



Proceedings of the  
Workshop on OCL and Textual Modelling  
(OCL 2011)

Preface

Jordi Cabot, Robert Clarisó, Martin Gogolla and Burkhart Wolff

3 pages

## Preface

**Jordi Cabot<sup>1</sup>, Robert Clarisó<sup>2</sup>, Martin Gogolla<sup>3</sup> and Burkhardt Wolff<sup>4</sup>**

<sup>1</sup>[jordi.cabot@inria.fr](mailto:jordi.cabot@inria.fr) - École des Mines de Nantes, France

<sup>2</sup>[rclariso@uoc.edu](mailto:rclariso@uoc.edu) - Universitat Oberta de Catalunya, Spain

<sup>3</sup>[gogolla@informatik.uni-bremen.de](mailto:gogolla@informatik.uni-bremen.de) - University of Bremen, Germany

<sup>4</sup>[Burkhardt.Wolff@lri.fr](mailto:Burkhardt.Wolff@lri.fr) - Université de Paris-Sud, France

**Abstract:** This preface reports on the 11th workshop on OCL and Textual Modelling held at the TOOLS Federated Conferences in 2011. The workshop focused on the current state of OCL (standard, tool support, adoption, ...) and the application of textual modelling to different domains. The workshop included presentations together with a discussion session. All the workshop materials are available at <http://gres.uoc.edu/OCL2011>.

**Keywords:** OCL, UML, modelling, constraint

## Overview

The Unified Modeling Language (or UML for short) has originally been conceived as a graphical modeling language, largely inspired by ER diagrams and a collection of other graphical notations used in software engineering. The UML became a widely-used industrial standard for the specification, construction and documentation of software and system components. However, practice has shown that in some areas textual notations of the UML are preferable. A first example was the Object Constraint Language (OCL): a notation to define complex integrity constraints in UML models which were hard to describe visually. Other limitations are integral to the Model-Driven Engineering paradigm, i.e. describing large and complex models, enabling model execution and transformation, or facilitating model management. All these issues can be addressed by using a suitable textual language, e.g. Textual MOF, Alloy, Epsilon, ...

The OCL 2011 workshop was established as an opportunity to bring together researchers and practitioners in textual modelling and discuss recent advances in the field, identify opportunities for cooperation and identify relevant research problems and future directions. In particular the workshop provided a forum for existing textual modelling standards such as OCL to be debated and to discuss the current state-of-the-art in textual modelling (tool support, industry adoption, ...).

All submissions were reviewed by three members of the Programme Committee:

- Michael Altenhofen, SAP, Germany;
- Thomas Baar, Tech@Spree, Germany;
- Mariano Belaunde, Orange Labs, France;

- Behzad Bordbar, University of Birmingham, UK;
- Achim D. Brucker, SAP, Germany;
- Tony Clark, Thames Valley University, UK;
- Manuel Clavel, IMDEA Software, Spain;
- Dan Chiorean, University of Cluj, Romania;
- Joanna Chimiak-Opoka, University of Innsbruck, Austria;
- Birgit Demuth, Technical University of Dresden, Germany;
- Miguel García, Swiss Federal Institute of Technology, Switzerland;
- Geri Georg, Colorado State University, USA;
- Alexander Knapp, University of Augsburg, Germany;
- Richard Paige, University of York, UK;
- Mark Richters, Astrium Space Transportation, Germany;
- Pieter Van Gorp, University of Eindhoven;
- Ed Willink, Thales, UK;
- Steffen Zschaler, King's College London, UK.

## Workshop Program

The workshop was scheduled with a full-day program divided into three sessions:

- **OCL:** The first session presented recent results on the analysis and validation of OCL constraints, proposals for the extension of OCL and an analysis of the current state of the OCL standard.
- **Textual modelling and applications:** The second session explored the application of textual modelling to several domains: simulation of embedded hardware, runtime monitoring of applications and workflow execution.
- **Community and discussion:** The workshop concluded with a presentation on the status of OCL tool support among several toolkits and a discussion session.

The discussion session included contributions from several participants, including stakeholders from academia and industry. The debate focused mainly on two topics:

- strategies for improving interoperability among OCL tools;

- directions of improvement for OCL.

Regarding tool interoperability, the current state of affairs was considered far from satisfactory: there is no widely supported exchange format that allows a comparison even among equivalent tools. Therefore, using the same UML/OCL model in several tools requires a manual and time-consuming recreation of the model in different GUIs and IDEs.

Part of the problem was blamed on the representation and exchange of the underlying UML model. The discussion focused on whether a textual syntax or a pivot model encapsulating the core UML concepts required in OCL would be a more adequate solution. Each alternative was discussed together with its advantages and shortcomings: the pivot model is a more elegant solution which is better aligned with the UML standard, while a textual syntax is more practical and facilitates tool adoption as OCL specifications would become self-contained. No consensus was reached on which one was the best alternative, but it was considered an important topic for future research and for the consolidation of OCL.

Regarding directions of improvement for OCL, the discussion focused on the adoption of OCL in industry. Industrial stakeholders complained of lack of pragmatism in the definition of the OCL standard, i.e. it should be easier to implement, rather than being general enough to support any underlying language or platform. Two specific areas that were blamed were: underspecification in the standard (e.g. polymorphism is not defined precisely in order to accommodate different languages and semantics) and platform independent constructs (e.g. unbounded data types).

The debate focused on how to bridge the gap between OCL and its implementation. Changes to the standard were considered detrimental, as introducing bounded data types as OCL basic types would introduce low-level issues in the definition of constraints. A view on this topic was that closeness to implementation should be achieved through extensions, e.g. component libraries. As the use of component libraries is not yet a common practice in OCL, this creates an interesting direction for future work.

**Acknowledgements:** The workshop organizers are grateful to authors, participants, program committee members, additional referees, the TOOLS organizers and the local organization at ETH for their work and their contribution.

Jordi Cabot, Robert Clarisó, Martin Gogolla and Burkhart Wolff  
July 2011