Adaptations to OCL for Ensuring Quality of Geographic Data*

Miro Casanova, Thomas Wallet, Maja D'Hondt

System and Software Engineering Laboratory - Vrije Universiteit Brussel
Pleinlaan 2 - 1050 Brussels - Belgium

{ mcasanov, twallet, mjdhondt } @vub.ac.be

This poster illustrates how we use the *Unified Modeling Language (UML*) [1] to represent concepts from geographic data, and how we employ the associated *Object Constraint Language (OCL)* [4] to describe quality criteria as complex domain-specific constraints. We propose some adaptations to OCL which originated from the geographic domain, though reusable in other domains where complex constraints need to be expressed on a conceptual, knowledge-oriented model. Finally we briefly describe the *Business-Rule Enabling Kernel*, which is a module of a large distributed quality assurance system for the geographic data production process of our industrial partner *Tele Atlas Data.* This module is used to write the constraints, hereafter also referred to as OCL constraints, that are automatically checked against the persistent geographic data.

1 Geographic Data

Digital geographic data is used in sophisticated applications such as Geographic Information Systems (GIS) which operate on persistent geographic data, obtained from source material such as satellite images or scanned maps. In order to improve the efficiency of capturing and producing geographic data, the *Geographic Data Files* standard [5] has been developed, thus providing a common reference model for clients and producers alike. GDF describes real-world concepts, attributes and relations in the geographic domain in a high-level and implementation-independent way. Nevertheless, current practices skip the high-level GDF description and immediately enter the data in the persistency layer at the implementation level, thereby amplifying the inevitable impedance mismatch.

Moreover, capturing digital geographic data from maps and images induces errors and inaccuracies, thus dramatically reducing data quality. Quality of geographic data is defined here as its integrity and well-formedness. Currently, at the conceptual level quality is specified by domain-specific quality constraints described in natural language, whereas at the implementation level it is manually hard-coded in some quality assurance tool.

Achieving the highest possible quality is essential in producing geographic data. One only needs to consider the consequences of using poor quality data in applications such as car navigation, and alarm call and dispatch.

2 Ensuring Quality of Geographic Data

The aforementioned difficulties lead us to the conclusion that a high-level conceptual representation of the geographic data is required. This representation of geographic concepts, attributes and relations is used for expressing the corresponding constraints, also at a conceptual level. This will ensure implementation independence and a close match with the real-world data in order to minimize loss of information. Moreover, expressing the constraints requires a level of intuitiveness and declarative power comparable to that of natural language without losing formality and correctness.

However, from this conceptual representation, the Business-Rule Enabling Kernel should automatically generate code that checks if none of the relevant persistent geographic data violates this particular constraint. This should be integrated in a larger quality assurance system.

Since concepts, attributes and relations of the GDF standard seamlessly map onto classes, attributes and associations of the

* This research, in collaboration with *Tele Atlas Data Ghent NV,* is part of the project *GeoObjects (IWT 990025),* funded by the *Flemish institute for the Improvement of the Scientific-Technological Research in the Industry (IWT).*

OOPSLA 2000 Companion Minneapolis, Minnesota

class diagram of UML, this notation was chosen as representation medium. UML's accompanying OCL was selected for expressing the constraints, since it adequately fulfills our requirements.

3 Adaptations to OCL

The specific domain of this project, and the kind of constraints needed by our industrial partner, induced us to use OCL in a slightly different way than proposed in the UML 1.3 specifications [5]. Since we do not need all the features of the original OCL, we started from a subset of OCL excluding all the features related to pre- and post-conditions, and defined some extensions that we present here.

- *Constraints on Multiple Classes*

Most of the constraints we use are invariants involving multiple classes. For this reason we propose a simple way to refer to several classes (contexts) within the same constraint, as also proposed in [3]. Thus, a constraint is no longer attached to the context of a single class.

- *Constraints on Multiple Instances*

Sometimes a constraint involves several instances of the same class at a time, in order to denote some relationship among them. To support this in a simple way, we introduce identifiers that can explicitly refer to different instances of a class in a constraint.

- *Parametric Constraints*

Due to the occurrence of similar constraints that operate on different types, we propose an extension where it is possible to specify some parametric types for a constraint. A concrete constraint can then be obtained by providing those type parameters. We also provide a way to bound the type parameters with a 'where' clause.

- *Referencing and Composing Constraints*

In order to refer within an OCL constraint to other constraints, we created a referencing mechanism for constraints, which provides a powerful way for composing constraints. Constraints can easily be used as atomic OCL expressions and then be composed together using all the expressive power of OCL. This referencing mechanism also provides a way to use a parametric constraint by referencing it and specifying its type parameters.

For examples and descriptions of evaluation issues of the resulting OCL, we refer to [2].

4 The Business-Rule Enabling Kernel

As a support for the resulting OCL, we developed a *Business-Rule Enabling Kernel*, which is a module integrated in a large quality assurance system of our industrial partner, providing the following functionalities:

- *Managing and editing OCL constraints*

Users can create, delete and modify OCL constraints. For each constraint it is possible to edit the corresponding OCL expression, as well as other properties associated.

- *Checking constraint validity*

Once an OCL constraint is written, our module checks it lexically, syntactically, but also performs type checking. This type checking is done against a UML conceptual model of the user-defined classes, in this case the geographic data model.

- *Generating code*

OCL constraints are automatically mapped to code for checking the corresponding constraints against the data of the geographic database of our industrial partner.

5 Conclusions

We present some adaptations to OCL that make it more concretely and easily usable in the domain of geographic data. The corresponding *Business-Rule Enabling Kernel* we developed enables users to write OCL constraints, reasoning about high-level concepts of their domain, in a simple and modular way. These constraints are automatically mapped to code for checking them against persistent geographic data.

In a nutshell, we show how to use OCL and UML concretely for ensuring quality of geographic data. However our adaptations to OCL and our specific use of it can be put to use in other knowledge-oriented domains where complex constraints need to be expressed.

References

[1] G. Booch, M. Fowler and K. Scott. "UML Distilled: A Brief Guide to the Standard Object Modeling Language". 2nd Edition. Addison-Wesley. 1999.

[2] M. Casanova, T. Wallet and M. D'Hondt. "Ensuring Quality of Geographic Data with UML and OCL". In *Proceedings of the <<UML>> 2000 Conference* (to appear). Springer-Verlag, 2000.

[3] S. Gaito, S. Kent and N. Ross. "A Meta-model Semantics for Structural Constraints in UML". In H. Kilov, B. Rumpe, and I. Simmonds editors, Behavioral specifications for businesses and systems, chapter 9, pages 123-141. Kluwer Academic Publishers, Norwell, MA, September 1999.

[4] A. Kleppe and J. Warmer. "The Object Constraint Language: Precise Modeling with UML". Addison-Wesley, 1999.

[5] "The Geographic Data Files Standard" Technical Committee 278 Road Transport and Traffic Telematics of the Comitée Européen de Normalisation.

[6] "UML 1.3 Specifications" (including OCL specifications). http://www.omg.org/uml