

MODEL DRIVEN ENGINEERING LANGUAGES AND SYSTEMS

**14th International Conference, MODELS 2011
Wellington, New Zealand, October 16-21, 2011
Proceedings**



START

SEARCH

LNCS 6981

**AUTHOR
INDEX**



Springer, Tiergartenstraße 17, 69121 Heidelberg, Germany
www.springer.com

Lecture Notes in
Computer Science

LNCS

LNAI

LNBI

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Jon Whittle Tony Clark Thomas Kühne (Eds.)

Model Driven Engineering Languages and Systems

14th International Conference, MODELS 2011
Wellington, New Zealand, October 16-21, 2011
Proceedings



Springer

Volume Editors

Jon Whittle

Lancaster University, School of Computing and Communications
InfoLab21, South Drive, Lancaster LA1 4WA, UK
E-mail: j.n.whittle@lancaster.ac.uk

Tony Clark

Middlesex University, School of Engineering and Information Sciences
The Burroughs, Hendon, London NW4 4BT, UK
E-mail: t.n.clark@mdx.ac.uk

Thomas Kühne

Victoria University, School of Engineering and Computer Science
P.O. Box 600, Wellington 6140, New Zealand
E-mail: thomas.kuehne@ecs.vuw.ac.nz

ISSN 0302-9743

e-ISSN 1611-3349

ISBN 978-3-642-24484-1

e-ISBN 978-3-642-24485-8

DOI 10.1007/978-3-642-24485-8

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011937287

CR Subject Classification (1998): D.2, D.3, K.6.3, D.2.9, F.3.3, D.1, D.2.2

LNCS Sublibrary: SL 2 – Programming and Software Engineering

© Springer-Verlag Berlin Heidelberg 2011

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

For the past 14 years, the MODELS conference has been the premier venue for the exchange of innovative ideas and experiences of model-based approaches in the development of complex systems. MODELS is universally recognized as one of the top conferences in software engineering research and is a highly selective conference, with an acceptance rate averaging 20% in recent years. The conference series covers all aspects of model-based development for software and systems engineering, including modeling languages, methods, tools, and their applications.

Research in software and system modeling is now a relatively mature field. Like any mature field, however, it can be a good idea to encourage fresh thinking. Whilst not wishing to reduce the importance of solid incremental research, the conference this year asked participants to think ahead to what modeling would be like a decade hence. For this reason, the Program Chairs selected *Modeling in 2020* as the theme for MODELS 2011. The theme was chosen to encourage new perspectives about the future role of modeling in complex systems engineering. As part of this effort, the conference solicited, for the first time, a new category of research papers—vision papers—that presented “outside the box” thinking. This category was introduced to encourage the submission of papers with new ideas that would take the community beyond its normal boundaries.

As part of the effort to encourage fresh perspectives, the conference invited three outstanding keynote speakers this year, two of which were from outside the software modeling domain.

Marian Petre is a Professor of Computing at the Open University in the UK. She is well known for her work considering software from a ‘design studies’ perspective and describes her role to ‘pick the brains of experts’ in studying how leading professional software developers reason about, represent, and communicate designs. Marian’s keynote reported on insights from many years of empirical studies of expert software designers.

The conference welcomed its first ever Academy Award winning speaker this year. Mark Sagar is Special Projects Supervisor at Weta Digital. He has developed technologies for interactive applications and for feature films and has won two consecutive Scientific & Engineering Academy Awards for his pioneering work in facial motion capture and realistic relighting of computer generated faces. He has specialized in bringing computer generated faces to life in some of Hollywood’s biggest blockbusters including “Avatar” and “King Kong”. Mark’s fascinating talk focused on creating models for simulating the face.

MODELS was also very lucky to welcome Wolfram Schulte as a keynote speaker. Wolfram is a principal researcher and the founding manager of Microsoft’s Research in Software Engineering (RiSE) team in Redmond,

Washington. In his talk, Wolfram presented Formula, a new formal specification language and toolset for describing, transforming and analyzing meta-models and instance models.

MODELS 2011 continued its strong tradition of soliciting both research-oriented papers (the Foundations Track) and practice-oriented papers (the Applications Track). The Foundations Track received 167 full paper submissions, of which 34 were finally selected for presentation by the program committee, giving an acceptance rate of 20%. Out of these, 3 papers were vision papers, selected out of a total of 20 vision paper submissions (15% acceptance rate). The Applications Track was particularly healthy this year: the program committee chose 13 out of 27 paper submissions (48% acceptance rate). In addition, two papers that were originally submitted to the Foundations Track were transferred and accepted into the Applications Track.

The Program Chairs would like to thank all those who submitted papers, as well as those who submitted proposals for workshops and tutorials. We would also like to express our gratitude to the many volunteers who contributed to the success of the conference, including organizers of the Educators' Symposium and Doctoral Symposium. Special thanks are due to Richard van de Stadt for his support of CyberChairPRO, the conference management system used for MODELS 2011. We thank our sponsors, ACM and IEEE, and host, the Victoria University of Wellington. Last, but certainly not least, we give special thanks to the Program Committee and other external reviewers for all their hard work in reviewing and discussing papers.

October 2011

Jon Whittle
Tony Clark
Thomas Kühne

Organization

General Chair

Thomas Kühne
Victoria University of Wellington,
New Zealand

Local Chairs

Stuart Marshall
Victoria University of Wellington,
New Zealand

Hui Ma
Victoria University of Wellington,
New Zealand

Program Chairs

Jon Whittle
Lancaster University, UK

(Foundations Track)
Tony Clark
Middlesex University, UK
(Applications Track)

Publicity Chair

Werner Heijstek
Leiden University, The Netherlands

Workshop Chair

Jörg Kienzle
McGill University, Canada

Tutorial Chair

Vasco Amaral
Universidade Nova de Lisboa, Portugal

Panel Chair

Colin Atkinson
Universität Mannheim, Germany

Doctoral Symposium Chairs

Jörg Evermann
Memorial University, Canada
Ivan Porres
Åbo Akademi University, Finland

Educators' Symposium Chairs

Marion Brandsteidl
Andreas Winter

Technische Universität Wien, Austria
Carl von Ossietzky University Oldenburg,
Germany

Program Committee: Foundations Track

Daniel Amyot
Colin Atkinson
Joanne M. Atlee
Don Batory
Benoit Baudry
Paulo Borba
Ruth Breu
Lionel Briand

University of Ottawa, Canada
University of Mannheim, Germany
University of Waterloo, Canada
UT Austin, USA
INRIA, France
Federal University of Pernambuco, Brazil
University of Innsbruck, Austria
Simula Research Laboratory and University of
Oslo, Norway

Jean-Michel Bruel
Jordi Cabot
Michel Chaudron
Marsha Chechik
Jane Cleland-Huang
Krzysztof Czarnecki
Juergen Dingel
Alexander Egyed
Gregor Engels
Rik Eshuis

University of Toulouse, France
INRIA / Ecole des Mines de Nantes, France
Leiden University, The Netherlands
University of Toronto, Canada
DePaul University, USA
University of Waterloo, Canada
Queens University, Canada
Johannes Kepler University, Austria
University of Paderborn, Germany
Eindhoven University of Technology,
The Netherlands

Robert France
Sébastien Gérard
Holger Giese

Colorado State University, USA
CEA LIST, France
Hasso Plattner Institute, University of
Potsdam, Germany

Martin Glinz
Martin Gogolla
Aniruddha Gokhale
Jeff Gray
John Grundy
Mark Harman

Universität Zürich, Switzerland
University of Bremen, Germany
Vanderbilt University, USA
University of Alabama, USA
Swinburne University of Technology, Australia
CREST Centre, University College London,
UK

Øystein Haugen
Mats Heimdahl
Patrick Heymans
John Howse
Zhenjiang Hu
Heinrich Hußmann

SINTEF, Norway
University of Minnesota, USA
University of Namur, Belgium
University of Brighton, UK
National Institute of Informatics, Japan
Ludwig-Maximilians-Universität München,
Germany

Paola Inverardi	University of L'Aquila, Italy
Jan Jürjens	TU Dortmund and Fraunhofer ISST, Germany
Shmuel Katz	Technion - Israel Institute of Technology, Israel
Gerti Kappel	Vienna University of Technology, Austria
Gabor Karsai	Vanderbilt University, USA
Jörg Kienzle	McGill University, Canada
Ingolf Krüger	UCSD, USA
Thomas Kühne	Victoria University of Wellington, New Zealand
Jochen Küster	IBM Research - Zurich, Switzerland
Michael Lawley	CSIRO, Australia
Timothy C. Lethbridge	University of Ottawa, Canada
Hong Mei	Peking University, China
Tom Mens	Université de Mons, Belgium
Ana Moreira	Universidade Nova de Lisboa, Portugal
Pierre-Alain Muller	Université de Haute-Alsace, France
Richard Paige	University of York, UK
Mauro Pezzè	University of Lugano, Switzerland and Università degli Studi di Milano Bicocca, Italy
Dorina C. Petriu	Carleton University, Canada
Alfonso Pierantonio	University of L'Aquila, Italy
Ivan Porres	Åbo Akademi University, Finland
Nicolas Rouquette	NASA JPL, USA
Bernhard Rumpe	RWTH Aachen University, Germany
Andy Schürr	Technische Universität Darmstadt, Germany
Bran Selic	Malina Software Corp., Canada
Miroslaw Staron	University of Gothenburg and Chalmers, Sweden
Perdita Stevens	University of Edinburgh, UK
Paul Strooper	University of Queensland, Australia
Kevin Sullivan	University of Virginia, USA
Dániel Varró	Budapest University of Technology and Economics, Hungary
Eelco Visser	Delft University of Technology, The Netherlands
Liming Zhu	NICTA, Australia
Steffen Zschaler	King's College, London, UK

Program Committee: Applications Track

Patrick Albert	IBM CAS France, France
Robert Baillargeon	Sodius, USA
Edward J. Barkmeyer	National Institute of Standards & Technology, USA
Balbir Barn	University of Middlesex, UK
Jorn Bettin	Sofismo, Switzerland
Mariano Belaunde	Orange Labs, France
Rao Bhaskar	Motorola, India
Behzad Bordbar	University of Birmingham, UK
William Cook	University of Texas at Austin, USA
Diarmuid Corcoran	Ericsson Software Research, Sweden
Huascar Espinoza	TECNALIA-European Software Institute, Spain
Geri Georg	Colorado State University, USA
Jack Greenfield	Microsoft Corporation, USA
Pavel Hruby	CSC, Denmark
Narendra Jussien	Ecole des Mines de Nantes, France
Vinay Kulkarni	Tata Consultancy Services, India
Stephen J. Mellor	UK
Dragan Milicev	University of Belgrade, Serbia
Hiroshi Miyazaki	Fujitsu, Japan
Juan Carlos Molina	CARE Technologies, S.A., Spain
Udaeta	
Syed Salman Qadri	The Mathworks, Inc., USA
Ina Schieferdecker	TU Berlin/Fraunhofer FOKUS, Germany
Bran Selic	Malina Software Corp., Canada
Richard Soley	Object Management Group, USA
Ingo Stürmer	Model Engineering Solutions GmbH, Germany
Jun Sun	Singapore University of Technology and Design, Singapore
François Terrier	CEA-LIST, France
Laurence Tratt	Middlesex University, UK
Michael von der Beeck	BMW Group, Germany
Thomas Weigert	Missouri University of Science and Technology, USA
Frank Weil	UniqueSoft, USA
Ed Willink	Eclipse Modeling Project, UK

Steering Committee

Geri Georg (Chair)	Thomas Baar
Gregor Engels (Vice Chair)	Jean Bézivin

Lionel Briand
 Jean-Michel Bruel
 Krzysztof Czarnecki
 Matthew Dwyer (ACM SIGSOFT)
 Øystein Haugen
 Heinrich Hussmann
 Stuart Kent
 Thomas Kühne
 Pierre-Alain Muller

Oscar Nierstrasz
 Dorina Petriu
 Rob Pettit
 Gianna Reggio
 Doug Schmidt
 Andy Schürr
 Steve Seidman (IEEE)
 Jon Whittle

Sponsors

ACM (<http://www.acm.org>)
 IEEE (<http://www.ieee.org>)
 Victoria University of Wellington

Additional Reviewers

Ebrahim Abbasi
 Mauricio Alferez
 Shaukat Ali
 Mohamed Almorsy
 Anthony Anjorin
 Michal Antkiewicz
 Kazuyuki Asada
 Marco Autili
 Iman Avazpour
 Fabian Büttner
 Omar Badreddin
 Prateek Bahri
 Kacper Bak
 Jan-Christopher Bals
 Olivier Barais
 Jorge Barreiros
 Saeed Ahmadi Behnam
 Gábor Bergmann
 Jason Biatek
 Dénes Bisztray
 Quentin Boucher
 John Brondum
 Petra Brosch
 Jens Brüning
 Jim Burton
 Peter Chapman

Eya Ben Charrada
 Hyun Cho
 Fabian Christ
 Antonio Cicchetti
 Harald Cichos
 Robert Clariso
 Andreas Classen
 Benoit Combemale
 Arnaud Cuccuru
 Duc-Hanh Dang
 Sylvain Degrandart
 Barry Demchak
 Zinovy Diskin
 Didier Donsez
 Dirk Draheim
 Sophie Ebersold
 Eban Escott
 Ramin Etemaadi
 Claudiu Farcas
 Emilia Farcas
 Andrew Fish
 Franck Fleurey
 Frédéric Fondement
 François Fouquet
 Miguel Garzón
 Gregory Gay

Nicolas Genon
Christian Gerth
Achraf Ghabi
Sepideh Ghanavati
Javier Gonzalez-Huerta
Danny Groenewegen
Tim Gülke
Lars Hamann
Brahim Hamid
Ali Hamie
Stefan Hanenberg
Ábel Hegedüs
Werner Heijstek
Zef Hemel
Soichiro Hidaka
Stephan Hildebrandt
Ákos Horváth
Arnaud Hubaux
Javier Gonzalez Huerta
Ferosh Jacob
Cédric Jeanneret
Adam Jensen
Rim Jnidi
Martin Johansen
Maartje de Jonge
Lennart C.L. Kats
Ali Hanzala Khan
Felix Klar
Natallia Kokash
Dimitrios S. Kolovos
Jens Krinke
Amber Krug
Mirco Kuhlmann
Thomas Kurpick
Jan-Christoph Küster
Leen Lambers
Philip Langer
Marius Lauder
Hervé Leblanc
Jaejoon Lee
Philipp Liegl
Qichao Liu
Roberto Lopez-Herrejon
Alexander De Luca
Torbjorn Lundkvist

Martin Mahaux
Nikos Matragkas
Kazutaka Matsuda
Max-Emanuel Maurer
Dieter Mayrhofer
Massimiliano Menarini
Marjan Mernik
Gunter Mussbacher
Benjamin Nagel
Stefan Neumann
Florian Noyrit
Hafees Osman
Sebastian Pape
Lars Patzina
Cedric Peeters
Christian Percebois
Gilles Perrouin
Hung Tuan Pham
Claas Pinkernell
Niusha Hakimi Pour
Alireza Pourshahid
Jorge Pinna Puissant
István Ráth
Aleks Radjenovic
Shekoufeh Kolahdouz Rahimi
Irum Rauf
Ruth Raventos
Alexander Reder
Holger Rendel
Márcio Ribeiro
Louis Rose
Rick Salay
Hendrik Schreiber
Andreas Seibel
Ed Seidewitz
Sagar Sen
Cristina Gómez Seoane
Filippo Seracini
Norbert Seyff
Azalia Shamsaei
Steven She
Christian Soltenborn
Gem Stapleton
Jim Steel
Reinhard Stoiber

Ragnhild Van Der Straeten
Espen Suenson
Andreas Svendsen
Jörn Guy Süß
Robert Tairas
Leopoldo Teixeira
Laurent Thiry
Massimo Tisi
Társis Toledo
Dragos Truscan
Hagen Völzer
Gergely Varró
Sander Vermolen
Thomas Vogel
Steven Völkel
Guido Wachsmuth
Fernando Wanderley
Bo Wang

Andrzej Wasowski
Ingo Weisemöller
Sven Wenzel
Magdalena Widl
Konrad Wieland
James R. Williams
Manuel Wimmer
Dustin Wüest
Sebastian Wätzoldt
Yingfei Xiong
Xiwei Xu
Eran Yahav
Li Yi
Tao Yue
Xiang Zhang
Xiaorui Zhang
Celal Ziftci
Olaf Zimmermann

Table of Contents

Keynote 1

The Value in Muddling Around Modelling (Abstract)	1
<i>Marian Petre</i>	

Model Transformations 1

Towards Quality Driven Exploration of Model Transformation Spaces . . .	2
<i>Mauro Luigi Drago, Carlo Ghezzi, and Raffaella Mirandola</i>	
Automated Model-to-Metamodel Transformations Based on the Concepts of Deep Instantiation	17
<i>Gerd Kainz, Christian Buckl, and Alois Knoll</i>	
Lazy Execution of Model-to-Model Transformations	32
<i>Massimo Tisi, Salvador Martínez, Frédéric Jouault, and Jordi Cabot</i>	

Model Complexity

Measuring UML Models Using Metrics Defined in OCL within the SQUAM Framework	47
<i>Joanna Chimiak-Opoka</i>	
Modeling Model Slicers	62
<i>Arnaud Blouin, Benoît Combemale, Benoit Baudry, and Olivier Beaudoux</i>	
Morsa: A Scalable Approach for Persisting and Accessing Large Models	77
<i>Javier Espinazo Pagán, Jesús Sánchez Cuadrado, and Jesús García Molina</i>	

Aspect-Oriented Modeling

Expressing Aspectual Interactions in Design: Experiences in the Slot Machine Domain	93
<i>Johan Fabry, Arturo Zambrano, and Silvia Gordillo</i>	
An Industrial Application of Robustness Testing Using Aspect-Oriented Modeling, UML/MARTE, and Search Algorithms	108
<i>Shaukat Ali, Lionel C. Briand, Andrea Arcuri, and Suneth Walawege</i>	

Aspect-Oriented Modelling for Distributed Systems 123
Wisam Al Abed and Jörg Kienzle

Analysis and Comprehension of Models

A Precise Style for Business Process Modelling: Results from Two
 Controlled Experiments 138
*Gianna Reggio, Filippo Ricca, Giuseppe Scanniello,
 Francesco Di Cerbo, and Gabriella Doderò*

Semantically Configurable Consistency Analysis for Class and Object
 Diagrams 153
Shahar Maoz, Jan Oliver Ringert, and Bernhard Rumpe

Identifying the Weaknesses of UML Class Diagrams during Data Model
 Comprehension 168
*Gabriele Bavota, Carmine Gravino, Rocco Oliveto,
 Andrea De Lucia, Genoveffa Tortora, Marcela Genero, and
 José Antonio Cruz-Lemus*

Domain-Specific Modeling

Engineering Android Applications Based on UML Activities 183
Frank Alexander Kraemer

Domain-Specific Model Transformation in Building Quantity
 Take-Off 198
Jim Steel and Robin Drogemüller

Improving Scalability and Maintenance of Software for
 High-Performance Scientific Computing by Combining MDE
 and Frameworks 213
Marc Palyart, David Lugato, Ileana Ober, and Jean-Michel Briel

Models for Embedded Systems

A Critical Review of Applied MDA for Embedded Devices: Identification
 of Problem Classes and Discussing Porting Efforts in Practice 228
Michael Lettner, Michael Tschernuth, and Rene Mayrhofer

Designing Heterogeneous Component Based Systems: Evaluation of
 MARTE Standard and Enhancement Proposal 243
Ali Koudri, Arnaud Cuccuru, Sebastien Gerard, and François Terrier

Semantic Clone Detection for Model-Based Development of Embedded
 Systems 258
Bakr Al-Batran, Bernhard Schätz, and Benjamin Hummel

Model Synchronization

Instant and Incremental QVT Transformation for Runtime Models	273
<i>Hui Song, Gang Huang, Franck Chauvel, Wei Zhang, Yanchun Sun, Weizhong Shao, and Hong Mei</i>	
Service-Oriented Architecture Modeling: Bridging the Gap between Structure and Behavior	289
<i>Mickael Clavreul, Sébastien Mosser, Mireille Blay-Fornarino, and Robert B. France</i>	
From State- to Delta-Based Bidirectional Model Transformations: The Symmetric Case	304
<i>Zinovy Diskin, Yingfei Xiong, Krzysztof Czarnecki, Hartmut Ehrig, Frank Hermann, and Fernando Orejas</i>	

Model-Based Resource Management

Enforcing S&D Pattern Design in RCES with Modeling and Formal Approaches	319
<i>Brahim Hamid, Sigrid Gürgens, Christophe Jowray, and Nicolas Desnos</i>	
A Model-Based and Automated Approach to Size Estimation of Embedded Software Components	334
<i>Kenneth Lind and Rogardt Heldal</i>	
MDE to Manage Communications with and between Resource-Constrained Systems	349
<i>Franck Fleurey, Brice Morin, Arnor Solberg, and Olivier Barais</i>	

Analysis of Class Diagrams

Diagram Definition: A Case Study with the UML Class Diagram	364
<i>Maged Elaasar and Yvan Labiche</i>	
Reducing Multiplicities in Class Diagrams	379
<i>Ingo Feinerer, Gernot Salzer, and Tanja Sisel</i>	

Keynote 2

Creating Models for Simulating the Face (Abstract)	394
<i>Mark Sagar</i>	

Verification and Validation 1

EUnit: A Unit Testing Framework for Model Management Tasks	395
<i>Antonio García-Domínguez, Dimitrios S. Kolovos, Louis M. Rose, Richard F. Paige, and Inmaculada Medina-Bulo</i>	
Verifying UML-RT Protocol Conformance Using Model Checking	410
<i>Yann Moffett, Alain Beaulieu, and Juergen Dingel</i>	
Model-Based Coverage-Driven Test Suite Generation for Software Product Lines	425
<i>Harald Cichos, Sebastian Oster, Malte Lochau, and Andy Schürr</i>	

Refactoring Models

Constraint-Based Model Refactoring	440
<i>Friedrich Steimann</i>	
Supporting Design Model Refactoring for Improving Class Responsibility Assignment	455
<i>Motohiro Akiyama, Shinpei Hayashi, Takashi Kobayashi, and Motoshi Saeki</i>	

Modeling Visions

Vision Paper: The Essence of Structural Models	470
<i>Dmitrijs Zaparanuks and Matthias Hauswirth</i>	
Vision Paper: Towards Model-Based Energy Testing	480
<i>Claas Wilke, Sebastian Götz, Jan Reimann, and Uwe Aßmann</i>	
Vision Paper: Make a Difference! (Semantically)	490
<i>Uli Fahrenberg, Axel Legay, and Andrzej Wąsowski</i>	

Logics and Modeling

Automatic Derivation of Utility Functions for Monitoring Software Requirements	501
<i>Andres J. Ramirez and Betty H.C. Cheng</i>	
Logic-Based Model-Level Software Development with F-OML	517
<i>Mira Balaban and Michael Kifer</i>	
Formal Verification of QVT Transformations for Code Generation	533
<i>Kurt Stenzel, Nina Moebius, and Wolfgang Reif</i>	

Development Methods

Model-Based (Mechanical) Product Design	548
<i>Mehdi Iraqi-Houssaini, Mathias Kleiner, and Lionel Roucoules</i>	
Applying a Model-Based Approach to IT Systems Development Using SysML Extension	563
<i>Sayaka Izukura, Kazuo Yanoo, Takao Osaki, Hiroshi Sakaki, Daichi Kimura, and Jianwen Xiang</i>	
Early Experience with Agile Methodology in a Model-Driven Approach	578
<i>Vinay Kulkarni, Souvik Barat, and Uday Ramteerthkar</i>	

Keynote 3

Finding Models in Model-Based Development (Abstract)	591
<i>Wolfram Schulte and Ethan K. Jackson</i>	

Model Transformations 2

CD2Alloy: Class Diagrams Analysis Using Alloy Revisited	592
<i>Shahar Maoz, Jan Oliver Ringert, and Bernhard Rumpe</i>	
Model-Driven Engineering and Optimizing Compilers: A Bridge Too Far?	608
<i>Antoine Floch, Tomofumi Yuki, Clement Guy, Steven Derrien, Benoit Combemale, Sanjay Rajopadhye, and Robert B. France</i>	
Towards a General Composition Semantics for Rule-Based Model Transformation	623
<i>Dennis Wagelaar, Massimo Tisi, Jordi Cabot, and Frédéric Jouault</i>	

Verification and Validation 2

Properties of Realistic Feature Models Make Combinatorial Testing of Product Lines Feasible	638
<i>Martin Fagereng Johansen, Øystein Haugen, and Franck Fleurey</i>	
Reasoning about Metamodeling with Formal Specifications and Automatic Proofs	653
<i>Ethan K. Jackson, Tihamér Levendovszky, and Daniel Balasubramanian</i>	
Correctness of Model Synchronization Based on Triple Graph Grammars	668
<i>Frank Hermann, Hartmut Ehrig, Fernando Orejas, Krzysztof Czarnecki, Zinovy Diskin, and Yingfei Xiong</i>	

Model Integration and Collaboration

A Toolchain for the Detection of Structural and Behavioral Latent System Properties	683
<i>Adam C. Jensen, Betty H.C. Cheng, Heather J. Goldsby, and Edward C. Nelson</i>	
Defining MARTE's VSL as an Extension of Alf	699
<i>Arnaud Cuccuru, Sébastien Gérard, and François Terrier</i>	
Using Delta Model for Collaborative Work of Industrial Large-Scaled E/E Architecture Models	714
<i>Rixin Zhang and Ajay Krishnan</i>	
Author Index	729

Identifying the Weaknesses of UML Class Diagrams during Data Model Comprehension

Gabriele Bavota¹, Carmine Gravino¹, Rocco Oliveto², Andrea De Lucia¹,
Genoveffa Tortora¹, Marcela Genero³, and José Antonio Cruz-Lemus³

¹ Software Engineering Lab, University of Salerno, Fisciano (SA), Italy
{gbavota,gravino,adelucia,tortora}@unisa.it

² STAT Department, University of Molise, Pesche (IS), Italy
rocco.oliveto@unimol.it

³ Dep. of Technologies and Information Systems, University of Castilla, La Mancha
{marcela.genero,joseantonio.cruz}@uclm.es

Abstract. In this paper we present an experiment and two replications aimed at comparing the support provided by ER and UML class diagrams during comprehension activities by focusing on the single building blocks of the two notations. This kind of analysis can be used to identify weakness in a notation and/or justify the need of preferring ER or UML for data modeling. The results reveal that UML class diagrams are generally more comprehensible than ER diagrams, even if the former has some weaknesses related to three building blocks, i.e., multi-value attribute, composite attribute, and weak entity. These findings suggest that a UML class diagram extension should be considered to overcome these weaknesses and improve the comprehensibility of the notation.

1 Introduction

A data model is a set of concepts that can be used to describe both the structure of and the operations on a database [1]. It represents the output of data modeling (or conceptual design), an activity that aims at creating a conceptual schema in a diagrammatic form and facilitating the communication between developers and users [1]. Understanding and interpreting data models represents a fundamental activity from the earliest stages of software development, e.g., requirement analysis. Thus, a comprehensive notation is really desirable to avoid misunderstanding that can lead to the introduction of errors very expensive to remove in the later phases of the software development. A comprehensive notation is also desirable during software maintenance, since it facilitates the comprehension activities that have to be performed to understand the design of the system before the analysis and the implementation of a change request.

Entity-Relationship (ER) and its extensions are the most used notations for database conceptual modeling and still remains the *de facto* standard [1]. The success of the Object-Oriented (OO) approach for software development has encouraged the use of this approach also for database modeling [2]. In particular, UML class diagrams can be used to represent the conceptual schema of the whole

software system, so the same notation can be used to model the functionality of the system as well as to represent its data. The structural constructs of the UML class diagram which represents the data structure is somewhat equivalent to Extended ER (EER) representation (e.g., object classes considered equivalent to entity and relationship types). The functionality is represented through “methods” that are attached to the object classes. However, while UML is becoming a *de facto* standard for the analysis and design of software systems, it is not exploited with the same success for modeling databases. Indeed, nowadays ER remains the most used notation to model databases and in some cases it complements UML in the design of software systems. A recent survey also indicated that in some cases both ER and UML class diagrams are employed to represent the same database [3]. Such behaviors might be the trigger for possible problems during the evolution of the data models. More effort is required to maintain the models and their implementation up-to-date, since out-of-data models can generate inconsistency and misunderstanding during software maintenance and evolution. All these considerations lead researchers to empirically compare the ER and UML diagrams to show the actual benefits given by one notation as compared to the other [3,4]. The results achieved in all these studies indicate that the support given by UML class diagrams in comprehension tasks is at least equal (and in some cases higher than) the support given by ER diagrams. However, a qualitative and quantitative analysis concerning the identification of the graphical elements of one notation that are more comprehensible than the corresponding element in the other notation is still missing (this kind of analysis is quantitatively performed in [2] during the comparison of EER and OO models). Such an analysis is vital to provide insight on why UML class diagrams are better than ER diagrams or *vice versa* and highlight strengths and limitations of the two notations. This kind of analysis can be used to (i) justify the need of preferring ER or UML class diagrams for data modeling; or (ii) identify weakness in a notation that could be overcome to improve its comprehensibility.

In this paper we aim at bridging this gap presenting the results of a controlled experiment and two replications to deeply analyze the support given by ER and UML class diagrams during the comprehension of data models. The experiments aimed at performing a fine-grained analysis to (quantitatively and qualitatively) compare the single building blocks, i.e., Entity, Primary Key/ID, Composite Attribute, Multi-value Attribute, Recursive relationship, Relationship cardinality, Ternary relationship, Generalization IS-A, Weak entity, M:N relationship, of the two notations. The experiment and its replications involved 156 students of the university of Salerno (Italy) with different academic background represented by fresher, bachelor, and master students.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 provides details of the design of the experiment and presents the results achieved while Section 4 discuss the possible threats to validity. Concluding remarks and directions for future work are given in Section 5.

2 Related Work

In the last two decades some papers have analysed, through controlled experiments, empirical studies, or surveys, graphical notations supporting the software development process.

To the best of our knowledge only four papers compare the ER notation, or its extensions, and Objected-Oriented (OO) models [5], [2], [6], [7]. In particular, Shoval and Shiran [5] compare Extended ER (EER) and OO data models from the point of view of design quality, where quality is measured in terms of correctness of the produced models, time to completely perform the design task, and designers' opinions. The goal of our empirical investigation is different, since we compare ER and UML diagrams from a maintainer perspective in order to verify whether the use of UML diagrams provides better supports during comprehension activities on data models. The comparison performed by Shoval and Shiran reveals that there are no significant differences between Extended ER (EER) and OO data models, except for the use of ternary and unary relationships since in this case EER models provide better results. Furthermore, the designers preferred to work with the EER models.

Shoval and Frumermann [2] also perform a comparison of EER and OO diagrams taking into account the user comprehension. As done by Shoval and Shiran [5], they separately examine the comprehension of various constructs of the analysed models. Their analysis reveals that EER schemas are more comprehensible for ternary relationships while for the other constructs no significant difference is found.

Bock and Ryan [6] also examine the correctness of the design for several constructs of the considered diagram types in an empirical analysis comparing EER and OO models from a designer perspective. The analysis reveals significant difference only in four cases (i.e., representation of attribute identifiers, unary 1:1 and binary m:n relationships) and no difference is found concerning the time to complete the tasks.

A comparison between OO and ER models from an end-user perspective is also carried out by Palvia *et al.* [7], whose aim is to establish which is more comprehensible. Differently from previous reported studies, they measure comprehension on overall terms, not considering specific constructs, and the results of their investigation suggested that OO schemas are superior in this respect.

3 Empirical Evaluation

This section describes in detail the design of the controlled experiment we performed and the analysis and interpretation of the achieved results. A discussion of the threats to validity is also presented at the end of the section.

3.1 Goal, Definition, and Context

The *goal* of our experimentation was to analyse whether UML class diagrams are more comprehensible than ER diagrams during the comprehension of data

models. Moreover, we are interested in performing a fine grained analysis to compare the single building blocks B_i of the two notations to identify possible weaknesses of the UML class diagrams with respect to the ER diagrams, where $B_i \in \{ \textit{Entity}, \textit{Primary Key/ID}, \textit{Composite Attribute}, \textit{Multi-value Attribute}, \textit{Recursive relationship}, \textit{Relationship cardinality}, \textit{Ternary relationship}, \textit{Generalization IS-A}, \textit{Weak entity}, \textit{M:N relationship} \}$.

The performed experiments involved students of the University of Salerno (Italy) having different academic backgrounds and, consequently, different levels of experience on ER and UML diagrams:

- *fresher students*, i.e., 1st year B.Sc. students that were starting their academic career when the experiment was performed;
- *bachelor students*, i.e., 2nd year B.Sc. students that attended Programming and Databases courses in the past and were attending the Software Engineering course when the experimentation was performed;
- *master students*, i.e., 1st year M.Sc. students that attended advanced courses of Programming and Software Engineering in the past and were attending an advanced Databases course when the experimentation was performed;

Note that in the Software Engineering course the design notation used is UML while for the Databases course the design notation is ER. The number of subjects involved in the original experiment were 37 bachelor students, while the first and second replications involved 52 master students and 67 fresher students subjects, respectively. We employed the data models of the following systems:

- Company, a software system implementing all the operations required to manage the projects conducted by a company;
- EasyClinic, a software system implementing all the operations required to manage a medical doctor’s office.

In particular, we exploited two different data models represented in terms of ER and UML class diagrams. Table 1 shows the characteristics of the data models we employed in the experiments. The selection of the objects for each experiment was performed ensuring that the data models had a comparable level of complexity. For this reason, we extracted sub-diagrams of comparable size from the original data models according to the “*the rule of seven*” given by Miller [8] to build comprehensible graphical diagrams¹. In the context of our experimentation we applied such a rule to select data models easy to comprehend. This was necessary because (i) each experiment was designed to be performed in a limited amount of time and (ii) a simple data model is preferred to a more complex data model since the latter might influence the comprehension activities.

3.2 Design

Each experiment was organised in two laboratory sessions. In particular, in the context of the experiment subjects had to perform two comprehension activities

¹ The rule of seven is the generally accepted claim that people can hold approximately seven chunks or units of information in their short-term memory at a time [8].

Table 1. Data models used in each controlled experiment

System	# entities	# attributes	# relationships
Company	7	17	5
EasyClinic-BookingManagement	6	18	5

Table 2. Experimental design

Group	Treatment	
	ER	UML
A	EasyClinic, Lab1	Company, Lab2
B	Company, Lab2	EasyClinic, Lab1
C	Company, Lab1	EasyClinic, Lab2
D	EasyClinic, Lab2	Company, Lab1

on the data models of two different software systems. Each subject analysed the UML diagram (or ER diagram) of one system in one laboratory session and the ER diagram (or UML diagram) of the other system in the other laboratory session. The organisation of each group of subjects² in each experimental lab session (*Lab1* and *Lab2*) followed the design shown in Table 2. In particular, the rows represent the four experimental groups, whereas the columns refers to the design notation used to represent the data model (i.e., ER and CD).

3.3 Comprehension Questionnaires

The main outcome observed in the three experiments was the comprehension level. To evaluate it, we asked the subjects to answer a questionnaire (similar to [9]) consisting of 10 multiple choice questions where each question has one or more correct answers. The number of answers is the same for each question (i.e., three answers), while the number of correct answers is different. The questions cover all the building blocks B_i of the two notations exploited to model a database. Figure 1 shows a sample question of the comprehension questionnaire regarding the system Company.

The same building blocks were qualitatively analysed through a questionnaire where subjects specified their preferences between the two considered notations. In particular, for each building block B_i they manifested a preference between ER diagram, No preference, and UML class diagram.

Moreover, at the end of each laboratory session a survey questionnaire was proposed to the subjects. This survey aimed at assessing the overall quality of the provided material as well as the clearness and difficulty of the comprehension tasks. In particular, the subjects provided answers to the following questions (one choice for each question):

- S1 : I had enough time to perform the tasks
- S2 : The task objectives were perfectly clear to me
- S3 : The tasks I performed was perfectly clear to me
- S4 : Judging the difficulty of the comprehension task

² The students were assigned to the four groups in a randomly balanced way.

where S1, S2, and S3 expected closed answers according to the Likert scale [10] from 1 (strongly disagree) to 5 (strongly agree), while S4 from 1 (very low) to 5 (very high).

3.4 Variable Selection

We performed a single factor within-subjects design, where the independent variable (main factor) is represented by the design notation used to represent a data model. This variable is denoted as **Method**, that can be ER diagram (*ER*) or UML class diagram (*CD*).

The dependent variable is **comprehension level**, which denotes the comprehension level achieved by the subjects using the two notations. To measure it we use two well known Information Retrieval metrics, namely recall and precision [11]. Indeed, since the questionnaire is composed of multiple-choice questions, we define recall and precision as follow:

$$recall_s = \frac{\sum_i |answer_{s,i} \cap correct_i|}{\sum_i |correct_i|} \% \quad precision_s = \frac{\sum_i |answer_{s,i} \cap correct_i|}{\sum_i |answer_{s,i}|} \%$$

where $answer_{s,i}$ is the set of answers given by the subjects s to the question i and $correct_i$ is the set of correct answers expected for the question i . Note that the measures defined above represent aggregations of the precision and recall values that have been obtained considering each question of the questionnaire. Differently from aggregate measures based on the mean of precision and recall values the adopted measures also consider the fact that subjects do not provide any answer for a given question [12].

Finally, it is worth noting that recall and precision measure two different concepts. Thus, we decided to use their harmonic mean (i.e., F-measure [11]) to obtain a balance between them and compute the comprehension level.

However, to better assess the effect of **Method** it was necessary to control other factors (called co-factors) that may impact the results achieved by the subjects and be confounded with the effect of the main factor. In the context of our study, we identify the following co-factors:

- **ER and UML experience**: fresher students did not know the ER and UML diagrams, while bachelor and master students had a fairly good knowledge of these notations and master students were more trained than bachelor students on the design methods. We were also interested in analysing the

Q4 Let us focus on the classes Project and Company.
Which of the following statements is true:

- A company has a unique office
- A project has a unique office
- A company may have multiple offices

Fig. 1. A question example

effect of the ER and UML experience since the different levels of education (and, consequently, the different levels of UML and ER experience) may impact the results achieved by subjects.

- **System:** even if we tried to select two software systems of a comparable size and tried to balance the complexity of the data models by using as heuristic the Miller’s rule, there is still the risk that the system complexity may have a confounding effect with **Method**. For this reason we also considered the modeled system as an experimental co-factor.
- **Lab:** the experiments were organised in two laboratory sessions. In the first session subjects performed the task using UML class diagrams (or ER diagrams) and in the other session they performed the task using ER diagrams (or UML class diagrams). Although the experimental design limits the learning effect, it is still important to analyse whether subjects perform differently across subsequent lab sessions.

3.5 Procedure and Data Analysis

Subjects performed the assigned tasks individually. Before the experiments, subjects were trained on both ER and UML class diagrams. To avoid bias (i) the training was performed on a data model not related to the systems selected for the experimentation and (ii) its duration was exactly the same for the experiment and the replications. Right before the experiments, the students attended a 30 minutes presentation where detailed instructions concerning the tasks to be performed were illustrated. The design, the material³ and the procedure were exactly the same for the experiment and its replications. Subjects represented the only substantial difference among the experiment and the two replications.

Since in our experiments each subject performed a task on two different models (i.e., *Company*, or *EasyClinic*) with the two possible treatments (i.e., ER, and CD), it was possible to use a paired Wilcoxon one-tailed test [14] to analyse the differences exhibited by each subject for the two treatments. A one-tailed paired t-test [14] can be used as alternative to the Wilcoxon test. However, we decided to use the Wilcoxon test since it is resilient to strong departures from the t-test assumptions [15]. The achieved results were intended as statistically significant at $\alpha = 0.05$. This means that if the derived p-value is less than 0.05, it can be concluded that there is significant difference between the support given by the treatments when performing comprehension tasks on data models. Furthermore, we analysed the students preferences about the single building blocks of the two notations using histograms, while the answers provided by subjects to the survey questionnaire were analysed using boxplots. The chosen design also permitted to analyse the effects of co-factors and their interaction with the main factor. To this aim we used the two-way Analysis of Variance (ANOVA) [14].

³ See [13] for the complete material used in the experiments.

Table 3. Descriptive statistics of comprehension by method and subjects group

Subjects	ER			CD		
	Mean	Median	St. Dev.	Mean	Median	St. Dev.
Fresher	0.801	1.000	0.307	0.816	1.000	0.280
Bachelor	0.849	1.000	0.242	0.845	1.000	0.278
Master	0.849	1.000	0.277	0.838	1.000	0.272

Table 4. Wilcoxon Test results of comprehension by method and subjects group

Subjects	CD\$FM - ER\$FM			p-value	effect size
	Mean	Median	St. Dev.		
Fresher	0.014	0.000	0.404	0.343	0.037
Bachelor	0.003	0.000	0.330	0.420	-0.011
Master	-0.012	0.000	0.383	0.817	-0.030

3.6 Analysis and Interpretation of the Results

Table 3 reports the descriptive statistics of the F-measure, i.e., comprehension level, achieved by the subjects in our experimentation. The results highlighted that the two notations provided comparable support when performing comprehension activities on data models. In particular, the higher difference between the two notations in terms of F-measure is just 1% (see Table 3). As designed, to analyse if the difference between the results obtained using the two notations is statistically significant, we performed the Wilcoxon test. Table 4 reports the achieved results that highlight no significant difference between the two notations when used to comprehend data models (p-value always higher than 0.05).

Our finding contrasts with the results achieved in [4] where the authors demonstrated the benefits provided by the UML class diagrams with respect to the ER diagrams during the comprehension of data models. To further investigate this discrepancy, we analysed the support given by the two notations at a fine-grained level, i.e., on each building block used in the definitions of data models. Table 5 reports the descriptive statistics of the results achieved in terms of F-measure (considering the subjects answers to questions related to each building block). The achieved results confirmed an overall “performance equilibrium” between the two notations. In particular, there are some building blocks that represent strengths of CD, e.g., Entity and Ternary Relationship, as well as building blocks that represent weaknesses of CD, e.g., Composite and Multi-value attributes. In order to statistically analyse the weaknesses of CD, Table 6 shows the results of the Wilcoxon test executed for each building block to verify where the ER performances are statistically better than those of CD. The achieved results revealed that ER has a comprehension level significantly higher than the comprehension level of CD for three building blocks, i.e., Composite attribute, Multi-value attribute, and Weak entity. These results held for all the subjects involved in the experimentation. The only exception is given by Bachelor students when analysing the Multi-value attribute building block. However, Table 5 shows that Bachelor students also achieved better results in terms of descriptive statistics with ER when answering the questions related to the Multi-value attribute. It

Table 5. Descriptive statistics of the results (F-measure)

Method Element		Fresher			Bachelor			Master		
		Mean	Median	St. Dev.	Mean	Median	St. Dev.	Mean	Median	St. Dev.
ER	Entity	0.887	1.000	0.260	0.936	1.000	0.125	0.872	1.000	0.281
	Primary Key/ID	0.784	1.000	0.406	0.955	1.000	0.179	0.907	1.000	0.277
	Composite attribute	0.883	1.000	0.159	0.897	1.000	0.146	0.920	1.000	0.140
	Multi-value attribute	0.859	1.000	0.195	0.847	1.000	0.168	0.862	1.000	0.213
	Recursive relationship	0.779	1.000	0.301	0.757	0.667	0.224	0.817	1.000	0.243
	Relationship cardinality	0.875	1.000	0.240	0.892	1.000	0.158	0.929	1.000	0.179
	Ternary relationship	0.741	1.000	0.347	0.828	1.000	0.220	0.804	1.000	0.321
	Generalization IS-A	0.684	0.667	0.369	0.734	1.000	0.363	0.712	1.000	0.379
	Weak entity	0.725	0.800	0.266	0.767	1.000	0.305	0.747	0.900	0.329
	M:N relationship	0.789	1.000	0.368	0.865	1.000	0.319	0.923	1.000	0.244
CD	Entity	0.961	1.000	0.108	0.937	1.000	0.234	0.926	1.000	0.145
	Primary Key/ID	0.875	1.000	0.296	0.937	1.000	0.234	0.926	1.000	0.246
	Composite attribute	0.742	0.667	0.255	0.781	0.800	0.251	0.815	1.000	0.308
	Multi-value attribute	0.775	0.667	0.259	0.788	0.667	0.257	0.801	0.667	0.209
	Recursive relationship	0.767	1.000	0.323	0.856	1.000	0.226	0.806	0.800	0.210
	Relationship cardinality	0.865	1.000	0.261	0.856	1.000	0.320	0.906	1.000	0.150
	Ternary relationship	0.827	1.000	0.265	0.888	1.000	0.150	0.855	1.000	0.162
	Generalization IS-A	0.828	1.000	0.225	0.838	1.000	0.290	0.804	1.000	0.328
	Weak entity	0.629	0.667	0.407	0.611	0.667	0.407	0.608	0.733	0.447
	M:N relationship	0.890	1.000	0.162	0.955	1.000	0.179	0.929	1.000	0.212

Table 6. Wilcoxon Test by Questions

Element	Fresher				Bachelor				Master						
	ERSFM - CDSFM		p-value effect size		ERSFM - CDSFM		p-value effect size		ERSFM - CDSFM		p-value effect size				
	Mean	Median	St. Dev.	Dev.	Mean	Median	St. Dev.	Dev.	Mean	Median	St. Dev.	Dev.			
Entity	-0.059	0.000	0.262	0.983	-0.257	-0.036	0.000	0.153	0.599	-0.032	-0.054	0.000	0.309	0.796	-0.161
Primary Key/ID	-0.091	0.000	0.517	0.927	-0.166	-0.027	0.000	0.198	0.415	0.059	-0.019	0.000	0.388	0.660	-0.049
Composite attribute	0.141	0.000	0.303	0.000	0.490	0.116	0.000	0.306	0.022	0.380	0.105	0.000	0.304	0.012	0.343
Multi-value attribute	0.085	0.000	0.316	0.014	0.269	0.059	0.000	0.324	0.141	0.180	0.061	0.000	0.311	0.080	0.196
Recursive relationship	0.012	0.000	0.401	0.455	0.024	-0.010	0.000	0.287	0.983	-0.345	0.011	0.000	0.308	0.536	0.037
Relationship cardinality	0.009	0.000	0.358	0.439	0.028	-0.009	0.000	0.200	0.446	0.094	0.023	0.000	0.224	0.258	0.103
Ternary relationship	-0.086	0.000	0.471	0.897	-0.184	-0.042	0.000	0.266	0.869	-0.221	-0.050	0.000	0.368	0.720	-0.135
Generalization IS-A	-0.145	0.000	0.421	0.999	-0.388	-0.104	0.000	0.476	0.905	-0.217	-0.093	0.000	0.526	0.903	-0.177
Weak entity	0.096	0.000	0.457	0.027	0.211	0.156	0.000	0.504	0.045	0.309	0.139	0.000	0.590	0.049	0.234
M:N relationship	-0.105	0.000	0.379	0.972	-0.249	-0.045	0.000	0.334	0.942	-0.252	-0.006	0.000	0.313	0.562	-0.020

bold if ER comprehension level statistically higher than CD comprehension level

is worth noting that the controlled experiments and replications reported in [4] did not consider these three building blocks to determine comprehension level provided by the two notations, i.e., the questionnaires used by the authors did not include questions related to Composite attribute, Multi-value attribute, and Weak entity. To verify whether the different findings between our experimentation and the results achieved in [4] was due to these three building blocks we also performed the comparison between ER and UML class diagrams without considering the answers of the students related to Composite attribute, Multi-value attribute, and Weak entity. In particular, we re-executed the Wilcoxon test to analyse if CD provided a significant higher comprehension level than ER. The results in Table 7 highlight that CD achieved statistically significant higher comprehension level than ER for the Fresher and Bachelor students. Moreover, CD provided better results than ER also for Master students even if this is not statistically significant (p-value 0.096).

Besides a quantitative analysis, we also conducted a qualitative comparison of the support given by the building blocks of the two notations. Figures 2, 3,

Table 7. Wilcoxon Test results of comprehension support by method and subjects' group without the identified weaknesses

Subjects	CD\$FM - ER\$FM			p-value	effect size
	Mean	Median	St. Dev.		
Fresher	0.066	0.000	0.410	0.000	0.161
Bachelor	0.052	0.000	0.290	0.010	0.120
Master	0.027	0.000	0.358	0.096	0.074

bold if CD comprehension level statistically higher than ER comprehension level

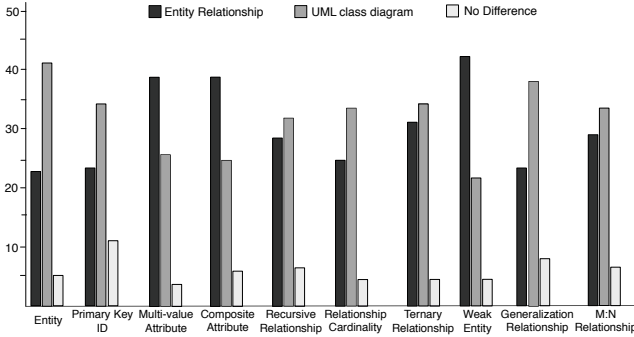


Fig. 2. Subject's preferences - Fresher

and 4 report the preferences expressed by the Fresher, Bachelor, and Master students, respectively. It is worth noting that the results of the quantitative analysis are confirmed by the preferences expressed by the students. In particular, the students preferred ER diagrams to represent the three building blocks identified as weaknesses of the UML class diagrams during the quantitative analysis, i.e., Multi-value attribute, Composite attribute, and Weak entity. Concerning the remaining building blocks, the students preferred UML class diagrams to represent the Entity, the Relationship cardinality, and the Generalization relationship, while they did not provide a clear preference for the Primary key/ID, Recursive relationship, Ternary relationship, and M:N relationship.

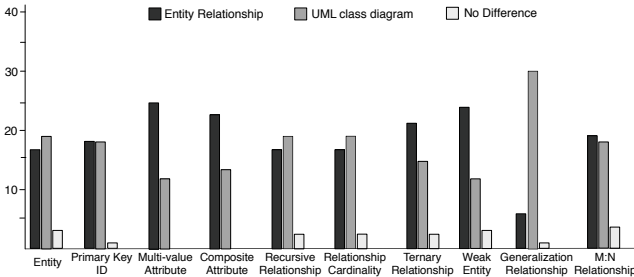


Fig. 3. Subject's preferences - Bachelor

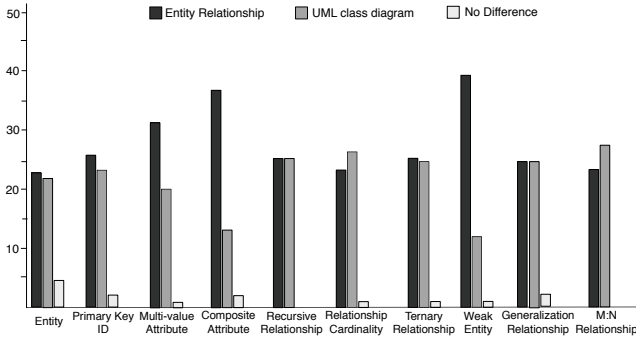


Fig. 4. Subject's preferences - Master

4 Discussion and Threats to Validity

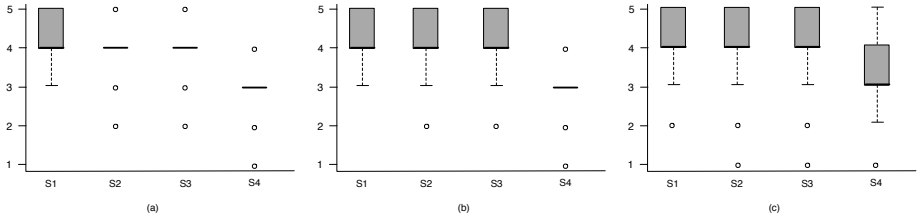
Summarising, the achieved (quantitative and qualitative) results highlighted that the UML notation is characterized by three weaknesses related to the representation of Composite attribute, Multi-value attribute, and Weak entity, with respect to the ER notation, when performing comprehension activity on data models. However, except for the three identified weaknesses, the UML notation is generally more comprehensible than the ER notation, confirming the findings of previous experiments [4]. These findings suggest that a UML class diagram extension focused on these three building blocks should be considered to overcome these weaknesses and improve the comprehensibility of data models given in terms of UML notation. All these findings could be affected by many threats to validity [16] discussed in the following.

Goal, Design, and Statistical analysis. Ease of comprehension was the only criterion examined, because comprehension is a key issue for a graphical notation. However, especially where the design of performance-critical, data-intensive software like databases is concerned, there are other key considerations as well, e.g., analysability. One may choose to sacrifice expressiveness for analysability or other properties. For this reason, future work will be devoted to evaluate other properties of the two notations.

As explained in Section 3 we captured the students' opinion about the quality of the provided material, the clearness of the comprehension tasks and the laboratory goals, and the difficulty in performing the comprehension tasks, to verify if the results of our experimentation could be influenced by these threats. Figure 5 shows boxplots of answers for (a) fresher, (b) bachelor, and (c) master students. The analysis suggested that students had enough time to carry out the tasks (S1) and the objectives and the tasks to perform were clear (S2 and S3), since the median of boxplots of answers was 4 (i.e., I agree). Furthermore, they experienced no particular difficulties when performing the comprehension tasks (S4) since the median of the answers was 3.

Table 8. ANOVA: analysis of the Lab and System co-factors

Factor	Fresher	Bachelor	Master	All
Lab	No (0.787)	No (0.163)	No (0.175)	No (0.216)
System	No (0.793)	No (0.636)	No (0.113)	No (0.229)
Method vs Lab	No (0.817)	No (0.833)	No (0.305)	No (0.439)
Method vs System	No (0.793)	No (0.817)	No (0.618)	No (0.679)

**Fig. 5.** Answers of subjects to survey questionnaire

The metric used to assess the subjects' performance (comprehension) is an aggregate measure of precision and recall that well reflects the results achieved by the subjects. We are also confident that the used tool (multiple-choice questions) actually measures the comprehensibility of the data models. This is also confirmed by the fact that previous empirical studies also used similar approaches to measure the same attributes (see for instance [2], [5], [6], [7], [9]).

Even if the chosen design mitigates the learning (or tiring) effect, there is still the risk that, during labs, subjects might have learned how to improve their comprehension performances. We tried to limit this effect by means of a preliminary training phase. In addition, as highlighted in [15], one possible issue related to the chosen experiment design concerns the possible information exchange among the subjects between the laboratories. To mitigate such a threat the experimenters monitored all the students during the experiment execution to avoid collaboration and communication between them. Finally, subjects worked on three different diagrams and, even if we tried to select diagrams having comparable size, there is still the risk that one diagram might be easier than another.

All these considerations suggest to account **Lab** and **System** as co-factors in the analysis of results. Indeed, the chosen design permitted to analyse the effect of co-factors and their interaction with the main factor. Table 8 shows the results of the ANOVA test by **Method** and **Lab**. The analysis did not reveal any significant influence of the two co-factors nor any significant interaction between the main factor and the two co-factors.

Since the assigned task had to be performed in a limited amount of time, the time pressure could represent another threat to validity. However, we decided the duration of each experiment taking into account previous laboratory exercises performed by the students involved in the experimentations during their courses. Furthermore, we also exploited our experience in performing similar controlled experiments in the past [4]. However, all the subjects completed the assigned

task and they declared (in the post-experiment questionnaire) that the available time was enough to complete the task. For these reasons we are confident that time pressure did not condition the results and thus we did not consider it as a confounding factor.

Proper tests were performed to statistically analyse the difference in the performance achieved employing the two experimented notations, i.e., ER and UML class diagram. Survey questionnaires, mainly intended to get qualitative insights, were designed using standard ways and scales [10] allowing us to use statistical analysis to analyse differences in the feedback provided by subjects.

Subjects and objects. The three controlled experiments involved students having different backgrounds, i.e., fresher, bachelor, and master students. Concerning the undergraduate and graduate students, they had an acceptable analysis, development, and programming experience. In particular, in the context of the Software Engineering courses, both master and bachelor students had participated to software projects, where they experienced software development and documentation production, including database design documents. Moreover, as highlighted by Arisholm and Sjoberg [17] the difference between students and professionals is not always easy to identify. Nevertheless, there are several differences between industrial and academic contexts. For these reasons, we plan to replicate the experiment with industrial subjects to corroborate our findings. We also plan in the future to conduct a survey involving people from database and software engineering communities aiming at obtaining opinions on why weak entity, multi-value and composite attributes are (or might be) problematic in the UML notation. In this way, we can perform a more notation-oriented discussion about the identified weaknesses.

The different backgrounds of the students involved in the experiments have been accounted as a co-factor to analyse its influence on and interaction with the main factor. As expected the ANOVA test revealed a statistically significant effect of **ER and UML Experience** (p -value < 0.001); bachelor and master students achieved statistically significant better performances than fresher students, while the performances achieved by bachelor and master are almost comparable. In addition, ANOVA did not reveal any interaction between **ER and UML Experience** and the main factor (p -value = 0.486).

To avoid social threats due to evaluation apprehension, students were not evaluated on the performances they achieved in the experiments. During the experiment, we monitored the subjects to verify whether they were motivated and paid attention in performing the assigned task. We observed that students performed the required task with dedication and there was no abandonment. Moreover, students were aware that our goal was to evaluate the impact of using ER or UML class diagrams during modelling activities, but they were not aware of the exact hypotheses tested and of the considered dependent variables.

Finally, the size of the data models is small compared to industrial cases, but it is comparable with the size of models used in other related experimentations (see, for instance, [5], [15], [9]). Future work will be devoted to assess the usefulness

of the notations on realistically sized artefacts. However, we believe that the comparison of the two notations on small/medium artefacts is still a worthy contribution.

5 Conclusion and Future Work

We have reported on the results of a controlled experiment and two replications aimed at analysing the support given by ER and UML class diagrams during the comprehension of data models. We have also performed a fine-grained analysis to compare the single building blocks of the two notations (e.g., entity, relationships). The results of the empirical analysis have suggested that UML class diagrams are generally more comprehensible than ER diagrams, confirming the results achieved in a previous study [4]. However, the fine-grained analysis has revealed some weakness of UML class diagrams with respect to ER diagrams. In particular, if we take into account the results about the weak entity, multi-value and composite attributes building blocks, the performances achieved with ER diagrams are superior than those obtained with UML class diagrams. Moreover, the performed qualitative analysis has also highlighted that the subjects preferred ER diagrams for specifying weak entities, multivalued and composite attributes. Taking into account these results, in the future we intend to exploit stereotypes, as done in other studies [9], [18], [19], [20], to extend the UML class diagrams and bridge the gap with ER diagrams about the specification of weak entity, multivalued and composite attributes building blocks. The aim is to improve the comprehensibility of UML class diagrams and candidate such notation as a new de facto standard also for data modeling.

As it always happens with empirical studies, replications in different contexts, with different subjects and objects, is the only way to corroborate our findings. It would be interesting to consider alternative experimental settings in several respects, but maybe the most important one is the profile of the involved subjects. Replicating this study with students/professionals having a different background would be extremely important to understand how UML class diagrams influence the results of these different sub-populations.

References

1. Navathe, S.B.: Evolution of data modeling for databases. *Commun. ACM* 35, 112–123 (1992)
2. Shoval, P., Frumermann, I.: OO and EER conceptual schemas: A comparison of user comprehension. *Journal of Database Management* 5(4), 28–38 (1994)
3. De Lucia, A., Gravino, C., Oliveto, R., Tortora, G.: Assessing the support of ER and UML class diagrams during maintenance activities on data models. In: 2th European Conference on Software Maintenance and Reengineering, CSMR 2008, pp. 173–182 (April 2008)
4. Lucia, A.D., Gravino, C., Oliveto, R., Tortora, G.: An experimental comparison of ER and UML class diagrams for data modelling. *Empirical Software Engineering* 15(5), 455–492 (2010)

5. Shoval, P., Shiran, S.: Entity-relationship and object-oriented data modeling: an experimental comparison of design quality. *Data Knowledge Engineering* 21, 297–315 (1997)
6. Bock, D., Ryan, T.: Accuracy in modeling with extended entity relationship and object oriented data models. *Journal of Database Management* 4, 30–39 (1993)
7. Palvia, P., Lio, C., To, P.: The impact of conceptual data models on end-user performance. *Journal of Database Management* 3(4), 4–15 (1992)
8. Miller, G.A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review* 63, 81–97 (1956)
9. Ricca, F., Di Penta, M., Torchiano, M., Tonella, P., Ceccato, M.: How developers experience and ability influence web application comprehension tasks supported by UML stereotypes: A series of four experiments. *IEEE Transactions on Software Engineering* 36(1), 96–118 (2010)
10. Oppenheim, A.N.: *Questionnaire Design, Interviewing and Attitude Measurement*. Pinter Publishers (1992)
11. Baeza-Yates, R.A., Ribeiro-Neto, B.: *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston (1999)
12. Antoniol, G., Canfora, G., Casazza, G., De Lucia, A., Merlo, E.: Recovering traceability links between code and documentation. *IEEE Transactions on Software Engineering* 28(10), 970–983 (2002)
13. Bavota, G., et al.: UML vs ER - experimental material (2011), <http://sesa.dmi.unisa.it/UMLvsER.html>
14. Conover, W.J.: *Practical Nonparametric Statistics*, 3rd edn. Wiley, Chichester (1998)
15. Briand, L.C., Labiche, Y., Penta, M.D., Yan-Bondoc, H.D.: An experimental investigation of formality in UML-based development. *IEEE Transactions on Software Engineering* 31, 833–849 (2005)
16. Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B., Wesslen, A.: *Experimentation in Software Engineering - An Introduction*. Kluwer, Dordrecht (2000)
17. Arisholm, E., Sjöberg, D.I.K.: Evaluating the effect of a delegated versus centralized control style on the maintainability of object-oriented software. *IEEE Transactions on Software Engineering* 30, 521–534 (2004)
18. Cruz-Lemus, J.A., Genero, M., Manso, M.E., Piattini, M.: Evaluating the effect of composite states on the understandability of UML statechart diagrams. In: Briand, L.C., Williams, C. (eds.) *MoDELS 2005*. LNCS, vol. 3713, pp. 113–125. Springer, Heidelberg (2005)
19. Genero, M., Cruz-Lemus, J.A., Caivano, D., Abrahão, S., Insfran, E., Carsí, J.Á.: Assessing the influence of stereotypes on the comprehension of UML sequence diagrams: A controlled experiment. In: Busch, C., Ober, I., Bruel, J.-M., Uhl, A., Völter, M. (eds.) *MODELS 2008*. LNCS, vol. 5301, pp. 280–294. Springer, Heidelberg (2008)
20. Staron, M., Kuzniarz, L., Thurn, C.: An empirical assessment of using stereotypes to improve reading techniques in software inspections. In: *Proceedings of the Third Workshop on Software Quality, 3-WoSQ*, pp. 1–7. ACM, New York (2005)