interpreting User Centered Design of Coordination Supports..... <i>Marcello Sarini, Carla Simone, Università degli Studi Milano-Bicocca, Italy</i>	95
Social Patterns for Designing Multiagent Systems..... <i>T. Tung Do, Manuel Kolp, Alain Pirotte, Catholic University of Louvain, Belgium</i>	103
 <b>Session 3A: Software Visualization and Comprehension 1</b>	
Automatic Abstraction of Graphs Based on Node Similarity for Graph Visualization..... <i>Xiaodi Huang and Wei Lai, Swinburne University of Technology, Australia</i>	111
Exploring UDDI Registries Using Modified OFDAV Browser..... <i>Biao Jiang, Mao Lin Huang, University of Technology, Sydney, Australia</i>	119
Extending UML to Visualize Design Patterns In Class Diagrams..... <i>Jing Dong and Sheng Yang, University of Texas at Dallas, USA</i>	124

Visualizing Software Architecture with Off-The-Shelf Components.....	132
<i>Jie Ren, Richard N. Taylor, University of California at Irvine, USA</i>	

**Session 3B: Formal Methods**

Semantics of State-Oriented Expressions in the Object Constraint Language.....	142
<i>Stephan Flake and Wolfgang Mueller, Paderborn University, Germany</i>	

Deriving Hierarchical Predicate/Transition Nets from Statechart Diagrams.....	150
<i>Zhijiang Dong, Yujian Fu, Xudong He, Florida International University, USA</i>	

Lurch: a Lightweight Alternative to Model Checking.....	158
<i>David Owen, Tim Menzies, West Virginia University, USA</i>	

TILCO Temporal Logic for Real-Time Systems Implementation in C++.....	166
<i>P. Bellini, A. Giotti, P. Nesi, D. Rogai, University of Florence, Italy</i>	

Automated Support for Property Specification Based on Patterns.....	174
<i>Oscar A. Mondragon, Arn Q. Gates, and Fadi Kassem, University of Texas at El Paso, USA</i>	

**Keynote 2**

The Product Flow Model.....	182
<i>Gio Wiederhold, Stanford University, USA</i>	

**Session 3A: Software Visualization and Comprehension 2**

The Role of Knowledge in Software Customization.....	187
<i>Jeff Michaud and Margaret-Anne Storey, University of Victoria, Canada.</i>	

Using Critiquing Systems for Inconsistency Detection in Software Engineering Models...	196
<i>Cleidson R. B. de Souza, Hamilton L. R. Oliveira, Cleber R. P. da Rocha, Kleider M. Goncalves, University of Par�, Brazil</i>	
<i>David F. Redmiles, University of California, Irvine, USA</i>	

Partial Slicing for Large Programs.....	204
<i>Zhenqiang Chen Baowen Xu, Southeast University, Nanjing, China</i>	
<i>William Chu, TungHai University, Taiwan</i>	
<i>Hongji Yang, De Montfort University, England</i>	
<i>Jianjun Zhao, Fukuoka Institute of Technology, Japan</i>	

Constructing Understandability Model from Design Metrics.....	208
<i>Matinee Kiewkanya, Nongyao Jindasawat, Nakornthip Prompoon, Pornsiri Muenchaisri, Chulalongkorn University, Thailand</i>	

**Session 4B: Agent Systems 2**

Developmental and Operational Processes for Agent-Oriented Database Navigation for Knowledge Discovery.....	216
<i>M. Brian Blake, Georgetown University, USA</i>	
<i>Andrew B. Williams, University of Iowa, USA</i>	

Dealing with software design issues using an Agent-Oriented methodology.....	224
<i>Anna Perini, Angelo Susi, ITC-irst, Italy</i>	

Specifying and Analyzing Agent Architectures using the Agent Competency Framework.	232
<i>K. S. Barber and D. N. Lam, University of Texas at Austin, USA</i>	

**Session 5A: Data Mining Approach to SE and KE**

Tree-Growth based Sequential and Associative Pattern Discovery.....	240
<i>M. Baumgarten, A.G. B�chner, J.G. Hughes, University of Ulster, UK</i>	

An Efficient Knowledge Management Technique for finding Association Rules.....	245
<i>Farhan Rafique, A.Anwar, NUST Institute of Information Technology, Pakistan</i>	

CHiC: A Fast Concept Hierarchy Constructor for Discrete or Mixed Mode Databases.....	250
<i>Ola Agren, Ume� University, Ume�, Sweden</i>	

Real-Time Mining of Partial Periodic Patterns.....	259
<i>Chih Lai, Lawrence Stephanie, Meng Fang, University of St. Thomas, USA</i>	

**Session 5B: Software and Knowledge Reuse**

An XVCL-based Approach to Software Product Line Development.....	267
<i>Hongyu Zhang and Stan Jarzabek, National University of Singapore, Singapore</i>	

Extracting Knowledge from Technical Documents.....	276
<i>Joseph Neelamkavil and Michael Kernahan, National Research Council, Canada</i>	

Case-based Framework Instantiation.....	281
<i>Carlos J. Fern�ndez-Conde, Pedro A. Gonz�lez-Calero, Univ. Complutense de Madrid, Spain</i>	

Selection and Reuse of Software Design Patterns Using CBR and WordNet.....	289
<i>Paulo Gomes, Francisco C. Pereira, Paulo Paiva, Nuno Seco, Paulo Carreiro, José L. Ferreira and Carlos Bento, Universidade de Coimbra, Portugal</i>	
ROADS: A Reusable, Optimizable Architecture for Decision Systems.....	297
<i>Kareem S. Aggour and Marc Pavese, GE Global Research, USA</i>	
<b>Session 6A: UML</b>	
Extending UML to Specify and Verify E-commerce Systems.....	306
<i>Mark Song, Adriano Pereira, Fernanda Lima, Gustavo Gorgulho, Sérgio Campos, Wagner Meira Jr., Federal University of Minas Gerais (UFMG), Brazil</i>	
Simulation-based Verification of UML models.....	314
<i>Luigi Lavazza, Politecnico di Milano, Italy, Giuseppe Occorso, Technology REPLY, Italy</i>	
A Brief Overview of HyM: A Methodology for the Development of Hybrid Intelligent Information Systems.....	322
<i>S. L. Kendal, K Ashton, X Chen, University of Sunderland, UK</i>	
Rigorous Description of Software Requirements with UML.....	327
<i>Luigi Lavazza, Politecnico di Milano, Italy</i>	
Case-Based Reuse of UML Diagrams.....	335
<i>Paulo Gomes, Francisco C. Pereira, Paulo Paiva, Nuno Seco, Paulo Carreiro, José L. Ferreira and Carlos Bento, Universidade de Coimbra, Portugal</i>	
<b>Session 6B: Software Process and Architecture 1</b>	
Software Engineering as Technology Transfer.....	340
<i>Daniel E. Cooke, Jason Denton, Texas Tech University, USA</i>	
Relational Program Architecture for High Quality Software Development.....	346
<i>Dongfeng Wang, Farokh B. Bastani, and I-Ling Yen, University of Texas at Dallas, USA</i>	
A Data Storage Mechanism for Peer-to-Peer Based Decentralised Workflow Systems.....	354
<i>Jun Yan, Yun Yang, Gitesh K. Raikundalia, Swinburne University of Technology, Australia</i>	
On Variants of Genericity.....	359
<i>Thomas Schöbel-Theuer, University of Stuttgart, Germany</i>	

A Framework for Guiding the Design of Effective Implementation Strategies for Software Process Improvement.....	366
<i>Mahmood Niazi, David Wilson and Didar Zowghi, University of Technology Sydney, Australia</i>	
<b>Session 7A: Knowledge Retrieval and Management</b>	
Knowledge Organization and Retrieval in the MILK System.....	372
<i>Roberto Boselli, Flavio DePaoli, Riccardo Dondi, Università di Milano Bicocca, Italy</i>	
A Practical Way to Use Clustering and Context Knowledge for Software Project Planning	377
<i>Jürgen Münch, Fraunhofer Institute for Experimental Software Engineering, Germany Jens Heidrich, Alexandra Daskovska, University of Kaiserslautern, Germany</i>	
On Identifying Deficiencies in a Knowledge Management System.....	385
<i>Prashant Baheti, Laurie Williams, North Carolina State Univ., USA Aldo Dagnino, Andrew Cordes, ABB Inc., USA</i>	
Knowledge Management for Project Planning and Enactment in Software Engineering....	393
<i>Sigrid Goldmann, Harald Holz, Mich M. Richter, University of Kaiserslautern, Germany</i>	
<b>Session 7B: Software Process and Architecture 2</b>	
An Integrated Method to Software Architecture.....	398
<i>Hassan Reza, University of North Dakota, USA Xudong He, School of Computer Science, Florida International University, USA</i>	
Reasoning about Variability and structural Adaptation in Sociotechnical Systems Using Dynamic Architectures.....	406
<i>Michael Cebulla, Technische Universität Berlin, Germany</i>	
A Layered Architecture to Manage Complex Multimedia Services.....	414
<i>M. Longo, University of Salerno, Italy, P. Asprino, A. Fresa, N. Gaito, CoRiTel, Italy</i>	
A Pattern-like Framework to Dynamically Change Components Behaviour.....	422
<i>Daniela Micucci, Andrea Trentini, Università Studi Milano, Italy</i>	
<b>Session 8A: Knowledge Acquisition</b>	
Relative Index Term Frequency (RITF) Method.....	427
<i>Zhou Jiehan, VTT Electronics, Finland, LAN Haiyang, Tsinghua University, China</i>	



Tagging Knowledge Acquisition Sessions to Facilitate Knowledge Traceability.....	432
<i>Heather Richter, Gregory D. Abowd, Georgia Tech, USA</i> <i>Chris Miller, Harry Funk, Smart Information Flow Technologies (SIFT)</i>	
Integrating Signature Matching into Knowledge-Based Program Understanding.....	440
<i>Yachai Limpiyakorn, Chulalongkorn University, Thailand</i> <i>Ilene Burnstein, Illinois Institute of Technology, USA</i>	
MFILM: a multi-dimensional fuzzy inductive learning method for knowledge acquisition.....	445
<i>Chen, Yao-Tsung, Institute of Information Science, Academia Sinica, Taiwan</i> <i>Jeng, Bingchiang, National Sun Yat-sen University, Taiwan</i>	
<b>Session 8B: Integrity, Security and Fault-Tolerance</b>	
Formal Software Architecture Design of Secure Distributed Systems.....	450
<i>Huiqun Yu, Xudong He, Shu Gao, and Yi Deng, Florida International University, USA</i>	
Specification and Validation of Fault-Tolerant Software Architectures Based on Actor Model.....	458
<i>Hui Ding, Can Zheng, Lai Sha, Gul Agha, Univ. of Illinois at Urbana-Champaign, USA</i>	
Automated Software Design of Process-Control Systems.....	467
<i>Dongfeng Wang, Farokh B. Bastani, and I-Ling Yen, University of Texas at Dallas, USA</i>	
Defining Change Management Properties for Component Interoperability Assessment.....	475
<i>T. Gamble, R. Gamble, L. Davis, University of Tulsa, USA</i>	
A Step toward building Dynamic Security Infrastructure.....	483
<i>Ibrahim S. Abdullah, Edgar. H. Sibley, George Mason University, USA</i>	
<b>Session 9A: Distributed Software Development</b>	
Great Challenges and Opportunities of Distributed Software Development - An Industrial Survey.....	489
<i>Seija Komi-Sirviö, Maarit Tihinen, VTT Electronics, Finland, and</i> <i>Center for Experimental Software Engineering, USA</i>	
GENESIS: A Flexible and Distributed Environment for Cooperative Software Engineering.....	497
<i>Lerina Aversano, Andrea De Lucia, University of Sannio, Italy</i> <i>Matteo Gaeta, Pierluigi Ritrovato, University of Salerno, Italy</i>	

A Distributed Document Management Approach for Workflow Support in the DBSA Architecture.....	503
<i>Andrea Trentini, Francesca Arcelli, Francesco Tisato, University of Milano, Italy</i>	
Integrating Publish/Subscribe into a Mobile Teamwork Support Platform.....	510
<i>S. Chaki, Carnegie Mellon University, USA, P. Fenkam, H. Gall, Technical University Vienna, Austria, S. Jha, University of Wisconsin, USA, E. Kirda, H. Veith, Technical University Vienna, Austria</i>	
A Graph-Oriented Approach to the Description and Implementation of Distributed and Dynamic Software Architecture.....	518
<i>Xiaoxing Ma, Jian Lu, Nanjing University Nanjing, China</i> <i>Jiamong Cao, Alvin T.S. Chan, Hong Kong Polytechnic University, Hong Kong, China</i> <i>Kang Zhang, University of Texas at Dallas, USA</i>	
<b>Session 9B: Spatial Reasoning and Search</b>	
Agent-assisted Distributed Requirements Elicitation and Management.....	526
<i>Chee Fon Chang, Aneesh Krishna and Aditya K. Ghose, University of Wollongong, Australia</i>	
Infotop - A Shared-context Information Workspace.....	534
<i>Ronald Maier, Martin-Luther-University Halle-Wittenberg, Germany</i> <i>Johannes Sametinger, Johannes Kepler University Linz, Austria</i>	
Analysing Web Ontology in Alloy: A Military Case Study.....	542
<i>Jin Song Dong, Jun Sun, Hai Wang, School of Computing, National University of Singapore, Chew Hung Lee, Hian Beng Lee, Defence Science Organisation, Singapore</i>	
Path Set Operations for Clipping of Parts of Web Pages and Information Extraction from Web pages.....	547
<i>Tetsuya SUZUKI, Takehiro TOKUDA, Tokyo Institute of Technology, Japan</i>	
3D Time-pillars: an information access metaphor for the TV Domain.....	555
<i>Fabio Pittarello, Università Ca' Foscari di Venezia, Italy</i>	
<b>Session 10A: Requirements Engineering</b>	
Early Multi-Level Software Architecture Performance Evaluations.....	561
<i>K. Suzanne Barber, University of Texas at Austin, USA</i> <i>Jim Holt, Geoff Baker, Motorola, USA</i>	

Revising Rules to Capture Requirements Traceability Relations: A Machine Learning Approach.....	570
<i>G. Spanoudakis, A. d'Avila Garcez, A. Zisman, City University, UK</i>	
From Requirement Specification to Prototype Execution: a Combination of Multiview Use-Case Driven Methods and Agent-Oriented Techniques.....	578
<i>E. Astesiano, M. Martelli, V. Mascardi and G. Reggio, Universita` di Genova, Italy</i>	
Requirements Negotiation under Incompleteness and Uncertainty.....	586
<i>An Ngo-The, Gunther Ruhe, University of Calgary, Canada</i>	
A Fast Binary Tree Based Face Recognition Algorithm.....	594
<i>Huaqing Li, Shaoyu Wang and Feihu Qi, Shanghai Jiaotong University, China</i>	
<b>Session 10B: Software Testing and Software Education</b>	
Management of the Growing Book as Generalized Objects.....	599
<i>Shuyi Shao and Shi-Kuo Chang, University of Pittsburgh, USA</i>	
Better Analysis of Defect Data at NASA.....	607
<i>Tim Menzies, West Virginia University, USA, Robyn Lutz, Iowa State University, USA Carmen Mikulski, California Institute of Technology, USA</i>	
Working with Extreme Programming in a Software Development Laboratory.....	612
<i>Patricio Letelier, José H. Canós and Emilio A. Sánchez, Polytechnic University of Valencia, Spain</i>	
The Algorithm Tutor.....	616
<i>Eugene Y. Sheng, Northern Illinois University, USA</i>	
<b>Session 11A: Data Mining for SE and KE 1</b>	
A Database-Independent Strategy for Confidence Determination.....	621
<i>Xiaowei Yan, Chengqi Zhang, and Shichao Zhang, University of Technology, Sydney, Australia</i>	
Validative measurement in software engineering: a data mining example.....	626
<i>Ned Chapin, InfoSci Inc., USA Mira Kajko-Mattsson, Stockholm University/Royal Institute of Technology, Sweden</i>	
A Data Mining Approach for Dynamic Software Project Plan Tracking.....	634
<i>Ching-Seh Wu and Dick B. Simmons, Texas A&amp;M University, USA</i>	

**Session 11B: Software Engineering and Decision Support 1**

Using Software Risk Management for Deriving Method Requirements for Risk Mitigation in COTS Assessment and Selection.....	639
<i>Michael Ochs, Fraunhofer Institute IESE, Germany</i>	
Traceability and Decision Capture in Semi-structured Contexts.....	647
<i>Leila Meshkat, Martin Feather, Steven Cornford, California Institute of Technology</i>	
Improved Software Engineering Decision Support Through Automatic Argument Reduction Tools.....	655
<i>Tim Menzies, West Virginia University, James Kiper, Miami University Martin S. Feather, California Institute of Technology</i>	
COO-flow: a Process Technology to Support Cooperative Processes .....	663
<i>D. Grigori, Université Versailles St-Quentin &amp; INRIA, France F. Charoy, C. Godart, Université Henri Poincaré &amp; INRIA, France</i>	
A Holonic Framework for Business Activity Management.....	671
<i>Jun-Jang Jeng (JJ), Steve Buckley, Henry Chang and Jen-Yao Chung, IBM T.J. Watson Research Center, USA</i>	
<b>Session 12A: Data Mining for SE and KE 2</b>	
Mining Access Patterns for Enhancing Navigational Access in Object-Oriented Database.....	679
<i>Chih Lai, Szara Loring, Joe Breuer, University of St. Thomas, USA</i>	
Software Warehouse and Its Management Strategies .....	685
<i>Honghua Dai, Wei Dai, Deakin University, Australia</i>	
An Integrated Framework for Knowledge Management and Discovery of Tropical Cyclone Movements.....	691
<i>James N.K. Liu, Raymond Kwong, Wang Meng and Danny K.Y. Sin, Hong Kong Polytechnic University, China</i>	
Application of Clustering Techniques to Component Architecture Design.....	699
<i>Shu-chuan Lo and Jia-hui Chang, National Taipei University of Technology, Taiwan</i>	
<b>Session 12B: Software Engineering and Decision Support 2</b>	
A Bayesian Approach to Software Testing Evaluation.....	706
<i>Sandro Morasca, Università degli Studi dell Insubria, Italy</i>	

A Tool for Decision Support Implementing OFNWA Approach: A Case Study.....	714
<i>Gerardo Canfora, Luigi Cerulo, Rosa Preziosi, Luigi Troiano, University of Sannio, Italy</i>	
Towards a Framework for Web Sites Quality Evaluation.....	721
<i>Paolo Ciancarini, Rocco Moretti, Dipartimento di Scienze dell'Informazione, Italy</i>	
Conceptual Analysis of Software Structure.....	726
<i>Richard Cole, Thomas Tilley, University of Queensland, Australia</i>	
Reviewers .....	734
Authors Index .....	735

# Towards an Ontology for Software Measurement

Marcela Genero, Francisco Ruiz, Mario Piattini, Félix García and Coral Calero

ALARCOS Research Group

Department of Computer Science

University of Castilla - La Mancha

Paseo de la Universidad, 4

13071 - Ciudad Real - SPAIN

{Marcela.Genero, Francisco.RuizG, Mario.Piattini,Felix.Garcia,Coral.Calero}@uclm.es

## Abstract

As software measurement is a relatively young discipline does not exist yet a common agreement about the concepts and terminology used in this field. We believe that in order to software measurement become a well established engineering discipline it is necessary to start reaching a consensus among software measurement researchers and practitioners about what we are talking about when we refer to "software measurement" or "software measurement process", "measure", "metric", "measurable attribute", "measurable object", etc. Therefore the main focus of this paper is to present a practical and semi-formal ontology for software measurement which was designed using the Unified Modeling Language (UML) as an ontology representation language and an adaptation of the Representation Formalism for Software Engineering Ontologies (REFSENO). Moreover, a software measurement metamodel which captures the ontology has been implemented in the MANTIS environment, a software maintenance engineering environment.

**Keywords:** Software measurement, software measurement process, metrics, ontology, metamodel, software maintenance, REFSENO.

## 1. Introduction

Software measurement suffers from several symptoms of any relatively young discipline [3]. Software measurement is currently in the phase in which terminology, principles, and methods are still being defined and consolidated. In spite of this, nobody can discuss the relevant role that software measurement plays in software engineering. Software measures could be used to build prediction systems for database projects [15], to understand and improve software development and maintenance projects [2], to maintain system quality, highlighting problematic areas [5], and to determine the best ways to help practitioners and researchers in their work [19], etc. Moreover software measures help to assess and institutionalise Software Process Improvement (SPI) in software-intensive organisations. In fact in the most popular initiatives such as CMM, SPICE and CMMi measurement plays a fundamental role. The ISO 9000 standard emphasises

the importance of measures for quality assurance and management.

We believe that a good starting point to contribute to software measurement become a mature discipline is to create a common agreement about the concepts and terminology used in this field. For this purpose it is necessary that software measurement researchers and practitioners reach to a consensus about what we are talking about when we refer to "software measurement" or "software measurement process", "measure", "metric", "measurable attribute", "measurable object", etc.

In order to obtain this goal we decided to build a software measurement ontology, i.e. a common conceptualisation of software measurement domain, where objects, concepts, entities and their relationships were explicitly represented. An ontology represents a certain view of an application domain in which the concepts that live in this domain are defined in an unambiguous and explicit way [6].

Our background in software measurement [4], [7], [9], [20] helped us to detect the main we need to consider, including their relationships and their importance within software measurement.

The software measurement ontology we propose in this paper is part of a "super-ontology" built within the MANTIS environment [23], a software maintenance engineering environment [28].

Moreover, an ontology facilitates knowledge sharing knowledge integration and knowledge reuse. All of these aspects are critical for MANTIS environment, since its goal is to promote the sharing and reusing of information and knowledge.

We believe this ontology in conjunction with other proposals such as Kitchenham et al. [14], Olsina et al. [17] and Briand et al. [3] allow to get and agreement in the building a repository that contains all the knowledge relevant to measurement projects.

The ontology of software measurement we propose has been developed based on three main documents:

- The software measurement process of the ISO 15939 [13].

<sup>1</sup> Vizcaino et al. [28] gives an overview of each of the "sub-ontologies" that compose the "super-ontology" used in the MANTIS Environment, but does not provide details of the software measurement ontology.

- The conceptual model for representing collections of software data provided by Kitchenham et al. [14].
- The software measurement conceptual model proposed by Becker-Kornstaedt and Webby (1999).

- The terminology provided in the ISO 9126 [12].

Therefore, the main focus of this work is to present a practical and semi-formal ontology for software measurement which was designed using UML [16] as an ontology representation language and an adaptation of the Representation Formalism for Software Engineering Ontologies (REFSENO) [25].

Although other sets of techniques exist for developing ontologies [10], [24], [26], [27] they have not been applied to software engineering as REFSENO has (An example of a GQM ontology developed using REFSENO appears in [25]).

REFSENO is an improved adaptation of Methontology [8], [10] which imitates the software life-cycle proposed by the IEEE 1074 standard [11]. REFSENO provides constructs to describe concepts where each concept represents a class of experience items. Besides concepts, its properties (called terminal attributes) and relationships (nonterminal attributes) are represented.

One relevant feature of REFSENO is that incorporates integrity rules such as: cardinalities and ranges of values for attributes, assertions and preconditions.

REFSENO extends the formalism of Osterag, et al. [18] by additional integrity rules and by clearly separating the schema definition and characterisation.

The rest of this paper is organized as follows: The conceptual description of software measurement is presented in Section 2. The proper definition of an ontology for software measurement comes in Section 3. A software measurement metamodel which allows ontology implementation is presented in Section 4. Finally, the last section presents some concluding remarks.

## 2. Conceptual Description of Software Measurement

In this section we present a conceptual description about what "to measure" means in this ontology for software measurement, i.e. the structures and the existing relationships between the information needs of any measurable object (products, projects, processes and services) and its relevant elements (artefacts, activities, resources or agents) which allow the obtaining the information products that satisfy such information needs.

- The selection and definition of metrics suitable for satisfying one information need start with a measurable concept, i.e. one idea of which are the measurable attributes related with the information need and how they are related.
- The value of an attribute of one element can be measured via one or more metrics (also called

measures). Each observation or measurement<sup>2</sup> allows the obtaining the value of an attribute in one instant or point of time for one of the associated metrics. Each metric has an specific measurement unit which belongs to an specific scale. Four types of scales exist: nominal, ordinal, interval and ratio. Each metric also has a default value which is the only value known before performing any observation based on such a metric.

- The observations can be direct or indirect. The direct observations consist of using some of the base metrics associated with one attribute. Each base metric is defined by an specific measurement method (operations which establish a correspondence between the attribute and the scale associated with the metric). The methods can be subjective or objective.

For performing indirect observations we have derived metrics and indicators. The derived metrics are not directly usable to satisfy the information needs, while the indicators are usable to satisfy information needs. The value of a derived metric is obtained applying a measurement function to two or more values of base metrics.

- The indicators can have different accuracy levels. For this reason, the value of an indicator can be quantitative or qualitative. The value of an indicator is obtained applying an analysis model, i.e. an algorithm for combining the values of one or more defined metrics (derived or base metrics) using certain decision criteria.

- One interpretation consists of an explanation, understandable for the stakeholders, about the results of one observation of one indicator.

- One information product is the result of the measurement process which satisfies one information need. It is composed of the collection of suitable interpretations.

This ontology defines concepts as generic as possible in order to achieve the goal of being applicable to different measurement models and measurement methods. In this way, the data representation model proposed by Kitchenham et al. [14] or the conceptual model for the metric domain proposed by Olsina et al. [17] are usable within this ontology.

## 3. An Ontology of Software Measurement

In figure 1 we present the UML class diagram of the ontology for software measurement. We have preferred to use UML class diagrams as representation language instead of the diagrams proposed in the REFSENO methodology, because UML is a widely used and well-known object oriented modeling standard that has recently been proposed as an ontology representation language [29].

<sup>2</sup> We have chosen the nomenclature "observation" instead of measurement for avoiding the confusion with the concept of measurement as a process.

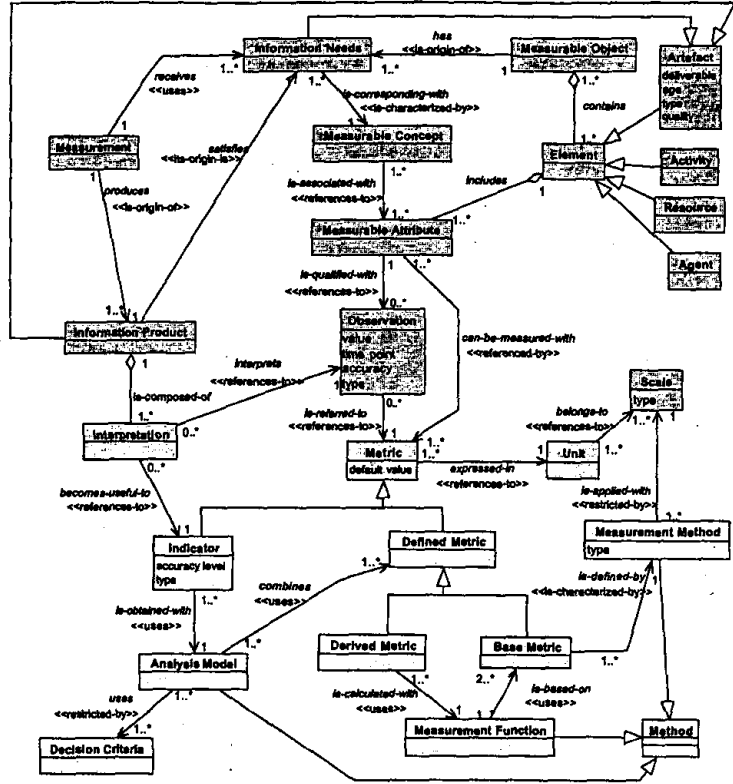


Figure 1. A UML class diagram for the ontology for software measurement

Hereafter the tables of the ontology are presented (an adapted version of REFSENO is used): Table 1 presents the glossary of concepts, table 2 is a table of attributes and finally table 3 presents the relationships.

Concept	Category	Description	Type
Activity	Element	Action that must be performed for achieving the goals of the measurable object.	To describe the task to be done
Agent	Element	Person, organization or software product which play an active role in the realization of a measurable object.	To identify the stakeholders of the measurable object
Analysis Model	Method	Algorithm or calculus that combines one or more metrics defined with certain decision criteria to obtain the value of an indicator and its interpretation.	To establish a way to measure
Artefact	Element	Part of a measurable object that is created or modified by the activities.	To define the structure of the measurable object
Base Metric	Defined Metric	Metric defined based on an attribute and the measurement method for obtaining its value. Synonymous: Direct metric.	To measure
Decision Criteria	Concept	Thresholds values, objectives or patterns used for determining the necessity of an action or for describing the level of goodness of a metric value. It helps to interpret the measurement results.	To measure

Concept	Category	Description	Type
Defined Metric	Metric	Metric for which exists an exact mathematic formula for obtaining its value. It can be a base metric or a derived metric.	To measure
Derived Metric	Defined Metric	Metric defined based on two or more values of base metrics. It captures information about more than one attribute or about the same attribute for more than one element.	To measure
Element	Concept	The elements of one measurable object can be artefacts, activities, resources or agents that can be characterized by measuring some of their attributes.	To manage the different types of elements of a measurable object
Indicator	Metric	Metric that provides an estimation (evaluation) of some determined attributes which respect to certain information needs. An indicator is the basis for making decisions and analyse the measurable object. The value of and indicator can be quantitative or qualitative.	To measure
Information Needs	Artefact	Artefact of document type which describes the necessary information for managing a measurable object.	To manage a measurable object
Information Product	Artefact	Artefact of document type which allows satisfying a certain information need, including the interpretations of the observations of one or more indicators.	To manage a measurable object
Interpretation	Artefact	Artefact of document type which consists of an explanation of the result of an observation of an indicator for satisfying one information need.	To manage a measurable object
Measurable Object	Concept	Object that can be measured, such as projects, processes, products and services. It is an aggregate of elements (artefacts, activities, resources and agents).	
Measurable Concept	Concept	Idea that establishes an abstract relationship between attributes and information needs. Examples: quality, performance, maturity, etc.	To identify objects to be measured
Measurement Function	Method	Algorithm or calculus performed to combine two or more values or base metrics for obtaining the value of a derived metric. The unit and scale of a derived metric depend on which was the measurement function.	To establish the way of measuring
Measurement	Concept	Process which consists of performing the necessary activities for obtaining the information products which satisfies the information needs produced during the management of a measurable object.	To manage a measurable object
Measurement Method	Method	Method which consists of logic sequence of operations used to quantify an attribute respect to a certain scale. Two measurement methods exist: subjective (based on human judgement) and objective (based on numeric rules).	To establish the way of measuring
Measurable Attribute	Concept	Property or characteristic of one element that can be distinguished both qualitatively and quantitatively.	To identify objects to be measured
Method	Concept	Systematic process which includes the steps and heuristics for allowing the realization one or more activities.	To define activities
Metric	Concept	Variable associated to one or more attributes which allows their qualification through observations (assignment of values). It can be a base metric, a derived metric or an indicator. Synonymous: measure.	To measure
Observation	Concept	Action that happens in an instant or a point in time which allows obtaining the value of an attribute referred to a metric.	To measure
Resource	Element	Something of which is not human characteristics which is necessary for performing an activity.	To manage resources
Scale	Concept	Ordered set of values, continuous or discrete, or a set of categories to which a metric for an attribute is associated.	To measure
Unit	Concept	Quantity, defined or adopted by convention, used for comparing quantities of the same type. Only quantities of the same type can be compared directly.	To measure

Table 2. Software measurement ontology: Glossary of concepts

Concept	Attribute	Description	Type
Artefact	Deliverable	Indicates if the artefact must be delivered to a client.	1
	Age	Number of years of existence	1
	Type	Own characteristic of the artefact. For example: document, module, component, file, etc.	1
	Quality	Qualitative measure of the quality.	1
Indicator	Accuracy Level	A quantification of the uncertainty or accuracy inherent to the way of obtaining the value of an indicator.	1
	Type	Own characteristics of the value obtained for an indicator. It can be numeric quantitative or qualitative.	1
Measurement Method	Type	Indicate the own characteristics of the operations used for quantifying an attribute. There exist two types: subjective (quantification influenced by the human judgement) and objectives (quantification based on numeric rules).	1
	Default Value	Initial value that a metric has before any observation of this metric was done.	1
Observation	Precision	Indicator of the precision that a value of an observation has.	1
	Time Point	Moment or instant when the observation is performed.	1
	Type	Indicates if an observation is direct (associated to a base metric) or indirect (associated to a derived metric or to an indicator).	1
	Value	Quantitative or qualitative result assigned to a metric (base, derived or indicator) Synonymous: Data.	1
Scale	Type	Indicates the own characteristics of the relationship between the values of a scale. It can take four values: nominal, ordinal, interval and ratio.	1

Table 3. Software measurement ontology: Table of attributes

Becomes-useful-to	An interpretation provides utility on a concrete necessity to an indicator.
Belongs-to	One measurement unit belong to the set of values of one or more scales (all of them of the same type).
Can-be-measured-with	A measurable attribute can be measured via several metrics.
Combines	One analysis model combines the values of one or more defined metrics (derived or base metrics) for obtaining the value of an indicator.
Contains	A measurable object can contain a collection of elements of different types, such as artefacts, activities, resources and agents.
Expressed-in	The results obtained with a metric are expressed in a measurement unit.
Has	A measurable object has some information needs that are necessary to satisfy its correct management.
Includes	A relevant element of a measurable object includes one or more measurable attributes.
Interprets	An interpretation interprets (explains) the result of an observation.
Is-applied-with	A measurement method is applicable with a specific scale.
Is-associated-with	A measurable concept is associated with one or more measurable attributes.
Is-based-on	The measurement function used to calculate a metric value is based on the value of two or more base metrics.
Is-calculated-with	The value of a derived metric is calculated with a specific measurement function.
Is-qualified-with	A measurable attribute is qualified with one or more observations.
Is-composed-of	An information product is composed of one or more interpretations.
Is-corresponding-with	A concrete information need of a measurable object is corresponding with one measurable concept.
Is-defined-by	A base metric is defined based on the measurement method used to obtain its value.
Is-obtained-with	The value, the accuracy level and the interpretation of an indicator are obtained with an specific analysis model.
Is-referred-to	An observation obtains the value of one attribute referred to a metric.
Produces	The measurement process produces information products as results of its activities.
Receives	The measurement process uses as input of its activities the information needs of a measurable object.
Satisfies	An information product satisfies an information need.
Uses	One analysis model uses one or more decision criteria.

Table 4. Software measurement ontology: Table of relationships

As we mentioned above this software measurement ontology is part of a "super-ontology" built within the MANTIS environment, a software maintenance engineering environment which consists of several sub-ontologies. Therefore it is almost impossible to avoid the duplicity of relationship names if we want to use self-explaining names.

This fact does not offer problems if the relationships that have the same name have the same type of relationships (i.e. they have the same semantic). Table 5 only shows those classes of relationships used in the software measurement ontology (in figure 1 are shown as stereotypes).

Relationships	Class Name	Class Purpose
Characterise-to	Is-characterised-by	An instance of a concept identify the instances of other concept.
Is-origin-of	Its-origin-is	An instance of a concept is originated as consequence and after the existence of other instance of other concept.
Refers-to	References-to	The definition of an instance of a concept it is done in reference to instances of other concept.
Restrict-to	Restricted-by	The possible states of an instance of a concept are limited by instances of other concept.
Uses	Is-used-by	An instance of a concept uses instances of other concept.

Table 5. Tables of classes of relationships (Ruiz, 2003)

#### 4. A Metamodel for Software Measurement

The software measurement ontology described in the previous section in conjunction with others ontologies designed within the MANTIS environment describes the most important knowledge of the software maintenance management. Therefore, in order to be able to take advantage of this ontological knowledge for the software tools it is necessary to have suitable

metamodels [22]. Figure 2 shows the general architecture of metamodeling of the MANTIS environment, within which it is the software measurement metamodel. This metamodel extends the software process metamodel. The software measurement metamodel consists of only one package that it is shown in figure 3.

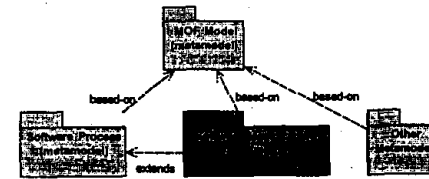


Figure 2. General metamodeling architecture of the MANTIS environment

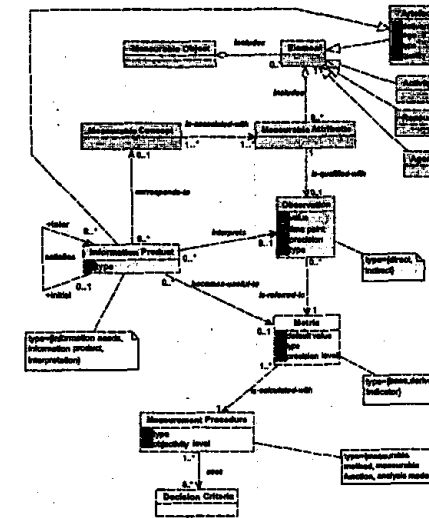


Figure 3. Package of the software measurement metamodel

#### 5. Conclusions

This paper presents a semi-formal ontology for software measurement contributing thus to reach a consensus about the concepts and terminology used in this field. Having a clear and precise ontology was a critical requirement because the ontology was designed with the goal of being implemented inside a software maintenance environment [23]. This gives the ontology a "practical" character.

To develop the ontology, a suitable methodology, REPESEN [25], has been used.

We hope this first proposal of an ontology for software measurement and its corresponding metamodel had a good diffusion within researchers and practitioners. We are aware that this proposal can and must be refined after a discussion process trying to integrate others approaches and ideas suggested by most of the experts in software measurement.

#### Acknowledgements

This research is partially supported by the TAMANSI project (grant number PBC-02-001) financed by the Consejería de Ciencia y Tecnología of the Junta de Comunidades de Castilla-La Mancha and the CALDEA project supported by Subdirección General de Proyectos de Investigación, Ministerio de Ciencia y Tecnología (TIC2000-0024-P4-02).

#### References

- [1] Becker-Kornstaedt, U. and Webby, R. (1999). A Comprehensive Schema Integrating Software Process Modeling and Software Measurement. Fraunhofer IESE - Report No 047.99, August 1999. Available at: [http://www.iese.fhg.de/Publications/iese\\_reports/](http://www.iese.fhg.de/Publications/iese_reports/)
- [2] Briand, L., Morasca, S. and Basili, V. (2002). An operational process for goal-driven definition of measures. IEEE Transactions on Software Engineering, 28(12), 1106-1125.

- [3] Briand, L., Morasca, S., and Basili, V. (1996): Property-Based Software Engineering Measurement, *IEEE Transactions on Software Engineering*, 22(1).
- [4] Calero, C., Piattini, M. and Genero, M. (2001). Empirical validation of referential integrity metrics. In *Information and Software Technology*, 43, 949-957.
- [5] De Champeaux, D. (1997). Object-oriented development process and metrics, Upper Saddle River, Prentice-Hall.
- [6] Deridder, D. (2002). A Concept-Oriented Approach to Support Software Maintenance and Reuse Activities. KBOSS Workshop. ECOOP, Málaga (Spain).
- [7] Diaz, O., Piattini, M. and Calero, C. (2001). Measuring triggering-interaction complexity on active databases. *Information Systems Journal*, 26(1), Elsevier Science.
- [8] Fernández, M., Gómez-Pérez, A. and Juristo, N. (1997). METHONTOLOGY: From Ontological Art Towards Ontological Engineering. AAAI Spring Symposium. University of Stanford; Palo Alto, California (Estados Unidos), 33-40.
- [9] Genero, M., Olivas, J., Piattini, M. and Romero, F. (2001). Using metrics to predict OO information systems maintainability, CAISE 2001, Lecture Notes in Computer Science (LCNS) 2068, Dittrich, K., Geppert, A. And Norrie, M.C. (eds.) Springer-Verlag, 388-401.
- [10] Gómez-Pérez, A. (1998). Knowledge sharing and reuse. In Jay Liebowitz, editor, *The Handbook of Applied Expert Systems*. CRC Press.
- [11] IEEE (1995). STD 1074-1995. IEEE Standard for Developing Software Life Cycle Processes.
- [12] ISO 9126 (2001). Software Product Evaluation-Quality Characteristics and Guidelines for their Use, ISO/IEC Standard 9126. Geneva.
- [13] ISO/IEC (2002). FDIS 15939. Software Engineering - Software Measurement Process (draft).
- [14] Kitchenham, B., Hughes, R. and Linkman, S. (2001). Modeling Software Measurement Data. *IEEE Transactions on Software Engineering*, 27(9), 788-804.
- [15] MacDonell, S., Shepperd, M. and Sallis, P. (1997): Metrics for Database Systems. An Empirical Study, Proc. Fourth International Software Metrics Symposium - Metrics'97, Albuquerque. IEEE Computer Society, 99-107.
- [16] Object Management Group. (1999). UML Revision Task Force, OMG Unified Modeling Language Specification, v. 1.3, document ad/99-06-08.
- [17] Olsina, L., Bertoa, M., Lafuente, G., Martín, M., Katrib, M., and Vallecillo, A. (2002). Un Marco Conceptual para la Definición y Explotación de Métricas de Calidad. VII Jornadas de Ingeniería del Software y Bases de Datos (IJSBD'2002). El Escorial, Madrid, 189-199.
- [18] Ostertag, E., Hendlar, R., Prieto-Díaz and C. Braun. (1992). Computing similarity in a reuse library system: An AI-based approach. *ACM Transactions on Software Engineering and Methodology*, 1(3), 205-228.
- [19] Pfleeger, S. (1997). Assessing Software Measurement, *IEEE Software*, 25-26.
- [20] Piattini, M., Genero, M. and Jiménez, L. (2001). A metric-based approach for predicting conceptual data models maintainability. *International Journal of Software Engineering and Knowledge Engineering*, 11(6), 703-729.
- [21] Ruiz, F. (2003). MANTIS. Definición de un entorno para la Gestión del Mantenimiento de Software. Ph.D. Thesis, University of Castilla-La Mancha (In Spanish).
- [22] Ruiz, F., Piattini, M., and Polo, M. (2001). An Conceptual Architecture Proposal for Software Maintenance. 13th International Symposium on System Integration (ISSI'01). Baden-Baden (Germany), VIII, 1-8.
- [23] Ruiz, F., Piattini, M. and Polo, M. (2002). An Integrated Environment for Managing Software Maintenance Projects. In "The Guide to IT Service Management, volume 1", chapter 31, Addison-Wesley, 567-588
- [24] Staab, S., Schurr, P. and Sure, Y. (2001). Knowledge Processes and Ontologies. *IEEE Intelligent Systems*, 16(1), 26-34.
- [25] Tautz, C. and Von Wangenheim, C. (1998): REFSENO: A Representation Formalism for Software Engineering Ontologies. Fraunhofer IESE-Report No. 015.98/E, version 1.1, October 20.
- [26] Uschold, M. and Gruninger, M. (1996). Ontologies: Principles, methods, and applications. *The Knowledge Engineering Review* 11(2), 93-136.
- [27] Van Heijst, G., Falasconi, S. and Abu-Hanna A., Schreiber, T. and Stefanelli, M. (1995). A case study in ontology library construction. *Artificial Intelligence in Medicine* 7, 227-255.
- [28] Vizcaino, A., Ruiz, F. and Piattini, M. (2003). An ontology for software maintenance. Submitted to the *Journal of Software Engineering and Knowledge Engineering*.
- [29] Wang, X. and Chan, C.W. (2001). Ontology Modeling Using UML. 7th International Conference on Object Oriented Information Systems Conference (OOIS'2001), 59-68.