Tutorial – An Overview of UML 2

Bran Selic
IBM Canada
770 Palladium Drive
Kanata, Ontario, CANADA K2V 1C8
1+(613) 591-7915
bselic@ca.ibm.com

ABSTRACT

This half-day tutorial covers the salient features of the first major revision of the Unified Modeling Language – UML 2. This short note summarizes the major topics covered by the tutorial.

Categories and Subject Descriptors

D.2.2. [**Design Tools and Techniques**]: Object-oriented design methods

General Terms

Design, Standardization, Languages

Keywords

Unified Modeling Language, UML, model-driven development, Model-Driven Archtiecture.

1. INTRODUCTION

The Unified Modeling Language (UML) was first announced in 1995 and, following an open community process, it was adopted as an industry standard by the Object Management Group (OMG) in 1997 [4]. Since then, its use has been steadily increasing in both industry and academia to the extent that it has become the prevalent general-purpose tool for modeling software. The cumulative experience with UML created strong pressures to expand the capabilities of the language. This led to the issuing of a new request for proposals (RFP) by the OMG, calling for a major revision. The new requirements sought more precisely defined language semantics, greater clarity of the specification, and a variety of new modeling capabilities.

These requirements were the result of the view—inspired in part by the success of UML)—that, due to their higher levels of abstraction, software models had the potential to be used for more than just documentation and high-level design. This view engendered an approach to software development called *model-driven development* (MDD) in which models, supported by suitable computer-based automation, play a central role in the development process. One of the key forms of automation is the automatic derivation of programs from semantically precise models. Other forms of automation include automatic test case generation, mathematically-based formal analyses of design properties using specialized computer programs, and the direct execution of high-level models on a computer.

MDD is not a new idea and it had been used sporadically in industrial practice long before UML 2 came around. However, what was interesting at this particular juncture, is that the supporting automation technologies had finally reached a level of capability that made MDD much more scalable and, hence, more practical. In response to these developments, the OMG launched its *Model-Driven Architecture* (MDA) initiative [5], with the intent of supporting MDD with a suitable set of open industry standards. UML 2 [3][6][7][8] is one of the more important MDA standards, but, by no means the only one. In fact, MDA presumes a spectrum of standard modeling languages for different purposes and domains. In support of this, it incorporates the Meta-Object Facility (MOF), a special standard to be used for defining modeling languages [9].

2. THE PRINCIPAL FEATURES OF UML 2

The principal differentiators that distinguish UML 2 from its predecessor are:

1. *Higher degree of precision in the language definition.* This is necessitated by the higher levels of automation inherent in MDD, which require the elimination of ambiguity and imprecision from models.
2. *A highly modular language architecture*, which allows the language to be learned and used in gradually more sophisticated increments based on the needs of the domain and project on hand.
3. *New modeling capabilities.* The majority of the new modeling features in UML 2 are extensions to existing capabilities that allow more direct and, generally, more scalable modeling of today's complex distributed applications (such as business processes and service-oriented architectures). To achieve scalability, many UML 2 modeling concepts are defined recursively, allowing very complex systems to be represented using basic hierarchical composition and decomposition. These are discussed in more detail in section 3 below.
4. *Enhanced support for domain-specific languages.* Although standard UML can be used without further specialization, it is, in essence, a general-purpose language base from which one can derive highly specialized domain-specific languages. This has the advantage that it enables direct reuse of widely-available UML tools and expertise.

Copyright is held by the author/owner(s).
ICSE'06, May 20-28, 2006, Shanghai, China.
ACM 1-59593-085-X/06/0005.

3. NEW MODELING CAPABILITIES

UML 2 includes a number of important new features that allow it to be used as an architectural description language. The basis for these capabilities comes from previous work on architectural description languages [1][2][11]. These languages support hierarchical modeling of run-time structures of communicating components. The concepts are generally defined recursively, so that components at one level in the system hierarchy can be decomposed into finer-grained structures of collaborating components, and so on. This allows modeling of systems at arbitrary levels of abstraction.

A similar approach is used to define behavior that occurs within such structures, with the ability to recursively decompose high-level specifications of object interactions into finer-grained ones and vice versa. The source for this work were earlier standards defined in the telecom domain, where they have been applied with great effectiveness.

The initial version of UML was primarily based on the object paradigm. However, there are many domains, including business process modeling and systems engineering, which have developed powerful modeling capabilities based on a functional view of behavior. UML 2 supports this through a significantly enhanced activity modeling formalism. The semantics foundation for activities is now based on Petri nets rather than on the more constraining state machines used in UML 1. In addition, major new capabilities have been added to support the kinds of functional flows that occur in complex business processes.

4. SUMMARY

UML 2 is a language that was designed for model-driven development methods – although it can still be used in the traditional way as an informal tool for capturing and describing ideas in a technology independent way. Of course, it represents just the first generation of modeling languages, comparable to Fortran as a harbinger of things to come. Without doubt, we will see new and more sophisticated modeling languages emerging in the near future based on the experience gained from applying and studying UML 2.

5. ACKNOWLEDGMENTS

The author would like to thank all the members of the UML 2.0 definition team as well as the members of the UML 2.1 revision task force.

6. REFERENCES

[1] D. Garlan, et al., "ACME: Architectural Description of Component-Based Systems," in G. Leavens and M. Sitarman (eds.), *Foundations of Component-Based Systems,* Cambridge University Press, pp. 47-68. 2000.

[2] ITU, "Specification and Description Language (SDL) – Z.100," Available at: http://www.itu.int/ITU-T/studygroups/com10/languages/Z.100_1199.pdf. November 1999.

[3] OMG, "UML 2.0 OCL – 2nd Revised Submission", OMG document ptc/03-01-07, Available at: http://www.omg.org/cgi-bin/doc?ad/2003-01-07. January 2003.

[4] OMG, The, "Unified Modeling Language, v1.5", OMG document formal/03-03-01, Available at: http://www.omg.org/cgi-bin/doc?formal/03-03-01. March 2003.

[5] OMG, "MDA Guide Version 1.0.1", OMG document omg/03-06-01, Available at: http://www.omg.org/docs/omg/03-06-01.pdf. June 2003.

[6] OMG, "UML 2.0 Diagram Interchange – Draft Adopted Specification", OMG document ptc/03-07-03, Available at: http://www.omg.org/cgi-bin/doc?ptc/2003-07-03. July 2003.

[7] OMG, "UML 2.0 Superstructure – Final Adopted Specification", OMG document ptc/03-08-02, Available at: http://www.omg.org/cgi-bin/doc?ptc/2003-08-02. August 2003.

[8] OMG, "UML 2.0 Infrastructure – Final Adopted Specification", OMG document ptc/03-09-15, Available at: http://www.omg.org/cgi-bin/doc?ptc/2003-09-15, September 2003.

[9] OMG, "MetaObject Facility – Final Adopted Specification," ptc/04-10-15, Available at: http://www.omg.org/cgi-bin/doc?ptc/2004-10-15, October 2004.

[10] J. Rumbaugh, et al., *The Unified Modeling Language Reference Manual*, Addison-Wesley-Longman, Reading, Mass., 1998.

[11] B. Selic, et al., *Real-Time Object-Oriented Modeling*, John Wiley & Sons, New York, 1994.