OCL and Model Driven Engineering

Jean Bézivin[1], Thomas Baar[2], Tracy Gardner[3], Martin Gogolla[4], Reiner Hähnle[5], Heinrich Hussmann[6], Octavian Patrascoiu[7], Peter H. Schmitt[8], and Jos Warmer[9]

[1] University of Nantes, France
Jean.Bezivin@sciences.univ-nantes.fr
[2] EPFL Lausanne, Switzerland
Thomas.Baar@epfl.ch
[3] IBM in Hursley, United Kingdom
tgardner@uk.ibm.com
[4] University of Bremen, Germany
gogolla@Informatik.Uni-Bremen.DE
[5] Chalmers University, Gothenburg, Sweden
reiner@cs.chalmers.se
[6] University of Munich, Germany
Heinrich.Hussmann@inf.tu-dresden.de
[7] Computing Laboratory, University of Kent, United Kingdom
O.Patrascoiu@kent.ac.uk
[8] Universität Karlsruhe, Germany
pschmitt@ira.uka.de
[9] De Nederlandsche Bank, Nederland
jos.warmer@ordina.nl

Abstract. Precise modeling is essential to the success of the OMG's Model Driven Architecture initiative. At the modeling level (M1) OCL allows for the precision needed to write executable models. Can OCL be extended to become a full high-level executable language with side-effects? At the meta-level (M2), queries, views and transformations are subjects that will be vital to the success of the OMG's Model Driven Architecture initiative. Will OCL 2.0 become an essential part of the Queries/Views/Transformations standard and what will be its application areas in industry? Can the features of OCL 2.0 be used in the Model Driven Engineering (MDE) approach? This workshop aims at bringing together people from academia that are expected to report on inspiring ideas for innovative application scenarios and tools, and industrial practitioners, which are expected to provide statements on their view of the future of OCL in the context of MDE.

1 Introduction

The workshop was organized as a part of Seventh International Conference on the Unified Modeling Language <<UML>> 2004 in Lisbon, Portugal. It continued a series of OCL workshops held at previous UML conferences: York, 2000, Toronto 2001, and San Francisco 2003 and outside UML conferences in Amsterdam and Canterbury. Following the successful model of its predecessors this workshop addressed both people from academia and industrial practitioners. The aim was to provide a forum for

N. Jardim Nunes et al. (Eds.): UML 2004 Satellite Activities, LNCS 3297, pp. 67 – 75, 2005.

the exchange of views on the future of OCL, to foster the identification of strategic goals for OCL and increase cooperation within OCL community.

OMG initiated in 2002 the standardization process for MOF 2.0 Query/ Views/ Transformations. In 2003, as a result of OMG's RFP, several proposals for the standardization of QVT were submitted. In this situation it is important to look ahead to the future of OCL. The main focus of this workshop was the investigation of OCL's relation with the OMG's Model Driven Architecture (MDA) framework, at the meta-model level (M2) with the future standard for QVT. There is a clear need for a high-level language to enable modelers to specify behavior at a high level of abstraction. OCL can be extended to become such an Executable UML language. An interesting question is what extensions need to be added to OCL enable this.

At the same time we solicited contributions using OCL as a constraint language on the application modeling level. Substantial progress has been achieved in this area over the last years and we encouraged in particular the submission of case studies and papers on the relation between OCL and annotation languages.

Precise modeling is essential to the success of the Model Driven Engineering (MDE) approach to develop software systems (SS). OCL can play a role at multiple levels. At the meta-level (M2), queries, views and transformations are subjects that will be vital to the success of the MDE. Will OCL 2.0 become an essential part of the Queries/Views/Transformations standard and what will be its application areas in industry?

At the modeling level (M1) OCL allows for the precision needed to write executable models. Currently OCL is restricted to side-effect free queries. Can OCL be extended to become a full high-level executable language with side effects?

How will the powerful features of OCL 2.0 be used in the Model Driven Engineering approach? Is OCL 2.0 more powerful than needed, or is not powerful enough? This workshop aimed at bringing together people from academia that are expected to report on inspiring ideas for innovative application scenarios and tools, and industrial practitioners, which are expected to provide statements on their view of the future of OCL in the context of Model Driven Engineering.

2 Objectives of the Workshop

The workshop focused on:

- Object Constraint Language and the OCL2.0 standard.
- Model Driven Engineering.
- OMG's Queries/Views/Transformations.

The objective of the workshop was to bring together a mix of leading industry, government, and university software architects, component software framework developers, researchers, standards developers, vendors, and large application customers to do the following:

- Better understand the features of OCL 2.0 and how far they go in solving problems in software industry.
- Better understand the relation between OCL and QVT.
- Identify key directions, convergence approaches and characterize open research problems and missing architectural notions in MDE.

The workshop consisted of a set of 9 invited presentations and a final discussion session. Topics of interest listed in the Call for Participation included (but were not limited to):

- OCL – the query language for Model Driven Engineering.
- Contributions to the standardization process of QVT.
- Extensions of OCL to support QVT.
- Reports on OCL or QVT case studies, tools, or applications.
- Theoretical/fundamental aspects of OCL.
- Case studies for precise modeling using OCL.
- OCL as an Executable UML language.
- Dynamic concepts in OCL.

3 Presented Papers

1. On Generalization and Overriding in UML 2.0, Fabian Büttner and Martin Gogolla, University of Bremen, Computer Science Department, Database Systems Group.

In the upcoming Unified Modeling Language specification (UML 2.0), subclassing (i.e., generalization between classes) has a much more precise meaning with respect to overriding than it had in earlier UML versions. Although it is not expressed explicitly, UML 2.0 has a covariant overriding rule for methods, attributes, and associations. In this paper, we first precisely explain how overriding is defined in UML 2.0. We relate the UML approach to the way types are formalized in programming languages and we discuss which consequences arise when implementing UML models in programming languages. Second, weaknesses of the UML 2.0 metamodel and the textual explanations are addressed and solutions, which could be incorporated with minor efforts, are proposed. Despite of these weaknesses we generally agree with the UML 2.0 way of overriding and provide supporting arguments for it.

2. OCL for the Specification of Model Transformation Contracts, Eric Cariou, Raphaël Marvie, Lionel Seinturier, and Laurence Duchien, LIFL - Université des Sciences et Technologies de Lille, UMR CNRS 8022 - INRIA Futurs, 59655 Villeneuve d'Ascq Cédex – France, {cariou,marvie,seinturi,duchien}@lifl.fr

A major challenge of the OMG Model-Driven Architecture (MDA) initiative is to be able to define and execute transformations of models. Such transformations may be defined in several ways and with various motivations. Our motivation is to specify model transformations independently of any transformation technology. To achieve this goal, we propose to define *transformation contracts*. We argue that model transformation contracts are an essential basis for the MDA, they can be used for specification, validation and test of transformations. This paper focuses on the specification of model transformation contracts. We investigate the way to define them using standard UML and OCL features. In addition to presenting the approach and some experimental results, this paper discusses the relevance and limits of standard OCL to define transformation contracts.

3. Rule-Based Simplification of OCL Constraints, Martin Giese, Reiner Hähnle, and Daniel Larsson, Chalmers University of Technology, School of Computer

Science and Engineering, 41 296 Gothenburg, Sweden, {giese, reiner, danla}@cs.chalmers.se

To help designers in writing OCL constraints, we have to construct systems that can generate some of these constraints. This might be done by instantiating templates, by combining prefabricated parts, or by more general computation. Such generated specifications will often contain redundancies that reduce their readability. In this paper, we explore the possibilities of simplifying OCL formulae through the repeated application of simple rules. We discuss the different kinds of rules that are needed, and we describe a prototypical implementation of the approach.

4. OCL as Expression Language in an Action Semantics Surface Language, Stefan Haustein and Jörg Pleumann, Computer Science Dept. VIII/X, University of Dortmund, Germany, {stefan.haustein,joerg.pleumann}@udo.edu

With the specification of Action Semantics in UML 1.5, the OMG laid ground to manipulating object diagrams in a formal way, which is a necessary prerequisite for QVT. In QVT, of course the manipulations take place at M1 level instead of M0, but due to the architecture of UML, the same mechanisms can simply be reused. Unfortunately, the Action Semantics specification does not mandate a surface language, limiting its practical application. Due to the high overlap with the Object Constraint Language, in this article we propose a surface language that is based on and aligned with OCL.

5. Disambiguating Implicit Constructions in OCL, Kristofer Johannisson, Department of Computing Science, Chalmers University of Technology and Göteborg University, S-41296 Göteborg, Sweden, krijo@cs.chalmers.se

A rule system for type checking and semantic annotation of OCL is presented. Its main feature is the semantic annotation and disambiguation of syntax trees provided by an OCL parser, in particular for implicit property calls and implicit bound variables. It is intended as a component to be plugged in to other systems that handle OCL. An implementation of the system is available.

6. Comparing Two Model Transformation Approaches, Jochen M. Küster and Shane Sendall and Michael Wahler, Computer Science Department, IBM Zurich Research Laboratory, CH-8803 Rüschlikon, Switzerland, email: {jku, sse, wah}@zurich.ibm.com

For the MDA vision to become a reality, there must be a viable means to perform model-to-model transformation. In this paper, we compare and contrast two approaches to model transformation: one is a graph transformation-based approach, and the other is a relational approach, based on the QVT-Merge submission for OMG's MOF 2.0 Query/View/Transformation Request for Proposal. We apply them both to a common example, which involves transforming UML state machines to a CSP specification, and we look at some of the concrete and conceptual differences between the approaches.

7. Composition of UML Described Refactoring Rules, Slavisa Markovic, Swiss Federal Institute of Technology, Department of Computer Science, Software Engineering Laboratory, 1015 Lausanne-EPFL, Switzerland, e-mail: Slavisa.Markovic@epfl.ch

Refactorings represent a powerful approach for improving the quality of software systems. A refactoring can be seen as a special kind of behavior preserving model transformation. The Object Constraint Language (OCL) together with the metamodel of Unified Modeling Language (UML) can be used for defining rules for refactoring UML models. This paper investigates descriptions of refactoring rules that can be checked, reused and composed. The main contribution of this paper is an algorithm to compute the description of sequentially composed transformations. This allows one to check if a sequence of transformations is successfully applicable for a given model before the transformations are executed on it. Furthermore, it facilitates the analysis of the effects of transformation chain and its usage in other compositions.

8. Embedding OCL expressions in YATL, Octavian Patrascoiu and Peter Rodgers, Computer Laboratory, University of Kent, UK , {O.Patrascoiu, P.J.Rodgers}@kent.ac.uk

Modeling is a technique used extensively in industry to define software systems, the UML being the most prominent example. With the increased use of modeling techniques has come the desire to use model transformations. While the current OMG standards such as Unified Modeling Language (UML) and Meta Object Facility (MOF) provide a well-established foundation for defining models, no such well-established foundation exists for transforming models. The current paper describes how the OCL expressions are integrated in a transformation language called YATL (Yet Another Transformation Language) to provide support for model querying. The paper presents also the transformation environment and the main features of YATL.

9. Relations in OCL, D.H.Akehurst, Computing Laboratory, University of Kent, D.H.Akehurst @kent.ac.uk

OCL is proposed as a query language within the QVT framework. The main QVT submission bases the specification of transformations on the concept of relations. Relations are not first class entities within the OCL. By extending OCL with the concept of Relations it can better serve the needs of the QVT framework. In particular this enables OCL to be used as a semantic interpretation of a QVT transformation language and may even facilitate the use of OCL as a transformation specification language.

4 Number of Participants

The workshop attracted 28 participants. There exists already a kind of “OCL community”, more and more people are interested in Model Driven Engineering, and many of these people attended the UML conference series.

5 Discussion Session

The final session discussed the following topics:

- Does OCL need extensions?
- Does OCL need refactoring?

- Is it possible to embed/include OCL in other languages/systems? If yes, how hard is it?
- What is the relation between OCL and QVT?
- Has OCL been used in industry in large scale projects?

6 Organizers

- Jean Bézivin, University of Nantes, France
- Thomas Baar, EPFL Lausanne, Switzerland
- Tracy Gardner, IBM in Hursley, United Kingdom
- Martin Gogolla, University of Bremen, Germany
- Reiner Hähnle, Chalmers University, Gothenburg, Sweden
- Heinrich Hußmann, University of Munich, Germany
- Octavian Patrascoiu, University of Kent, United Kingdom (contact)
- Peter H. Schmitt, Universität Karlsruhe, Germany
- Jos Warmer, De Nederlandsche Bank, Nederland

Jean Bézivin is professor of Computer Science at the University of Nantes, France. He got his Master degree from the University of Grenoble and Ph.D. from the University of Rennes. Since 1980 he has been very active in Europe in the object-oriented community, starting the ECOOP series of conference (with Pierre Cointe), the TOOLS series of conferences (with Bertrand Meyer), the Objet'9X industry meeting (with Sylvie Caussarieu and Yvan Gallison), and more recently the <<UML>> series of conferences (with Pierre-Alain Muller). He founded in 1979, at the University of Nantes, one of the first Master programs in Software Engineering entirely devoted to Object Technology (Data Bases, Concurrency, Languages and Programming, Analysis and Design, etc.). His present research interests include object-oriented analysis and design, reverse engineering, knowledge-based software engineering, product and process modeling, model engineering and more specially the techniques of model transformation. He is a member of the ATLAS group, a new INRIA team created at the University of Nantes in relation with the LINA CNRS Lab. On the subjects of model-driven engineering and MDA(tm), he has been recently leading the OFTA industrial group in France, co-animating a CNRS specific action and a Dagstuhl seminar. He is currently involved in several EU projects.

Thomas Baar holds a diploma degree in Computer Science from Humboldt-University of Berlin and a doctoral degree from University of Karlsruhe. In his doctoral thesis, a formal semantics of OCL based on metamodeling techniques is proposed. He published about 10 papers focusing on theoretical and practical issues of OCL. Since 2003, he is a post-doc assistant at the EPFL, Lausanne, Switzerland. His current research area is specification, verification, and testing of software.

Tracy Gardner has a Mathematics and Computer Science degree from the University of Bath and a PhD in the area of programming/modelling language design which was a winner of the CPHC/BCS Distinguished Dissertations award 2000. Tracy has spent time as a practitioner of model-driven development, using the UML-based

Rational Rose Real-Time product while working for Marconi Telecommunications Ltd. Since joining IBM in 2001 Tracy has been involved in model-driven component technologies for business integration. Dr Gardner's current work is on applying Model-Driven Development to the Business Integration domain; she was the main contributor to a UML profile for automated business processes with a mapping to BPEL4WS and is now collaborating on IBM's response to the OMG's Business Process Definition Metamodel and MOF 2.0 Queries/Views/ Transformations RFPs. Tracy has presented on model-driven development at a number of industry conferences (including OMG MDA? Implementers' Workshops, Enterprise UML 2003, 1st European Conference on Model-Driven Software Engineering).

Martin Gogolla is professor for Computer Science at University of Bremen, Germany and is the head of the Research Group Database Systems. His research interests include object-oriented design, formal methods in system design, semantics of languages, and formal specification. Before joining University of Bremen he worked for the University of Dortmund and the Technical University of Braunschweig. His professional activities include: teaching computer science; publications in journals and conference proceedings; publication of two books; speaker to university and industrial colloquia; referee for journals and conferences; organizer of workshops and conferences (e.g. the UML conference); member in national and international program committees; contributor to international computer science standards (OCL 2.0 as part of UML 2.0).

Reiner Hähnle is a Professor in Computer Science at Chalmers University of Technology, Gothenburg, Sweden since 2000. He received diploma and PhD degrees in Computer Science from University of Karlsruhe in 1987 and 1992, respectively. He received a habilitation degree from Technical University of Vienna in 1997. His main research interests are non-classical logics, automated deduction, and the use of formal methods in software engineering. He authored and/or edited three books and is in the author list of over 60 publications. He wrote commissioned articles for both the Handbook of Philosophical Logic (2^{nd}ed) and the Handbook of Automated Reasoning. He was president of the Technical Committee on Multiple-Valued Logic of IEEE CS from 2000 to 2001. He is a member of the steering committees of the IJCAR, FTP, Tableaux, and FloC conference, and co-founder of the Intl Tableaux Conference. He organized numerous workshops and conferences. In 2002, he was conference chair of CADE. Currently, he is involved in IJCAR, MDAFA, and CASSIS as PC member or invited speaker. He is member of the editorial board of Soft Computing, Multiple-Valued Logic, and QPQ (an online journal for publishing peer-reviewed source code for deductive software components). He has been involved in numerous national and international research projects as leader and grant holder. He has been reviewer for several research funding agencies such as the NSF of the US or FP6 of the EU.

Heinrich Hussmann holds a diploma degree in Computer Science from Munich University of Technology and a doctoral degree from University of Passau. He did research and education work at universities in Munich, Passau and Dresden. For several years, he was a systems engineer and team leader in the advanced development laboratory of the telecommunications division of Siemens. From 1997 to

2002 he was full professor for Computer Science at Dresden University of Technology, and since March 2003 he is full professor for Computer Science (Media Informatics) at the University of Munich (LMU). He participated in over 10 national and international projects in the area of software engineering and telecommunications, and is author of over 50 scientific publications, including three internationally published books. He is member of the program committee of the UML conferences since 1999 and a member of the steering committee since 2003. He was conference chair of the UML conference 2002 in Dresden.

Octavian Patrascoiu focused at the beginning of his academic career on programming languages and language processors. He published four books about programming languages, programming techniques and programming language processors. He also presented research papers at numerous conferences. In the last few years, he moved into the area of developing software tools for software quality assesment, software modelling and code generation. He had collaborations with software companies like IBM, Verilog, Telelogic, and TLC.

Peter H. Schmitt holds a diploma and doctoral degree in Mathematics from the University of Heidelberg. His main research contributions at that time lay in the area of Mathematical Logic and Universal Algebra. From 1985 to 1988 he worked for IBM Germany. Since 1988 he is a full professor for theoretical computer science at the University of Karlsruhe. From 1994 to 2000 he has been the chairman of the special interest group on logic and computer science of the German Computer Science Society (GI). Since 1998 he is a member of the Scientific Directorate of SCHLOSS DAGSTUHL, International conference and research centre for Computer Science. He has been involved in numerous, national and international, research projects on automated deduction and non-classical logic. He is author of some 50 scientific papers. He wrote a textbook on the theory of Logic Programming and co-edited a three-volume handbook on Automated Deduction. He is currently working in the area of formal specification and verification of programs.

Jos Warmer is one of the founders of OCL. He was responsible for OCL in the UML 1 core team and has been the leader of the OCL 2 submission team. He has been written books on UML, OCL and recently about MDA. He is a member of the programming committee of the <<UML>> series of conferences. He has been involved in organizing OCL workshops at the <<UML>> conferences, and is co-editor of the LNCS book that was the result of these workshops.

7 Conclusions

This workshop was of clear relevance to the OCL community since it discussed the future role of OCL in the MDE world. The presented papers and the final discussion lead to the following ideas:

1. OCL needs to be refactored by extending the standard library and providing a better concrete syntax.
2. The OCL2.0 standard needs to be improved to avoid misunderstandings and ambiguities.

3. OCL can be easily embedded in other languages and systems (see papers 4 and 8).
4. Both OCL and QVT share a common package of classes at the abstract syntax and semantic levels (e.g. types and expressions).
5. OCL should be used as a query language in QVT
6. OCL can be used in large-scale systems to specify constraints and contracts (see paper 2).