

WHAT UML SHOULD BE

UML is at a crossroads. Which of the proposed revisions will bring it closer to meeting user needs and winning tool-vendor commitment? What if UML2 instead combined the best features of each proposal?

By Joaquin Miller, *Guest Editor*

WITH the Unified Modeling Language, the craft of building information systems began to fashion the kind of tool mature engineering disciplines have had for centuries—a shared graphical language for descriptions and specifications. UML1, standardized in 1997 by the Object Management Group, was widely adopted by users motivated by their own practical needs, despite somewhat lackluster implementation by tool vendors and consternation among researchers. A stream of publications continues to debate what the UML specification means and detail its

ILLUSTRATION N BY PAUL ZWOLAK

apparent contradictions and unmet needs.

As you read this editorial debate, the OMG membership is considering proposals for a second version of UML (see the sidebar “OMG and UML”). The five statements represent competing schools of thought about UML resulting from a discussion in which members of UML2 proposal teams exchanged (sometimes vitriolic, sometimes politic) position papers, then carried on occasionally heated email exchanges (see the sidebar “UML Proposals Scorecard”).

Evolution. Bran Selic et al., all members of the UML2 Partners (U2P) group, maintain the course of UML is necessarily evolution, not revolution. In its fall 2000 request for proposals, OMG provided 52 requirements for UML2, including minimizing the effect on users switching from UML1 and providing software architects the ability to model objects and connectors. Selic et al. discuss their understanding of the purpose of UML and describe the OMG requirements they consider key, including support for a family of languages with shared meaning, executable models that enable automatic code generation, and precision and clarity in the language specification.

Family of languages. Keith Duddy, representing the Australian Enterprise Distributed Systems Technology Cooperative Research Centre (DSTC) proposal, explores the challenge of building a family of languages, explaining why UML1 cannot be the foundation for that family and providing test cases for determining whether UML2 might support such a family. He argues that extensions to UML should be specified

OMG and UML

The OMG began in 1989 as the developer of the CORBA standards. In 1995, Mary Loomis and others led it to issue a request for a facility to enable the sharing of models among tools. Instead of a model-sharing facility, OMG was offered and ultimately adopted UML1, a modeling language combining and conforming a number of important modeling techniques and establishing a uniform graphical notation.

Unfortunately, while UML1 unified several of the competing schools of modeling, some equally important ideas for good object-oriented design did not influence UML1, including the responsibility-driven design concepts of Rebecca Wirfs-Brock and the elegant systems engineering viewpoint of Sally Shlaer and Stephen Mellor.

Still looking for ways to share models, OMG kept trying, ultimately adopting the XML Model Interchange (XMI) language. But tool vendors did not adopt a common interoperable version of XMI. Meanwhile, OMG revision task forces made substantial improvements in the UML1 specification through clarification and minor modifications.

In the fall of 2000, OMG requested proposals for a major revision of UML. Five groups submitted proposals in the summer of 2001, and the summer of 2002 saw the beginning of collaboration among three of the groups on a possible single submission.

OMG now expects to adopt a UML2 standard sometime in 2003. A yet unnamed standard for diagram interchange is also expected, aiming to make models truly portable. **C**

using a model of the language (a metamodel) and the OMG Meta-Object Facility (MOF), or a metamodel extension capability in a UML tool. The combination, he writes, of reusable concepts in UML2 and the first-class MOF extension mechanism will allow the currently disjoint OMG modeling languages—CWM, EAI, and EDOC—to be brought back into the UML language family.

In addition to requiring a first-class extension mechanism, OMG requires inclusion of the UML1 extension mechanism—profiles—as a way for users to customize UML in tools lacking complete meta-modeling capability. The DSTC proposal calls for abolition of profiles.

The UML2 and MOF2 standards together will certainly

enable a first-class language extension mechanism in modeling tools, though tool vendors will each determine which extension mechanism its users actually get.

OMG is also working on a suite of standards to increase the value of UML models in system development. The Model Driven Architecture (MDA) it adopted in September 2001 is not a framework for implementing distributed systems, as is CORBA (OMG's Common Object Request Broker Architecture). MDA is a framework for developing standards, as are the OMG Object Management Architecture (OMA) and Reference Model of Open Distributed Processing (RM-ODP, X.900). MDA standards enable the use of models throughout the development process to improve software quality while reducing life cycle cost. MDA is another small step on the long road to turning the craft of building information systems into

The tension is between OMG members who want to get UML out and those who want to get UML right.

an engineering discipline.

Models as assets. Stephen J. Mellor, a champion of the Unambiguous UML (2U) proposal, describes a suite of MDA tools, including model simulators and translators, that would be possible if models were truly executable, translatable to programs, and interchangeable among tools. Explaining these ideas and their roles in MDA, Mellor writes that, if a model had a clear meaning, it would be an asset, rather than a cost. He calls for a layered UML with a small well-defined, executable, and translatable kernel, along with a small set of language components automatically merged to ensure the correctness of the language specification. Countering the ambiguity of UML1, 2U is consistent in using an explicit model-merging mechanism (a technique of pattern composition and pattern application) to specify UML.

Precision and clarity. William Frank and Kevin Tyson call for a clear, clean, concise foundation for UML (the 3C proposal), describing a small set of concepts specified using the axiomatic method. 3C proposes to start with a few primitives, then specify UML using strict definitions and invariants. This approach delivers precision and clarity—without changing the look and feel of UML for either modelers or model readers.

Unlikely revolution. Finally, Dov Dori, who submitted the Object Process Methodology (OPM) proposal, calls attention to problems with the usability of UML and with the understandability of the specification. He offers a glimpse of OPM, which aims to unify the necessary modeling concepts with a single diagram type, concluding that a revolutionary change to UML is in order but highly unlikely.

Surprisingly Different

The mix of divergent goals, from modeling business processes and systems to easing the tasks of the Java programmer, from clarity and simplicity to backward compatibility, along with the competitive and secretive OMG proposal process itself, produced the five surprisingly different proposals argued and countered here. One such difference is the treatment of objects. In some of the initial proposals, objects were no longer even in the UML model; instead, they were in the system being modeled or in a formal semantic domain. Another is that one of the proposals has a

formal semantics, while the others do not. Some proposals treat object as a general category, to be used in many ways (such as for specialization to represent software objects), while others treat object as a specific to software objects. And some proposals distinguish inheritance from subtyping, while others do not.

However, instead of a winner-take-all result, the best possible outcome may yet be a joint proposal from several proposal groups. One recent sign of the desire of the OMG membership for a consensus version of UML occurred last June when the U2 Partners began collaborating with the developers of the 2U and 3C proposals, working toward a possible joint proposal that might combine the best features from each.

Meanwhile, OMG is wrestling with the highly contentious issue of whether to vote soon, adopting a particular proposal, or to require further work from the proposal writers. The tension is between OMG members who want to get UML out and those who want to get UML right. Some feel UML2 is overdue, others that the proposals are simply not ready for a vote.

Some are convinced UML will be damaged unless there is a vote this year, others that OMG should not miss a long-awaited opportunity to slip a solid foundation under UML and subsequent MDA standards.

Prospects

The craft of building information systems needs a shared language; that language needs a solid foundation and well-understood meaning, must suit the full range of system-building specialities, and must not impose unnecessary restrictions on users. A language with these qualities may be a long way off, though UML2 will certainly take us a step closer. The authors here debate the size and direction of this next step; another step will surely follow. To contribute to the development of UML, see the Wiki (public editable Web site) at www.community-ml.org.

We'll know we have what we need when, as with blueprints, topographic maps, and circuit diagrams, such debates are no longer necessary. ■

UML2 Proposal Scorecard

The five initial UML2 kernel proposals and the section authors representing them:

UML 2.0 Partners (U2P). Selic, Ramackers, and Kobryn;
Distributed Systems Technology Center (DSTC). Duddy;
Unambiguous UML (2U). Mellor;
Clear, Clean, Concise UML (3C). Frank and Tyson;
Object Process Methodology (OPM). Dori.

See the proposals at www.community-ml.org/UML2.htm ■

JOAQUIN MILLER (joaquin@acm.org) is the Chief Architect of Financial Systems Architects, New York.

© 2002 ACM 0002-0782/02/1100 \$5.00