

Preface

*Shall I go off to South America?
Shall I put out in my ship to sea?
Or get in my cage and be lions and tigers?
Or – shall I be only Me?*

A. A. Milne, When We Were Very Young

The Unified Modeling Language (UML) is a language for specifying, visualising, constructing and documenting the artifacts of software systems. The UML originated from the wave of object-oriented analysis and design methods (OOA & D) that appeared in the early 1980's and 1990's. It's formation came about in direct response to a call for a standard OOA & D method by the Object Management Group (OMG) in 1996/7.

Since then, the UML has arguably become the lingua franca of the software engineering community – it is rare to find a new CASE tool, software engineering text, course or method that does not support UML in some way. This success can be attributed to many factors. The most important of these has been UML's capitalization of a general move in the software industry towards open standards. This has demonstrated that industry is more interested in a common, standardised modelling approach than in the particular philosophies that distinguished the earlier approaches to OOA & D. In addition, the UML standard has clearly benefited from its association with the OMG, which has provided an excellent open forum for its development. Currently, the OMG/UML standard is at version 1.4. However, at the time of writing, UML is on the verge of a lengthy period of major review and revision, which will result in version 2.0.

Version 2.0 of the UML represents an opportunity to realise many exciting and visionary ideas that are emerging in the field of object modelling. The first of these is the opportunity to develop a generally applicable standard modelling language, whose semantics and notation can be adapted to suit a wide variety of application domains. Such a language has a clear advantage over fixed definition languages, in that new variants can be quickly developed to meet different modelling requirements. In terms of UML, this means being able to view UML as a family of languages, i.e. a set of variations within the confines of a common core semantic base. The term that has emerged to describe these variations is a 'profile'. A profile is a UML semantic definition which extends and tailors the UML meta-model to a specific domain, process or application. Already, many examples of UML profiles are emerging (some of which are described in these proceedings). These include, user interaction, common warehouse data, software process engineering and real-time profiles, among many others.

Currently, UML 1.3 and, in particular, UML 1.4 provide a number of extension mechanisms which support profile design. However, significant work is being done to understand how version 2.0 can build on and extend these facilities. In

particular, it seems important that the core part of UML that forms the root of all profiles be clearly delineated and precisely defined. Furthermore, UML 2.0 needs to provide the methodologies and tools necessary to support the difficult task of profile building. Ideally, some sort of profile building facility is required, which combines powerful meta-modelling tools, a meta-modelling language (itself a profile), appropriate meta-modelling patterns and tool generators.

Tools are another vital component of the UML 2.0 vision. Until recently, UML tools mainly offered diagram editors, design repositories and basic checking facilities. However, tools are now emerging that support sophisticated analysis and checking of UML models. This includes support for the Object Constraint Language (OCL), UML's standard language for describing constraints. These tools (a number of which are described in these proceedings) are examples of the first of a new breed of industrial strength software modelling tools. As the UML continues to develop, we fully expect that these types of tools will offer the modeller with increasingly powerful means of checking, interacting with and testing models. This will include the ability to analyse incomplete, non-executable and under-determined models, thus greatly improving confidence in the correctness of abstract specifications.

Underlying the development of profiles, tools and notations for UML is the need for greater precision within the next version of the standard. At present, the informal nature of the UML semantics means that there is significant scope for misinterpretation and misuse of the language. Much work being carried out in academia aims to address this problem by developing formal specifications of the UML semantics. However, it is increasingly apparent that formal specifications alone do not address the needs of the UML community. Industrial practitioners require a semantics that it is readily understandable, that can be interacted with using tools, and which is scalable and adaptable to the needs of profile definition. To this end, it appears likely that OO technologies (object modelling, patterns, reflection, component-based development, frameworks, product lines, and so on) have a key role to play in the production of such a semantic definition.

Finally, the widespread application of UML is leading to further refinements of the language's notations and supporting methodologies. Practical experiences in many other fields, including architectural design, data modelling, real-time systems and user interaction are significantly contributing to the development of the language. As such, this is where the true worth of UML will be determined and reflected in future versions of the standard. For only if a component of UML is effective in practice will it continue to flourish. Understanding the importance of practicality, and the experiences of other disciplines, is a vital step towards recognising the primary reason for UML's success in the first place.

The objective of \ll UML \gg 2000, in line with the \ll UML \gg '99 and \ll UML \gg '98 conferences, is to bring together researchers and developers from both academia and industry, to present and discuss *their* visions of the future of the UML standard.

In total 102 abstracts and 82 papers were submitted to the conference, of which 36 were selected by the programme committee for presentation. Previ-

ous «UML» conferences have been primarily structured around paper presentations and discussion panels. However, this year's conference also included a two-day tutorial and workshop session, in which 6 tutorials and 6 workshops were scheduled. These were selected from 20 tutorial and 12 workshop submissions. The primary purpose of these sessions was to provide a more informal forum for discussing state-of-the-art research in UML. Topics included: real-time UML, web applications, the OCL, interactive systems, tool support, extreme modelling and component-based development. Links to the workshops can be found at the conference web site: <http://www.cs.york.ac.uk/uml2000>.

We would like to express our deepest appreciation to the authors of submitted papers, tutorials, workshops and panels, and the programme committee members and the additional referees. In particular, Jaelson Castro and Stephen Mellor did an excellent job of managing the workshop and tutorial submissions. We would also like to thank Steve Cook, Ivar Jacobson and Cris Kobryn for agreeing to present invited talks at the conference. James Willans and Sara-Jayne Farmer at the University of York are also thanked for their contribution to setting up the conference web site and in organising and handling the electronic submission process. The START program (<http://www.cs.umd.edu/~rich/start.html>) was used to gather and organise submitted papers and reviews, and was extended to deal with an online voting process. We would also like to thank the «UML» steering committee for their advice, Jean-Michel Bruel and Robert France for maintaining the mailing list, and Kathy Krell for agreeing to act as on-site meeting organiser.

August 2000

Andy Evans
Stuart Kent
Bran Selic

Organisation

«UML»2000 was organised by Andy Evans from the Department of Computer Science at the University of York, UK, by Stuart Kent from the Computing Laboratory at the University of Kent, UK, and by Bran Selic from Rational Software Inc., Canada, under the auspices of the IEEE Technical Council on Complexity in Computing, and in cooperation with ACM SIGSOFT.

Executive Committee

General Chair:	Bran Selic (Rational Software Inc., Canada)
Conference Chair:	Stuart Kent (University of Kent, UK)
Programme Chair:	Andy Evans (University of York, UK)
Tutorial Chair:	Stephen Mellor (Project Technology Inc., USA)
Workshop Chair:	Jaelson Castro, (Universidade Federal de Pernambuco, Brazil)

Organising Team

On-site Coordinator:	Kathy Krell
Web Programming :	James Willans and Sara-Jayne Farmer
Publicity Chair (Europe) :	Jean-Michel Bruel
Publicity Chair (Americas):	Robert France

Programme Committee

Colin Atkinson (Germany)	Haim Kilov (USA)
Jean Bezivin (France)	Cris Kobryn (USA)
Grady Booch (USA)	Craig Larman (USA)
Alan Brown (USA)	Peter Linnington (UK)
Jean-Michel Bruel (France)	Ian Maung (USA)
David Bustard (UK)	Stephen Mellor (USA)
Jaelson Castro (Brazil)	Richard Mitchell (UK)
Betty Cheng (USA)	Ana Moreira (Portugal)
Derek Coleman (UK)	Pierre-Alain Muller (France)
Steve Cook (UK)	John Mylopoulos (US)
John Daniels (UK)	Gunnar Overgaard (Sweden)
Desmond D'Souza (USA)	Niall Ross (UK)
Gregor Engels (Germany)	James Rumbaugh (USA)
Eduardo. B. Fernandez (USA)	Bernhard Rumpe (Germany)
Robert France (USA)	Andy Schuerr (Germany)
Martin Gogolla (Germany)	Keng Siau (USA)
Brian Henderson-Sellers (Australia)	Ed Seidewitz (USA)
Heinrich Hussmann (Germany)	Perdita Stevens (UK)
Pavel Hruby (Denmark)	Jos Warmer (Netherlands)
Peter Hrushchka (Germany)	Anthony Wasserman (USA)
Jean-Marc Jezequel (France)	Alan Cameron Wills (UK).

Additional Referees

Joao Araujo	Emanuel Grant
Mohammed Benattou	Luuk Groenewegen
Lutz Bichler	Reiko Heckel
Jim Bieman	Wolfram Kahl
Alan Birchenough	Manuel Kolp
Christian Bunse	Thomas Kuehne
Luis Caires	Jiahne Liao
Laura Campbell	Katharina Mehner
Robert Clark	Wai Ming Ho
Tony Clark	Ansgar Radermacher
Ralph Depke	Aamod Sane
Mike Fischer	Stefan Sauer
Falk Fünfstück	Anne Thomas

Sponsoring Association



IEEE Technical Council 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/>



OMG (The Object Management Group), <http://www.omg.org>.
UML and «UML» are trademarks of the OMG.

Sponsoring Companies



Kennedy-Carter, UK. <http://www.kc.com/>



Project Technology Inc. USA. <http://www.projtech.com/>



Rational Software Inc. Canada. <http://www.rational.com>