SDN Contribution

Getting Started with UML

Summary

This article provides a short introduction into UML.

Created on: 18 April 2006

Author Bio

David Frankel's career in the software industry spans over 25 years, during which he has had vast experience as a software developer, architect, and technical strategist. He is the author of many published articles and sole author of the book Model-Driven Architecture®: Applying MDA® to Enterprise Computing, published by John Wiley & Sons in 2003. He also is lead editor of the book The MDA Journal: Model Driven Architecture Straight from the Masters, published by Meghan-Kiffer Press in 2004. He served several terms as an elected member of the OMG Architecture Board, and was intimately involved in the OMG's launch of Model Driven Architecture. He is the co-author of several industry standards, including COM-CORBA Interworking, the UML® Profile for CORBA®, and the UML Profile for EJB. He currently is Lead Standards Architect for Model-Driven Systems at SAP Labs.

Table of Contents

What is UML?

UML, the Unified Modeling Language, is an object-oriented software modeling language. It is a standard managed by the Object Management Group (OMG). The OMG has a UML resource page.

UML's Origins and Current Status

As object-oriented analysis and design (OOAD) techniques spread during the early 1990s, the OOAD industry balkanized into three camps, roughly corresponding to the followers of Grady Booch, Ivar Jacobson, and Jim Rumbaugh. Each had its own notation, methodological approaches, and tools.

In the late 1990s the camps merged to create UML. The merger began when Rational Software Corporation brought Rumbaugh and Jacobson into the company to join Booch. The three pioneers--now known as the 'three amigos,â? wrote the first UML specification. Later they decided to submit UML to the Object Management Group (OMG) for standardization.

UML has become a widely-used standard for object-oriented modeling. UML tools are available from a number of vendors. The OMG now owns the UML trademark and logo and manages the standard's evolution.

UML 1.x supports a number of kinds of models, including use case, class, collaboration, sequence, state, activity, deployment, and component models. It consists of a metamodel, a graphical syntax, and an interchange format. The metamodel defines UML's underlying metadata schema, i.e. the schema of the information that UML models capture. The metamodel conforms to the Meta Object Facility (MOF), the OMG's metadata management standard. The interchange format conforms to XMI, which is the MOF to XML mapping.

Recently, the OMG finalized UML 2.0. The most important change over UML 1.x is that all of the diagram types support hierarchical decomposition, which is critical for modeling large-scale software architectures. UML 2.0 did not fulfill all the hopes that were placed in it, but it now handles hierarchical decomposition quite well. Classes, elements of sequence models, elements of activity models, and elements of state models, and so on can all be decomposed into multiple hierarchical levels. UML 2.0 also has significantly modernized its component modeling facilities.

Tools that support UML 2.0 are coming on line. IBM Rational's new Rational Software Modeler (RSM) is a successor to its flagship Rational Rose UML modeling tool. RSM supports UML 2.0, and is a new code base that leverages the Eclipse Modeling Framework. RSM also has a sophisticated model transformation framework that can be used to create code generators. Rational Software Architect is basically RSM plus a set of out-of-the-box transformations that target the WebSphere platform.

UML's Role in Model Driven Architecture (MDA)

Since MDA is architected to support multiple modeling languages, technically speaking UML is but one possible MDA modeling language. However, pragmatically speaking, UML is the most important one, because its use is the most widespread.

UML was built to be extensible. UML out of the box does not have constructs for modeling many aspects of modern enterprise systems, including security, distribution tiers, quality of service, and so forth. UML profiles are specializations of UML created via UML's built-in extension mechanisms. The OMG is gradually standardizing various profiles, and serious enterprise projects that use UML also define their own profiles. The advantage of UML profiles is that modelers can employ them via generic UML modeling tools, rather than having to build or purchase new modeling tools to support the extensions. Most UML tools make it possible to plug in code generators that users or third-party companies tailor to their UML profiles.

Nevertheless, UML does not meet all needs, and must co-exist with other modeling languages and various kinds of metadata. UML models are one more form of metadata that enterprises have to manage in addition to database schemas, workflow models, business process models, deployment descriptors, and so on. That is why UML's conformance with MOF, the metadata management standard that lies at the core of MDA, is important. MOF is the centerpiece of MDA's goal to break down the metadata silos in our enterprises.

Kinds of UML Models

Use Case Model--Defines how external actors interact with the various functions of a system

Class Model--Defines classes at some chosen level of abstraction

Package Model--Defines dependencies among packages. Packages are mechanisms for grouping elements of UML models.

Object Model--For modeling sample or required instances of classes

Sequence Model--Defines how system components interact (i.e exchange messages)

Activity Model--Models activity flow. In UML 2.0, activity models have Petri-net semantics

State Model--Defines the state machine of an object

Deployment Model--Defines how software modules are deployed on physical resources

Component Model--Defines components that provide and consume interfaces

SAP has some UML resources:

UML course: This course has the objective to impart UML knowledge and to promote a specific approach to using UML at SAP. See the course description, course dates, and course materials.

Visio shapes for UML and SAP block diagrams: The UML specification itself merely mandates UML shapes, but not their look and feel beyond that. SAP has created its own shapes reflecting the SAP preferred look and feel. These Visio shapes can be down-loaded from the SAP Vision corner at Ency. Note that these shapes and shapes for SAP block diagrams are available only for Visio, not for other modeling tools.

Disclaimer and Liability Notice

