OBJECT MANAGEMENT GROUP

Model Driven Architecture (MDA) FAQ

Q: What is the Model Driven Architecture (MDA) and how is it different from other architectures?
A: The MDA is a new way of writing specifications, based on a platform-independent model. A complete MDA specification consists of a definitive platform-independent base UML model, plus one or more platform-specific models and interface definition sets, each describing how the base model is implemented on a different middleware platform.

The MDA focuses primarily on the functionality and behavior of a distributed application or system, not the technology in which it will be implemented. It divorces implementation details from business functions. Thus, it is not necessary to repeat the process of modeling an application or system's functionality and behavior each time a new technology (e.g., XML/SOAP) comes along. Other architectures are generally tied to a particular technology. With MDA, functionality and behavior are modeled once and only once. Mapping to the supported MDA platforms will be implemented by tools, easing the task of supporting new or different technologies.

Q: Why is the OMG going in a new direction? What prompted it?
A: It's not really a new direction, when you take OMG's history into account. In 1997, OMG expanded its scope to include modeling with the adoption of the Unified Modeling Language (UML) and Meta-Object Facility (MOF). Although it has always been true that UML models can be implemented on *any* platform, the continuing proliferation of middleware "silver bullets" suggested that a platform-independent UML model is the secret to software stability and ROI – a stake that remains fixed in the ground while the infrastructure around it shifts over time. The MDA unites OMG's well-established modeling standards with not only CORBA but also every other middleware technology – past, present, and future – to integrate what you've built, with what you're building, with what you're going to build. Rather than focusing on yet another "next best thing," MDA raises the bar and *designs* portability and interoperability into the application at the model level.

Q: What is the role of UML in the MDA?
A: UML is the key enabling technology for the Model Driven Architecture: Every application using the MDA is based on a normative, platform-independent UML model. By leveraging this universally accepted modeling standard, the MDA allows creation of applications that are portable across, and interoperate naturally across, a broad spectrum of systems from embedded, to desktop, to server, to mainframe, and across the Internet.

Q: What is the role of middleware platforms in the MDA?
A: In the MDA, a specification's platform-independent base UML model is used to define one or more platform-specific models and interface definition sets, each describing

how the base model is implemented on a different middleware platform. Because the base model, platform-specific model, and interface definitions will all be part of an MDA specification, OMG will adopt specifications for multiple middleware platforms as it puts this new architecture into place. While CORBA's platform and language independence and proven, deployed transactional and secure nature continue to make it the best choice for systems from embedded to desktop to Internet, MDA makes porting to, and interoperating with, other middleware platforms easier and cheaper.

Q: What about CORBA, then?
A: OMG continues to promote and develop CORBA, and the CORBA market continues to expand, particularly in real-time & embedded systems, and the large, mission-critical, fault tolerant systems essential to enterprise computing. Because CORBA is the only multi-platform, multi-language solution for systems integration, many enterprises will use CORBA to build and integrate applications defined in the MDA.

OMG and its member companies have always recognized the value of interoperating with other standards as well as proprietary platforms & languages. OMG created the COM/CORBA interoperability specification in 1995 (for connections to Microsoft's proprietary desktop systems) and expanded it in 1997, and designed and constructed the many ways that CORBA works with Java and XML. Amongst its other benefits, MDA continues this work of defining cross-middleware interoperability, and will provide tool support to speed and automate the process. This will be a great benefit to users who find themselves supporting multiple middleware platforms.

Q: How does MDA enable cross-platform interoperability?
A: In the MDA, interfaces and implementations of a specification all derive from a common base UML model. This structure of linked models allows automated building of the bridges that connect implementations on those various middleware platforms. And, when the base model for a new specification is being designed, interoperability with other specifications and services can be designed into it.

Q: How does MDA compare or compete with Microsoft's .NET or Sun's ONE?
A: MDA works at a different level than .NET and ONE. These are platforms, aimed at specific albeit broad application targets, while the MDA is (as its name declares) a *Model Driven Architecture* that works above the level of *every* middleware platform, .NET and ONE included. A middleware platform is incorporated into the MDA as a platform-specific profile. As ONE and .NET establish market share, OMG members will define platform-specific profiles for them, allowing them to participate in the MDA along with the other platforms which will almost certainly include Java/EJB, XML and one or more protocols dictated by the industry or the marketplace (SOAP or XP), and others.

Q: Who is responsible for the MDA?
A: Although the original impetus for the MDA came from OMG staff, it is now supported by the membership as demonstrated by unanimous votes of the technical representatives attending the organization's meeting in late February, 2001. Like all the other work of the OMG, MDA was defined and will be developed by the OMG

membership which includes a diverse cross-section of computer vendors, software suppliers, and many end-users. The wealth of experience contributed by these hundreds of organizations is one of the great strengths of OMG's process, and has been put to good use in defining and refining the MDA. The initial vision was drafted in late 2000 in a paper available at http://doc.omg.org/mda, and subsequently refined with the help of many individual contributors into a technical perspective, available at http://doc.omg.org/ab/1-2-4..

Q: What are the top three benefits of the MDA to enterprises trying to cope with today's computing environment?
A: There are many benefits to using the MDA approach, with the most important being:

- An architecture based on the MDA is always ready to deal with yesterday's, today's and tomorrow's "next big thing".
- The MDA will make it easier to integrate applications and facilities across middleware boundaries.
- Domain facilities defined in the MDA by OMG's Domain Task Forces will provide much wider interoperability by always being available on a domain's preferred platform, and on multiple platforms whenever there is a need.

Q: How will the MDA be delivered? In what kind of tools? And when?
A: Several key parts of the MDA vision have already been standardized, including not only the UML, MOF, XMI and CWM, but also the first middleware mapping (to OMG's own CORBA). Several other major MDA foundation specifications are "in the chute," including a middleware-independent mapping for enterprise systems (called "UML for Enterprise Distributed Object Computing").

In terms of products, MDA will be implemented by modeling tools and we expect the first generation to emerge late this year. Additional vendors' products will join these soon after, so that almost all OMG vendor members (and many non-members) will be represented in the marketplace by products in about eighteen months.

The biggest benefit of MDA will be the generation of application code from an MDA model through an automated or semi-automated series of steps. Although fully automatic code generation is unlikely for some platforms, examples with limited scope are running today and demonstrate the practicality of this vision. MDA tools will initially move beyond modeling with the generation of code for interfaces (such as OMG IDL), for functionality constrained by a specification such as the CORBA Component Model or EJB, for wrappers around programmer code that make it transactional or secure, and for operations that get and set the values of variables declared in the model. A subsequent version may code execution of simple business rules.

Q: How is OMG doing, anyways?
A: OMG is bigger than ever, and doing well. With about 800 member companies, OMG continues to be the largest software standards organization of its kind. There are more systems deployed using OMG's standards than ever, with new success stories appearing

daily. Some recent examples include major design wins at a large airline reservation system and two of the world’s biggest multinational automobile manufacturers.

In terms of the OMG standards process, there are now more adoptions in process than at any other time in OMG’s twelve year history. Our meetings, which typically attract more than 500 members and guests, regularly host industry workshops and co-host meetings of other organizations.

Q: Will MDA adversely impact the CORBA-based product I've installed or plan to install in the near future?
A: Absolutely not. First, OMG plans to continue support for CORBA at current levels at least; demand from CORBA users in realtime, embedded, fault-tolerant, and enterprise systems will actually increase the tempo of CORBA standardization. CORBA will also be one of the most prominent platform-specific models in the MDA. MDA will make it practical to either keep all of your CORBA applications and connect to other platforms, or to port to them, as dictated by business factors.