Teaching OCL Standard Library: First Part of an OCL 2.x Course

Joanna Chimiak–Opoka, Birgit Demuth

5 pages
Teaching OCL Standard Library: First Part of an OCL 2.x Course

Joanna Chimiak–Opoka¹, Birgit Demuth²

¹ University of Innsbruck, Austria, joanna.opoka@uibk.ac.at
² Technische Universität Dresden, Germany, birgit.demuth@tu-dresden.de

Abstract: Our aim is to provide a complete set of materials to teach OCL. They can be used in bachelor or master programs of computer science curricula and for training in an industrial context. In this paper we present the first part of the course related to the OCL Standard Library. This part provides model independent examples to teach OCL types and their operations. It enables users to gain a basic understanding of the OCL Standard Library, which can be used as a starting point to write model constraints (OCL specifications) or model queries. Additionally, to the content of the paper, we provide a set of OCL packages, exercise proposals and lecture slides.

Keywords: teaching material, OCL types, model independent OCL expressions

1 Introduction

The Object Constraint Language (OCL) is crucial in precise modeling [WK03]. Despite its usability it is far less frequently used in the industrial context than the Unified Modeling Language. We view three main reasons for this fact: weakness of standardisation, basic tool support, and lack of social acceptance. The OCL standard, also the recent 2.2 version [OMG10], has several known inconsistency and incompleteness issues. Fortunately, regardless of this fact there are many tools supporting OCL. Currently there is a collection of OCL tools and UML tools with OCL support¹. The social acceptance of OCL, results from the previous issues. There exists a belief that it is hard to use, learn and teach OCL.

Our aim is to support learning and teaching of OCL. Based on our teaching experience we designed a complete OCL course material. The course consists of several parts introducing basic types and operations, model and meta model specifications, model constraints and queries. This paper presents the first part of the course with the basic concepts. It focuses on the OCL Standard Library teaching its types by model-independent OCL expressions. Note that it is required to define an empty model context for this purpose. According to our knowledge, our course is the first resource focusing on the core of the OCL (Section 2.2). In our opinion teaching the core of the language is important to its successful application in context of modeling and model transformation. Understanding of model-independent OCL expressions and proficiency in writing them is a prerequisite of an efficient specification of model constraints or model queries. These skills enable focusing on the content instead of on the form of the OCL expressions when writing large specification in OCL.

In the first line we assume the course to be used in the academic context in a computer science curricula. An initial version of the course was successfully taught at the University of Innsbruck

¹ List of tools at OCL Portal: http://st.inf.tu-dresden.de/oclportal/ OCL Software
in model engineering course in the master program. In Dresden, OCL is one of the topics in a course about advanced software engineering and is planned as part of a future course about model–driven software engineering. The improved version of the course will be taught at the University of Innsbruck and at the TU Dresden. In both cases, students have strong basic skills in object–oriented software development (analysis, design including design patterns and Java programming) as well as knowledge about first–order logic. Additionally, the course will be freely available to academic teachers, students, professional software developers and modelers with basic skills in object–oriented software development. We provide two variants of the OCL specifications: standard and extended one. The standard TU Dresden variant is suitable for language purists and for easier tool interoperability as the following constraint shows:

```
context Model inv addRealReal1:
let v1: Real = 1.0 in let v2: Real = 1.0 in v1 + v2 = 2.0
```

It can be used with different tools, however due to inconsistencies in implementations we can only grant the proper usage with the tools we tested. The extended University of Innsbruck variant incorporates libraries, tests and documentation comments [CO09]. It is intended to make learning OCL easier by enriching it with techniques used in programming such as the following analogous OCL test example shows:

```
modelinstance none
test tRealArithmeticOperators:
let v1: Real = 1.0 in let v2: Real = 1.0 expected v1 + v2 = 2.0
```

Teachers have to decide what variant is the most appropriate one in their didactic context. As explained above we will initially consider the basic part of OCL, which is independent of any model. In Fig. 1 we present basic types, omitting types related to modeling, such as Model Element, Enumeration, Unlimited Natural, OCL Message and OCL State. Examples and exercises for all basic OCL types and issues are provided within our course material (Section 2.1). For didactic purposes we introduced several additional elements to the type hierarchy to make it easier to understand. The elements (Type, PrimitiveType, OCLSpecialType, CompositeType) are defined as abstract and depicted in orange in Fig. 1. Please notice that they are not defined in the standard, at least not this way.

The remainder of the paper is structured as follows: In Section 2 we describe teaching resources of our course and list alternative ones. And finally, Section 3 gives a conclusion and future work.

2 Teaching Resources

In this section we provide information about resources that we provide for the described course (Section 2.1). Furthermore, we compare it with other available resources for teaching OCL (Section 2.2).
2.1 Course on OCL Standard Library

The course package\(^2\) contains lecture slides, a long version of this paper with a more detailed presentation of the OCL basic types, source files with tasks for students, and OCL expressions tested with the SQUAM OCL editor and Dresden OCL.

Lecture slides provide an extended version of diagrams for all types, their definitions and issues to be careful about. The slides, with the project and the set of OCL expressions, can be used as self–learning material.

The project consists of 16 packages with over 200 OCL expressions. In the basic part, expressions correspond to particular OCL types (Fig. 1). The advanced part consists of explicit definitions of predefined iterator expressions and complex method definitions for numbers and strings. There are, in total, over 40 exercises to test or extend existing OCL expressions or to write new ones.

The source files are provided as explained in Section 1 in two variants: standard and extended. The overview of available versions is given in Tab. 1. Please note that there are unfortunately differences in the semantics how different tools evaluates OCL expressions.

Table 1: Overview of OCL resources with corresponding OCL versions and tool configurations.

<table>
<thead>
<tr>
<th>version</th>
<th>variant</th>
<th>OCL editor</th>
<th>OCL parser</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCL 2.0</td>
<td>extended</td>
<td>SQUAM OCL Editor</td>
<td>Galileo Eclipse MDT/OCL</td>
</tr>
<tr>
<td>OCL 2.1</td>
<td>extended</td>
<td>SQUAM OCL Editor</td>
<td>Helios Eclipse MDT/OCL</td>
</tr>
<tr>
<td>OCL 2.2</td>
<td>standard</td>
<td>Dresden OCL 2.2 editor</td>
<td>Dresden OCL 2.2 parser</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>tool</th>
<th>webpage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dresden OCL</td>
<td><a href="http://dresden-ocl.sourceforge.net/">http://dresden-ocl.sourceforge.net/</a></td>
</tr>
<tr>
<td>SQUAM OCL Editor</td>
<td><a href="http://squam.info/ocleditor/">http://squam.info/ocleditor/</a></td>
</tr>
<tr>
<td>Eclipse MDT/OCL</td>
<td><a href="http://wiki.eclipse.org/MDT/OCL">http://wiki.eclipse.org/MDT/OCL</a></td>
</tr>
</tbody>
</table>

\(^2\) http://squam.info/ocleditor/doc/OCLCourse/
2.2 Alternative Examples

As far as we know, our course is the only one focusing on model–independent expressions and covering the semantics and newest methods introduced in OCL 2.2. However, there are other valuable teaching resources we would like to mention here.

The well–known and most extensively used example in OCL teaching is the “Royal and Loyal” system example introduced and used in OCL text books as in [WK03]. The example provides an extensive set of model–dependent OCL expressions for earlier versions of OCL (up to 2.0). OCL expressions for this example are provided with several tools, among others with Dresden OCL, Eclipse MDT/OCL, and ITP/OCL.

Several OCL courses (scripts, slides, OCL expressions) are available from the OCL Portal. Additionally, further examples of OCL expressions are provided by Martin Gogolla together with USE and as teaching materials including UML and OCL in Conceptual Modeling and Exercises for Teaching OCL Constraints.

UML tools which support OCL typically also provide OCL examples for documentation or evaluation such as Papyrus. Screencasts and videos can also be helpful to learn how to use...
OCL in a development environment. MagicDraw UML\textsuperscript{11} and Borland Together\textsuperscript{12} provide such online tutorials. Besides UML model examples, students can study OCL usage for models based on other metamodels than UML, such as the Java metamodel, XSD (XML Schema), and EMF Ecore\textsuperscript{13}.

3 Conclusion

Teaching OCL is a challenging task because of the imprecise and incomplete standard specification [OMG10]. This fact has, over the years, caused confusion about the nature of OCL and differences in its semantics and implementations. As a result there is a resistance to teaching and to learning this language. Our intention was to help to overcome this resistance by providing a solid course on the core part of OCL. We provided instructions for teachers and learning materials which include OCL expressions tested with OCL tools developed at our universities.

The course presented in this paper is the first part of a larger OCL course we plan to provide. It will be integrated into and distributed with our OCL tools to give users a comprehensive example set of them. The second part of the planned OCL course will present examples how to specify business rules by OCL constraints on the model layer. OCL constraints for the specification of well-formed rules on the metamodel layer as well as model queries will be subject of the third part of the planned course.

Acknowledgements: The research herein is partially conducted within the competence network Softnet Austria (www.soft-net.at) and funded by the Austrian Federal Ministry of Economics (bm:wa), the province of Styria, the Steirische Wirtschaftsförderungsgesellschaft mbH (SFG), and the city of Vienna in terms of the center for innovation and technology (ZIT).

We would like to thank our colleagues for their support in conceptual and technical aspects of our teaching project, especially Colin Atkinson, Hannes Mösl and the Dresden OCL developers.

Bibliography


\textsuperscript{11} http://www.magicdraw.com/files/viewlets/MD_viewlets_Validation_viewlet.swf.html
\textsuperscript{12} http://www.borland.com/de/products/together/