Lecture Notes in Computer Science 1723
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Singapore
Tokyo

Robert France Bernhard Rumpe (Eds.)

«UML»'99 – The Unified Modeling Language

Beyond the Standard

Second International Conference
Fort Collins, CO, USA, October 28-30, 1999
Proceedings

Springer

Volume Editors

Robert France
Department of Computer Science
Colorado State University
Fort Collins, CO 80523-1873, USA
E-mail: france@cs.colostate.edu

Bernhard Rumpe
Institut für Informatik, Technische Universität München
D-80290 München, Germany
E-mail: rumpe@in.tum.de

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

The unified modeling language : second international conference ; proceedings / UML '99: Beyond the Standard, Fort Collins, CO, USA, October 28 - 30, 1999. Robert France ; Bernhard Rumpe (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 1999
(Lecture notes in computer science ; Vol. 1723)
ISBN 3-540-66712-1

CR Subject Classification (1998): D.2, D.3

ISSN 0302-9743
ISBN 3-540-66712-1 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN 10705440 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

≪UML≫'99 Preface

"While in geometry attempts to square the circle never succeeded, the UML has achieved it: states can be implemented as classes." – "We have made much progress from the time clouds were used."

The Unified Modeling Language is described as a language for "specifying, visualizing, constructing, and documenting the artifacts of software systems" and for business modeling (OMG UML V1.x documents). The UML reflects some of the best experiences in object-oriented modeling, thus it has the potential to become a widely-used standard object-oriented modeling language.

As a generally-applicable standard the UML has to be both flexible (extensible, adaptable, modifiable) and precise. Flexibility is needed if the UML is to be used in a variety of application domains. Tailoring of UML syntax and adaptation of UML semantics to system domains is highly desirable. Incorporating domain-specific concepts into the language will yield modeling languages that more effectively support system development in these domains. Tailoring may involve determining a subset of the UML that is applicable to the domain, extending or modifying existing language elements, or defining new language elements. One can envisage UML variants that are tailored to specific domains, for example, UML for real-time systems, multimedia systems, and for internet-based systems. Furthermore, one can also define UML variants that determine levels of sophistication in the use of the UML. For example, one can define a "UML-Light" that utilizes basic UML concepts, a "UML-Advanced" that utilizes more advanced concepts, and a "UML-Expert" that uses concepts that require substantial experiences in the use of the UML. In this respect, one can consider the UML to be a family of languages rather than a single, coherent language.

As in the case of natural languages, one does not need to understand the full language before one can express oneself. Consequently, lightweight versions for different purposes are needed, but extensions of the UML beyond stereotypes and tagged-values wherever necessary should be considered in the future. In the fields of business modeling, timed and analogous systems, as well as architectural descriptions, enhancements will surely come, perhaps bringing new specialized kinds of diagrams into the UML.

Precision is needed if the UML is to effectively serve as a standard. A precise language supports effective communication of intent and enables the development of rigorous analysis tools. Work on developing precise semantics for the UML is the main thrust of UML research in academia. The development of a pragmatic and precise semantics for the UML requires both technical and social processes. It is imperative that the semantics support a common-sense usage of the UML in practice. It is not good enough to propose a precise semantics in a formal notation. One must also demonstrate that the proposed semantics

supports commonly held views of how the UML is to be applied and that the semantics is consistent with widely-perceived successful industrial applications of the language. Furthermore, the semantics should give tool-developers useful insight to support the development of semantic analysis tools.

The flexibility and precision qualities may seem at odds with each other. Regarding UML as a family of languages suggests that there cannot be a single precise UML semantics. On the other hand, the multiple languages must have a common language core if they are to be considered UML variants and not new languages. Work on defining a precise semantics for the UML should focus on (1) identifying this core, (2) developing precise characterizations of the core concepts, and (3) developing mechanisms that can be used to extend and modify the core semantics to support the tailoring of the UML to different usages and domains.

Balancing the demands for UML extensions and adaptations with the need to consolidate and unify concepts to create a coherent standard will be a major challenge as the UML evolves. Both forces can contribute significantly to the development of the UML only if appropriately balanced. Demands for extensions and adaptations can be analyzed together to identify common concepts that can be usefully and consistently added to a UML core, but identifying common concepts and determining the consistency of new concepts with existing standard UML concepts are challenging activities.

The evolution of the UML can benefit significantly from the best experiences in other computer science communities. Experiences that can be exploited in the development of the UML include work on conceptual modeling and knowledge engineering in the Artificial Intelligence community, work on rigorous/formal software development in the Software Engineering community, work on data modeling in the Database community, and work on denotational and operational semantics, type theories, and higher-level programming languages in the Programming Language community. For example, it is conceivable that one can use a sub-language of the UML as a higher-level programming language, thus paving the way for the use of the UML as a wide-spectrum development language.

Closely linked to UML issues are questions related to how and where to use and apply it. Current interest in methodical issues and the definition of development processes reflects this awareness. Methods-in-the-Large and project management issues are rather well elaborated, and the "methods in the small" will receive far more attention in the future. We need more techniques that allow composing or refining of the various kinds of diagram types, translate between them, and trace information across diagrams. Proprietary solutions for some techniques are coded in the tools, and need scientific examination to allow further improvement.

We are waiting for the day when the (core) UML will be regarded as a semantically sound and precise language.

The objective of the ≪UML≫'99 conference is to bring together researchers and developers from academia and industry, and from a variety of computer science communities, to present and discuss works that can potentially contribute to the evolution of the UML. In particular, the ≪UML≫'99 conference aims to foster closer working relationships between researchers and developers in industry and researchers in academia. As indicated above, the successful evolution of the UML will require theoretical and industry-driven contributions. Past work on the UML provides ample evidence that concepts developed in academia can be effectively interwoven with practical experiences. The intent of the UML conferences is to enhance such interactions by providing an open forum for discussing and analyzing theoretical and practical challenges facing the development of the UML.

In keeping with the scientific orientation of ≪UML≫'99, the conference is primarily structured around paper presentations and discussion panels. The presentations and panels are targeted to an audience that is at least familiar with the basic elements of the UML, and has a significant interest in the development of the UML as a well-founded standard. In total 166 papers were submitted to the ≪UML≫'99 conference, of which 44 were selected by the programme committee for presentation. The selected papers touch upon a variety of issues and reflect numerous perspectives on how the UML should evolve. The concerns and issues mentioned above, and more, are addressed in varying degrees in the selected papers.

We would like to express our deepest appreciation to the authors of submitted papers, the programme committee members, those committee members who also acted as shepherds for some of the papers, the external referees, Ljiljana Döhring for handling the paper printing process, Adrian Bunk for setting up and handling the electronic submission process, and Matthias Rahlf for setting up the Web page for the electronic programme committee meeting. We would also like to thank the numerous people who have been involved in the organisation of ≪UML≫'99 and, in particular, the organisers of last year's conference in Mulhouse, Jean Bézivin and Pierre-Alain Muller for their helpful advice, the publicity chairs, in particular, Jean-Michel Bruel for maintaining the mailing list, the poster chair, Jim Bieman, and the conference coordinator, Kathy Krell, who kept all the pieces together and made the organisation a much smoother process. We would also like to thank the IEEE-CS conference support staff for their invaluable help.

September 1999 Robert France, Bernhard Rumpe

Organisation

≪UML≫'99 was organised by Robert France from the Department of Computer Science at Colorado State University, and by Bernhard Rumpe from the Computer Science Department at the Technische Universität München, under the auspices of IEEE Computer Society Technical Committee on Complexity in Computing, and in cooperation with ACM SIGSOFT and SIGPLAN (Association for Computing Machinery, Special Interest Group for Software Engineering, Special Interest Group on Programming Languages).

Executive Committee

Conference Chair:	Robert France	(Colorado State University, USA)
Programme Chair:	Bernhard Rumpe	(Technische Universität München, Munich, Germany)

Organising Team

Conference Coordinator:	Kathy Krell
Poster Chair:	Jim Bieman
Panel Chair:	Bernhard Rumpe
Publicity Chair (Europe, Africa):	Jean-Michel Bruel
Publicity Chair (Americas):	Jim Bieman
Publicity Chair (Asia, Pacific):	Junichi Suzuki

Adrian Bunk, Ljiljana Döhring, Emanuel Grant, Matthias Rahlf, and all our on-site student volunteers.

Programme Committee

Colin Atkinson (Universität Kaiserslautern, Germany)
Jean Bézivin (Université de Nantes, France)
Jim Bieman (University of Colorado, USA)
Gregor v. Bochmann (University of Ottawa, Canada)
Ruth Breu (Technische Universität München, Germany)
Jean-Michel Bruel (Université de Pau et des Pays de l'Adour, France)
Frank Buschmann (Siemens AG, Germany)
Betty Cheng (Michigan State University, USA)
Derek Coleman (Hewlett-Packard, USA)
Steve Cook (IBM EMEA Object Technology Practice, UK)
John Daniels (Syntropy Limited, UK)
Desmond D'Souza (Platinum Technology, USA)
Gregor Engels (Universität Paderborn, Germany)
Andy Evans (University of Bradford, UK)
Eduardo B. Fernandez (Florida Atlantic University, USA)
Martin Fowler (Independent Consultant, USA)
Eran Gery (i-Logix, Israel)
Martin Gogolla (Universität Bremen, Germany)
Martin Griss (Hewlett-Packard, USA)
Radu Grosu (University of Pennsylvania, USA)
David Harel (The Weizmann Institute of Science, Israel)
Brian Henderson-Sellers (Swinburne University, Australia)
Pavel Hruby (Navision Software, Denmark)
Peter Hruschka (The Atlantic Systems Guild, Germany)
Heinrich Hussmann (Technische Universität Dresden, Germany)
Ivar Jacobson (Rational Software Corporation, USA)
Gerti Kappel (Johannes Keppler Universität Linz, Austria)
Stuart Kent (University of Kent, Canterbury, UK)
Haim Kilov (Genesis Development Corporation, USA)
Cris Kobryn (EDS, USA)
Philippe Kruchten (Rational Software Corporation, USA)
Kevin Lano (Imperial College, UK)
Gary Leavens (Iowa State University, USA)
Stephen Mellor (Project Technology, USA)
Richard Mitchell (University of Brighton, UK)
Ana Maria Dinis Moreira (Universidade Nova de Lisboa, Portugal)
Pierre-Alain Muller (ObjeXion, France)
Linda Northrop (Carnegie Mellon University, USA)
Gunnar Övergaard (The Royal Institute of Technology, Sweden)
Barbara Paech (Fraunhofer Institute for Experimental Software, Germany)
Jim Rumbaugh (Rational Software Corporation, USA)
Andy Schürr (Universität der Bundeswehr München, Germany)

Programme Committee (continued)

Ed Seidewitz (DHR Technologies, USA)
Bran Selic (ObjecTime Limited, Canada)
Richard Mark Soley (OMG, USA)
Jos Warmer (Klasse Objecten, The Netherlands)
Anthony Wasserman (Software Methods and Tools, USA)
Alan Wills (TriReme International, UK)
Rebecca Wirfs-Brock (Wirfs-Brock Associates, USA)

Additional Referees

Daniel Amyot
João Araujo
Jean-Michel Bruel
Luis Caires
John Cheesman
Birgit Demuth
Jawad Drissi
Falk Fünfstück
Reiko Heckel
James Ivers
Elisabeth Kapsammer
Thomas Khüne
Annig Lacayrelle
Katharina Mehner
John O'Hara
Aziz Salah
Stephane Some
Annika Wagner
Joaquim Aparicio
Michael Breu
Christian Bunse
S. Jeromy Carriere
Robert G Clark
Ralph Depke
Mike Fischer
Emanuel Grant
Martin Hitz
Erik Kamsties
Ismail Khriss
Frank-Ulrich Kumichel
Oliver Laitenberger
Paul Mukherjee
Gianna Reggio
Stefan Sauer
Jean Vaucher
Jörg Zettel

Sponsoring Association

IEEE Computer Society Technical Committee on Complexity in Computing, http://www.computer.org/

Cooperating and Supporting Associations

ACM SIGSOFT (Association for Computing Machinery, Special Interest Group for Software Engineering). http://www.acm.org/sigsoft/

ACM SIGPLAN (Association for Computing Machinery, Special Interest Group for Programming Languages). http://www.acm.org/sigplan/

OMG (The Object Management Group), http://www.omg.org/. UML is a trademark of OMG.

Sponsoring Company

Rational Software Corporation, http://www.rational.com/

Table of Contents

Invited Talk 1 (Abstract)

Architecting Web-Based Systems with the Unified Modeling Language 1
Grady Booch

Software Architecture

Extending Architectural Representation in UML with View Integration ... 2
Alexander Egyed, Nenad Medvidovic

Enabling the Refinement of a Software Architecture into a Design 17
Marwan Abi-Antoun, Nenad Medvidovic

Using the UML for Architectural Description 32
Rich Hilliard

UML and Other Notations

Viewing the OML as a Variant of the UML 49
Brian Henderson-Sellers, Colin Atkinson, Don Firesmith

A Comparison of the Business Object Notation and the Unified Modeling Language 67
Richard F. Paige, Jonathan S. Ostroff

Formalizing the UML Class Diagram Using Object-Z 83
Soon-Kyeong Kim, David Carrington

Formalizing Interactions

A Formal Approach to Collaborations in the Unified Modeling Language .. 99
Gunnar Övergaard

A Formal Semantics for UML Interactions 116
Alexander Knapp

Panel 1

UML 2.0 Architectural Crossroads: Sculpting or Mudpacking? 131
Moderator: Chris Kobryn
Michael Jesse Chonoles, Steve Cook, Desmond D'Souza,
Sridhar Iyengar, Guus Ramackers

Meta-Modeling

Core Meta-Modelling Semantics of UML: The pUML Approach 140
Andy Evans, Stuart Kent

A Metamodel for OCL 156
Mark Richters, Martin Gogolla

Tools

Tool-Supported Compressing of UML Class Diagrams 172
Ferenc Dósa Rácz, Kai Koskimies

A Pragmatic Approach for Building a User-Friendly and Flexible UML Model Repository 188
Mariano Belaunde

Components

Modeling Dynamic Software Components in UML 204
Axel Wienberg, Florian Matthes, Marko Boger

Extending UML for Modeling Reflective Software Components 220
Junichi Suzuki, Yoshikazu Yamamoto

UML Extension Mechanisms

Nine Suggestions for Improving UML Extensibility 236
Nathan Dykman, Martin Griss, Robert Kessler

A Classification of Stereotypes for Object-Oriented Modeling Languages . . 249
Stefan Berner, Martin Glinz, Stefan Joos

First-Class Extensibility for UML - Packaging of Profiles, Stereotypes, Patterns 265
Desmond D'Souza, Aamod Sane, Alan Birchenough

Process Modeling

UML-Based Fusion Analysis 278
Shane Sendall, Alfred Strohmeier

Using UML for Modelling the Static Part of a Software Process 292
Xavier Franch, Josep M. Ribó

Framework for Describing UML Compatible Development Processes 308
Pavel Hruby

Invited Talk 2

On the Behavior of Complex Object-Oriented Systems 324
David Harel

Real-Time Systems

UML-RT as a Candidate for Modeling Embedded Real-Time Systems in the Telecommunication Domain 330
Dominikus Herzberg

Modeling Hard Real Time Systems with UML – The *OOHARTS* Approach 339
Laila Kabous, Wolfgang Nebel

UML Based Performance Modeling Framework for Object-Oriented Distributed Systems 356
Pekka Kähkipuro

Constraint Languages

Defining the Context of OCL Expressions 372
Steve Cook, Anneke Kleppe, Richard Mitchell, Jos Warmer, Alan Wills

Mixing Visual and Textual Constraint Languages 384
Stuart Kent, John Howse

Correct Realizations of Interface Constraints with OCL 399
Michel Bidoit, Rolf Hennicker, Françoise Tort, Martin Wirsing

Analyzing UML Models 1

Generating Tests from UML Specifications 416
Jeff Offutt, Aynur Abdurazik

Formalising UML State Machines for Model Checking 430
Johan Lilius, Iván Porres Paltor

Panel 2

SDL as UML: Why and What 446
Moderator: Bran Selic
Philippe Dhaussy, Anders Ek, Øystein Haugen, Philippe Leblanc, Birger Møller-Pedersen

Coding 1

UML Behavior: Inheritance and Implementation in Current Object-Oriented Languages 457
Jean Louis Sourrouille

UML Collaboration Diagrams and Their Transformation to Java 473
Gregor Engels, Roland Hücking, Stefan Sauer, Annika Wagner

Analyzing UML Models 2

Towards Three-Dimensional Representation and Animation of UML Diagrams 489
Martin Gogolla, Oliver Radfelder, Mark Richters

Typechecking UML Static Models 503
Tony Clark

Precise Behavioral Modeling

Analysing UML Use Cases as Contracts 518
Ralph-Johan Back, Luigia Petre, Iván Porres Paltor

Closing the Gap Between Object-Oriented Modeling of Structure and Behavior 534
Holger Giese, Jörg Graf, Guido Wirtz

Static Modeling

Black and White Diamonds 550
Brian Henderson-Sellers, Franck Barbier

Interconnecting Objects via Contracts 566
Luís Filipe Andrade, José Luiz Fiadeiro

How Can a Subsystem Be Both a Package and a Classifier? 584
Joaquin Miller, Rebecca Wirfs-Brock

Applying the UML

Using UML/OCL Constraints for Relational Database Design 598
Birgit Demuth, Heinrich Hussmann

Towards a UML Extension for Hypermedia Design 614
Hubert Baumeister, Nora Koch, Luis Mandel

Why Unified is Not Universal? – UML Shortcomings for Coping with Round-Trip Engineering 630
Serge Demeyer, Stéphane Ducasse, Sander Tichelaar

Sequence Diagrams

Timed Sequence Diagrams and Tool-Based Analysis – A Case Study 645
Thomas Firley, Michaela Huhn, Karsten Diethers, Thomas Gehrke, Ursula Goltz

Timing Analysis of UML Sequence Diagrams 661
Xuandong Li, Johan Lilius

Coding 2

The Normal Object Form: Bridging the Gap from Models to Code 675
Christian Bunse, Colin Atkinson

Modeling Exceptional Behavior 691
Neelam Soundarajan, Stephen Fridella

Panel 3

Advanced Methods and Tools for a Precise UML 706
Moderator: Andy Evans
Steve Cook, Steve Mellor, Jos Warmer, Alan Wills

Author Index 723