OCL for (Meta-)Models in Multiple Application Domains

Dan Chiorean[1], Birgit Demuth[2], Martin Gogolla[3] and Jos Warmer[4]

[1] "Babeş-Bolyai" University of Cluj-Napoca, Romania,
[2] Technische Universität Dresden, Germany,
[3] University of Bremen, Germany,
[4] and Ordina, The Netherlands
chiorean@cs.ubbcluj.ro, bd1@inf.tu-dresden.de,
gogolla@informatik.uni-bremen.de and jos.warmer@ordina.nl

Abstract. The workshop OCLApps 2006 was organized as a part of MoDELS/UML Conference in Genova, Italy. It continues the series of five OCL (Object Constraint Language) workshops held at previous UML/MoDELS conferences between 2000 - 2005. Similar to its predecessors, the workshop addressed both people from academia and industry. The advent of the MDA (Model Driven Architecture) vision and the rapid acceptance of MDE (Model Driven Engineering) approaches emphasize new application domains (like Semantic Web or Domain Specific Languages) and call for new OCL functionalities. In this context, the OCLApps 2006 Workshop, was conceived as a forum enabling researchers and industry experts to present and debate how the OCL could support these new requirements.

1 Motivation and Goals

In recent years, MDE, MDA and associated methodologies, multiple approaches and languages emphasized the role that OCL has to play in MDE development. Beyond using OCL in querying models, in specifying assertions and operations, new approaches and visions revealed beneficial usage of OCL to model behavior description, compilation and evaluation of models, model transformation, and code generation. All these usages concern modeling languages in general, and UML in particular. As visual modeling languages evolve, complementary textual formalisms must evolve in accordance.

The workshop promoted contributions and experience reports related to the adequacy of OCL specifications, new and classical usages, OCL refactoring and extensions in order to support new usages of OCL specifications and new applications domains. Topics of interest included (but were not limited to): OCL as a textual formalism for specifying (meta-)models, irrespective of the abstraction level; use of OCL in Domain Specific Languages (DSLs); use of OCL in specifying model behavior; OCL usage for navigating and querying models and databases; tools supporting model compilation and execution; OCL and Model

Transformations, needed extensions, experience reports; using OCL specifications in metamodeling; OCL and business process modeling; OCL and ontology modeling, experience reports on OCL usage in profile specification and profile usage in application domains; aspects needing to be clarified in OCL specifications.

2 Organization

The workshop continued the series of OCL workshops held at previous MODELS/UML conferences: York, 2000, Toronto, 2001, San Francisco, 2003, Lisbon, 2004, and Montego Bay, 2005. The workshop was organized by Dan Chiorean, Birgit Demuth, Martin Gogolla, and Jos Warmer. The organizers were at the same time co-chairs of the Program Committee. Each submission was reviewed by two to three members of the Program Committee. Based on the reviews, the decisions regarding papers acceptance were taken unanimously. Three papers were co-authored by one of the workshop PC members. The review process ensured that the authors had no influence on the acceptance/rejection decision for papers written by them.

The PC (in addition to the workshop organizers) consisted of: Thomas Baar (EPFL Lausanne, Switzerland), Jordi Cabot (Universitat Oberta de Catalunya, Spain), Tony Clark (Xactium, United Kingdom), Andy Evans (Xactium, United Kingdom), Robert France (Colorado State University, USA), Heinrich Hussmann (LMU Munich, Germany), Marcel Kyas (University of Oslo, Norway), Richard Mitchell (Inferdata, USA), Octavian Patrascoiu (LogicaCMG, United Kingdom), Mark Richters (EADS, Germany), Shane Sendall (Switzerland), Peter Schmitt (Universität Karlsruhe, Germany), Burkhart Wolff (ETH Zurich, Switzerland).

3 Topics and approaches of accepted papers

The 18 accepted papers, [DCGW06] cover a large spectrum of OCL related topics. They reflect research contributions and experience reports about using OCL for models and metamodels in multiple application domains. In order to manage the paper presentations, the accepted papers were divided into four sections: new applications, model transformations, implementation of OCL support in tools and language issues.

In their paper *Customer Validation of Formal Contracts*, [HJ] Heldal and Johannisson present a modality of translating OCL assertions into natural language. In order to obtain good results, the formal specifications have to comply with some rules. As the authors mentioned, taking into account that the formalisms used by UML have different degrees of formality, the obtained results could be used in a broader framework for UML model transformation.

Kolovos, Paige and Polack describe in the paper *Towards Using OCL for Instance-Level Queries in Domain Specific Languages*, [KPP] an ongoing work towards defining a rigorous approach meant to align the OCL query and navigation facilities with DSLs. In their approach, the authors used a new language, very closed to OCL, defined by themselves and named EOL. The work presented confirmed again that OCL is well-suited for model querying in metamodeling.

In the paper *OCL-based Validation of a Railway Domain Profile*, [Ber] Kirsten Berkenkötter proves that using UML, tailored by means of dedicated profiles, represents an alternative to DSLs. Undoubtedly, each of these above mentioned approaches presents both advantages and drawbacks. Therefore, now, accumulating experiences related to these different ways represents a mandatory task. Berkenkoötter's paper, succeeds in proving that using OCL in specifying profiles for a real application represents a valuable approach. The quantity and quality of application code, automatically generated, is among the main benefits obtained by using profiles specified with OCL.

An experience showing that OCL 2.0 is expressive enough to be used as a query language for model analysis is presented by Joanna Chimiak-Opoka and Chris Lenz in *Use of OCL in a Model Assessment Framework: An experience report*, [COL]. The assessment framework was composed by three different components: a Modeling Environment, a Model Data Repository and an Analysis Tool. The considered metamodel was meant to support optimization of clinical processes in the framework of a real project named MedFlow. The paper proves that using OCL in real applications is profitable.

Tsukasa Takemura and Tetsuo Tamai state that using OCL in modeling business processes, their approach can bypass a well-known problem, named the "Business-IT gap". Their position is presented in the paper *Rigorous Business Process Modeling with OCL*, [TT]. The approach uses activity diagrams enriched by OCL expressions. Also, in order to fully support the specific requirements, appropriate OCL extensions are proposed.

In model transformations, a crucial activity for MDE success, OCL usage is one preferred approach for researchers. In this context, a metamodel-driven model interchange was proposed by Dragan Gasevic and his colleagues in *On Interchanging between OWL/SWRL and UML/OCL*, [MGG$^+$]. The paper aims to reuse the results obtained in MDA to Semantic Web Languages. Moreover, the described work represents a new step towards the reconciliation of Meta Object Facility (MOF) based languages and Semantic Web Languages. The authors and go beyond the approach established by the OMG's ODM (Ontology Definition Metamodel) specification that only addresses mappings between OWL and UML, and extend this with mappings between SWRL and OCL.

In the paper *Realizing UML Model Transformations with USE*, [BB] - Fabian Büttner and Hanna Bauerdick enrich the specification language of USE with imperative elements in order to provide a flexible instrument meant to support experiments with different transformations and transformation formalisms.

A modality for specifying OCL constraints in an automated manner by using "constraints patterns" is described by Wahler, Koehler and Bruckner in the paper *Model-Driven Constraint Engineering*, [WKB]. Inspired by design patterns, the authors prove that the use of patterns could be useful for constraints also. Patterns support modelers to focus on relevant aspects enabling the use of verified specifications. The example used for introducing the problem proves that even very simple UML models need in some cases complex OCL specifications. An important conclusion is that OCL represents the appropriate tool supporting models in a complete and unambiguous manner.

The efficiency of OCL support in large scale modeling environments is analyzed by Altenhofen, Hettel and Kusterer in the paper entitled *OCL support in an industrial environment*, [AHK]. The described problem concerns the reduction of necessary evaluations for OCL constraints, when the underlying model changes. The experiences presented by authors were meant to realize an efficient OCL integration into the next generation modeling infrastructure of SAP, called MOIN.

Stölzel, Zschaler and Geiger discuss in the paper *Integrating OCL and Model Transformations in Fujaba*, [MSG] the integration of the Dresden OCL Toolkit into the Fujaba Tool Suite. This integration adds OCL support to class diagrams and makes OCL usable in Fujaba's model transformations. The Fujaba support for model transformations gets more powerful, platform independent and easier to read for developers familiar with OCL. The possibility of using OCL in Fujaba story diagrams in order to support code generation is also presented.

An optimization algorithm for the evaluation of OCL constraints used in graph rewriting based model transformations is presented by Mezei, Levendovszky and Charaf in their paper *Restrictions for OCL constraint optimization algorithms*, [MLC]. This is a very important topic because efficient constraint handling is essential in UML, in metamodeling, and also in model transformation.

Brucker, Doser and Wolff describe in the paper *An MDA Framework Supporting OCL*, [BDWa] a tool chain for processing UML/OCL specifications, including a proof environment and flexible code generation. In the presented framework, named SecureUML, OCL supports both model verification and transformation, including code generation. Using conjoint functionalities for checking, prooving and transforming offers opportunities for new and up to now unexpected applications.

Amelunxen and Schürr explain in the paper *On OCL as part of the metamodeling framework MOFLON*, [AS] the role of OCL in the metamodeling framework MOFLON, designed to support the definition of domain specific languages with MOF, OCL and graph transformations. Beyond this interesting problem, the authors present a set of constraints (well-formedness rules) which corrects, completes and improves MOF 2.0 for the application as graph schema language.

In the paper *Ambiguity issues in OCL postconditions*, [Cab] Cabot identifies common ambiguities appearing in OCL postconditions and proposes an approach to automatically disambiguate these situations by means of providing a default interpretation for each kind of ambiguous expression. The advantages of the proposed solutions are: a more rigorous specification and a better quality of the code generated for OCL postconditions.

Akehurst, Howells and McDonald-Maier discuss in the paper *UML/OCL - Detaching the Standard Library*, [AHMM] about an OCL Standard Library that is independent from different modeling languages. Considering the increasing use of OCL as a query language for models, the independence of the standard library is an important user requirement that has to be supported by OCL implementations in DSLs.

The results of a long-term project, intended to provide a formalized, machine-checkable semantic basis for a theorem proving environment for OCL is described by Brucker, Doser and Wolff in the paper entitled *Semantic Issues of OCL: Past, Present, and Future*, [BDWb]. The final objective of the above mentioned project, is to make OCL more fit for future extensions towards program verifications and specification refinement.

The paper *Improving the OCL Semantics Definition by Applying Dynamic Meta Modeling and Design Patterns*, [CP] by Chiaradía and Pons presents an alternative definition for the OCL semantics metamodel by applying the Visitor design pattern. The proposal claims to avoid circularity in the OCL definition, and to increase its extensibility, legibility and accuracy.

Süß proposes in the paper *Sugar for OCL*, [Süß] three shorthand notations for the layout of OCL in the Latex, HTML, and Unicode (a la ObjectZ) encoding systems, in order to obtain specifications which more readable.

4 Discussion and Conclusion

In the following, we summarize some of the topics that were discussed at the workshop.

OCL offers support for model transformation, including automated code generation. In this last context, the OCL use is advantageous from two perspectives: firstly, for the support in querying models and secondly, for offering important additional information for code generation. The use of the constraint language in large scale models needs to consider the efficiency of evaluating specifications. As different language extensions are required for an efficient use of OCL in different domains (like Business Process Modeling or the Semantic Web) considering OCL as a family of languages becomes a more important requirement. Fixing different ambiguities concerning the OCL must be realized. As Eclipse tends to be a quasi unanimously tool in the modelers world, efficiently promoting OCL needs strong support within the Eclipse framework.

References

[AHK] Michael Altenhofen, Thomas Hettel, and Stefan Kusterer. OCL support in an industrial environment. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 126–139. TUD-FI06-04.

[AHMM] D. H. Akehurst, W.G.J. Howells, and K.D. McDonald-Maier. UML/OCL - Detaching the Standard Library. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 205–212. TUD-FI06-04.

[AS] C. Amelunxen and A. Schürr. On OCL as part of the metamodeling framework MOFLON. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 182–193. TUD-FI06-04.

[BB] Fabian Büttner and Hanna Bauerdick. Realizing UML Model Transformations with USE. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 96–110. TUD-FI06-04.

[BDWa] Achim D. Brucker, Jürgen Doser, and Burkhart Wolff. An MDA Framework Supporting OCL. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 166–181. TUD-FI06-04.

[BDWb] Achim D. Brucker, Jürgen Doser, and Burkhart Wolff. Semantic Issues of OCL: Past, Present, and Future. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 213–228. TUD-FI06-04.

[Ber] Kirsten Berkenkötter. OCL-based Validation of a Railway Domain Profile. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 38–52. TUD-FI06-04.

[Cab] Jordi Cabot. Ambiguity issues in OCL postconditions. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 194–204. TUD-FI06-04.

[COL] Joanna Chimiak-Opoka and Chris Lenz. Use of OCL in a Model Assessment Framework: An experience report. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 53–67. TUD-FI06-04.

[CP] Juan Martín Chiaradía and Claudia Pons. Improving the OCL Semantics Definition by Applying Dynamic Meta Modeling and Design Patterns. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 229–239. TUD-FI06-04.

[DCGW06] Birgit Demuth, Dan Chiorean, Martin Gogolla, and Jos Warmer, editors. *OCL for (Meta-)Models in Multiple Application Domains*, TUD-FI06-04. Institut für Software- und Multimediatechnik, Technische Universität Dresden, 09 2006. ISSN 1430-211X.

[HJ] Rogardt Heldal and Kristofer Johannisson. Customer Validation of Formal Contracts. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 13–25. TUD-FI06-04.

[KPP] Dimitrios S. Kolovos, Richard F. Paige, and Fiona A. C. Polack. Towards Using OCL for Instance-Level Queries in Domain Specific Languages. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 26–37. TUD-FI06-04.

[MGG+] Milan Milanovic, Dragan Gasevic, Adrian Giurca, Gerd Wagner, and Vladan Devedzic. On Interchanging Between OWL/SWRL and UML/OCL. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 81–95. TUD-FI06-04.

[MLC] Gergely Mezei, Tihamer Levendovszki, and Hassan Charaf. Restrictions for OCL constraint optimization algorithms. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 151–165. TUD-FI06-04.

[MSG] Steffen Zschaler Mirko Stölzel and Leif Geiger. Integrating OCL and Model Transformations in Fujaba. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 140–150. TUD-FI06-04.

[Süß] Jörn Guy Süß. Sugar for OCL. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 240–252. TUD-FI06-04.

[TT] Tsukasa Takemura and Tetsuo Tamai. Rigorous Business Process Modeling with OCL. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 68–80. TUD-FI06-04.

[WKB] Michael Wahler, Jana Koehler, and Achim D. Brucker. Model-Driven Constraint Engineering. In *OCL for (Meta-)Models in Multiple Application Domains*, pages 111–125. TUD-FI06-04.