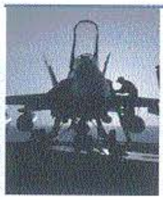
2002

Model Scheme A Good Fit for C4ISR

Andreas Tolk
Old Dominion University, atolk@odu.edu



COTS

View

Model Scheme A Good Fit for C4ISR

C4ISR apps already enjoy many benefits from commercial standards. The OMG's Model Driven Architecture could raise that trend to a new level.

Andreas Tolk, Ph.D., Senior Research Scientist, Virginia Modeling, Analysis & Simulation Center, Old Dominion University

One vivid example of the success of the COTS initiative is the considerable number of open and commercially driven standards that comprise the DoD's Common Operating Environment (COE). COE is the software component architecture mandated for use when building C4ISR systems.

What has yet to find its way into COE is the sophisticated and widely accepted standard for describing software solutions called the Unified Modeling Language (UML). UML is controlled by the Object Management Group (OMG), the leading standards organization in the software community. To be able to understand the potential of UML, it's important to look into a new element now in the process of incorporation within UML called the Model Driven Architecture (MDA).

The MDA (Figure 1) is based on the idea of Meta-modeling. The MDA actually goes beyond the scope of UML. It merges the different OMG standards, having been developed and used separately so far, into a common view by applying common Meta models to them. The OMG standards that will become part of the MDA include the Extensible Markup Language (XML) and the XML Metadata Interchange specification (XMI), the Unified Modeling Language (UML), middleware solutions supporting CORBA, Sun's Enterprise JavaBeans, or Microsoft's DOT-NET(.NET), the Common Warehouse Metamodel (CWM), the Meta Object Facility (MOF) and many more.

Neutrality Is Key

The central idea in the MDA is the use of a common stable model, which is language-, vendor- and middleware-neutral. This model must be a meta-model of the concept. The MDA offers concepts for such a model. With such a model in the center users gain the ability to derive code for various sub-levels. Even if the underlying infrastructure shifts over time, the meta-model remains stable and can be ported to various middleware implementations as well as platforms. To do so, the MDA defines an approach to system specifications that separates the specification of the system functionality from the specification of the platform specific implementation.

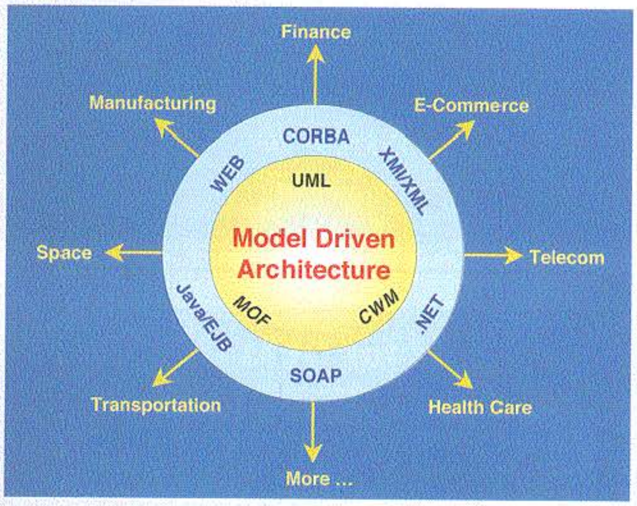


Figure 1

The Model Driven Architecture merges different OMG standards, such as UML, XML and so on, into a common view by applying common Meta models to them. Existing standards for Finance, Manufacturing, E-Commerce and many more are served by the architecture.

The first step when creating an MDA-based application is to create a Platform-Independent application Model (PIM). In the MDA, a model is defined to be a representation of a part of the function, structure and/or behavior of a system. The PIM is expressed in UML in terms of the appropriate core model. The core models are available in the form of UML Profiles, a number of which are already well along their way to being standardized by the OMG.

The next step—assuming the model is to run as an application—is to convert this model from general application to a Platform-Specific Model (PSM). The PSM is derived from the

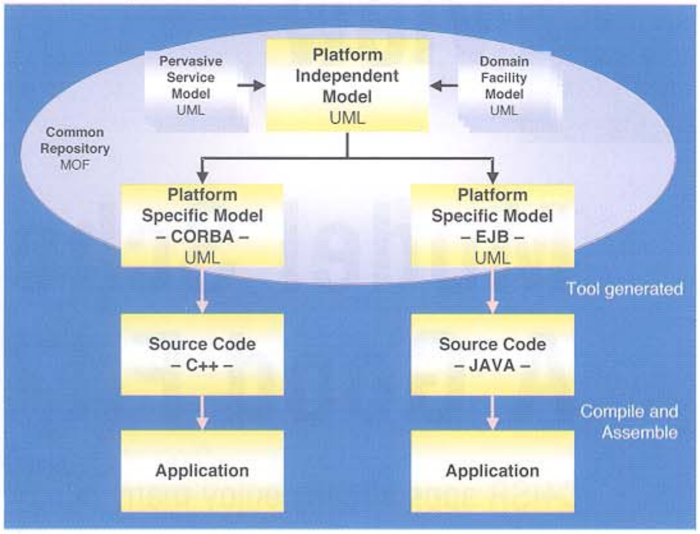

Pervasive Service Model UML
Platform Independent Model UML
Domain Facility Model UML
Common Repository MOF
Platform Specific Model – CORBA – UML
Platform Specific Model – EJB – UML
Tool generated
Source Code – C++ –
Source Code – JAVA –
Compile and Assemble
Application
Application

Figure 2

The process of creating a Model Driven Architecture-based application involves two steps. First, a Platform-Independent application Model (PIM) is created expressed in UML in terms of the appropriate core model. The PIM is then converted from general application to a Platform-Specific Model (PSM).

PIM using standardized transformation rules. While the PIM defines the necessary functionality, the PSM specifies how this functionality is realized on a special platform.

The idea is not to develop a Platform-Independent application Model (PIM) from scratch. In fact, the concept to standardize basic functionality and reuse the findings of respective CORBA-based standardization groups found its way into the MDA in the form of Domain Facilities Models and Pervasive Services Models. The Domain Facilities Models are reusable standard frameworks for standard functions in their application space.

There are already standards for Finance, Manufacturing, E-Commerce and many more. As Figure 1 shows, some of the domains are enumerated as targeted areas. Pervasive Services are the basic services being used for every distributed application, such as finding objects, naming, routing, security issues and many more. When developing the PIM for an IT solution, all this knowledge can be reused, which reduces the development risk, the necessary test efforts and so forth.

Broad Tool Support for MDA

Thanks to broad industry support within OMG, there's likely to be a large number of tools supporting MDA in the short term. And because UML, XML and XMI are all part of the MDA, all tools and applications for those standards are automatically part of the MDA world already. Clearly most of the steps following the development of the PIM will be supported by computer-based tools.

That means the Platform-Specific Model (PSM) will be derived automatically from the PIM, the source code will be generated automatically from the PSM, and the application will be compiled and assembled automatically as well. In other words, after having created the PIM, the user chooses his Platform (for example a web-based XMI environment), his language (for example Java), and the rest is done by the MDA development system. Figure 2 exemplifies the application of the MDA.

As described earlier, the knowledge about an application domain is captured in respective Domain Facility Models. This is done by Domain Task Forces (DTF) of OMG members. Each DTF produces standard frameworks for standard functions in its application space inheriting the work already done in the CORBA community. Each DTF uses the same language to specify their model of their operations.

CORBA Already Making its Mark

Granted it's unlikely that non-military partners will use other standards than the ones that are commercially supported. It is therefore likely that potential partners in operations-other-than-war (OOTW) or in homeland security operations will use models that will be mapped to respective PIMs sooner or later. Many IT systems used by the police departments and fire departments today are CORBA-based, the telecommunication world belongs to the supporters of the OMG, and the IT support of the Centers for Disease Control and Prevention will be based on open and commercially supported standards as well.

It's not only the technical aspects that are interesting when dealing with the MDA. Only very recently, the Cebrowski Institute for Information Innovation and Superiority (CINFINIS) initialized a course on Critical Infrastructure Protection. Affiliated with the Naval Postgraduate School in Monterey, California, this new institute is focusing on homeland security challenges for computer science. In the referenced course, they use the definitions and requirements of the Department of the Navy. The Navy has subdivided its enterprise infrastructure into ten major areas: Logistics, Space, Command-Control-and-Communications, Transportation, Information Infrastructure (Computers and Networks), ISR (Intelligence, Surveillance and Reconnaissance), Public Works, Personnel, Health and Finance.

The course makes the point that all those sectors depend not only upon Navy and Marine Corps capabilities, but upon the Federal, State, Local and Commercial sectors. The interdependency of many aspects of that infrastructure is recognized, but poorly understood. The domain facility models of the MDA can be applied there. All the major areas of the critical infrastructure can be mapped nearly one-to-one to the domain facility models.

The next generation of IT/C4ISR systems is one command and control system based on heterogeneous information techniques. What's needed is a paradigm of component-based software engineering to merge components of various application domains to improve the IT support for the Warfighter. The technical means to make this vision become reality are at hand and the military requirements are formulated. Domain-specific stove-piped solutions should finally become something of the past. Legacy software applications can be migrated by reverse engineering. New applications can be realized in a well defined, well documented, and configurable and flexible way. The Model Driven Architecture has the potential to become a valuable framework for future C4ISR migration and development. ▌▌

Virginia Modeling, Analysis & Simulation Center
Old Dominion University, Suffolk, VA.
(757) 686-6203.
[www.odu.edu].