

Why do we need standards for the modernization of existing systems?

OMG ADM Task Force

There is a vast amount of highly functional, operational software representing enormous commercial value deployed in organizations around the globe. To be precise, existing “systems are defined as any production-enabled software, regardless of the platform it runs on, language it’s written in, or length of time it has been in production”¹. These entrenched software systems often resist evolution because their strategic value and ability to adapt has diminished through factors not exclusively related to its functionality. Common examples of such factors are a system's inability to be understood or maintained cost-effectively, inability to interoperate or dependence on undesired technologies or architectures. There is, therefore, a *need* to understand and evolve existing software assets for the purpose of:

- Software improvement
- Modifications
- Interoperability
- Refactoring
- Restructuring
- Reuse
- Porting
- Migration
- Translation
- Integration
- Service-oriented architecture deployment

Collectively, these activities can be defined as Architecture-Driven Modernization—or ADM. ADM is the process of understanding and evolving existing software assets. ADM restores the value of existing applications because it extracts and leverages the investment in the intellectual property of entrenched software to deliver new solutions that address changing business requirements. ADM is used when existing IT practices fail to deliver against business objectives.

Modernization, within the context of the ADM OMG initiative, is architecture-driven because the task force wanted to emphasize the need to capture and retool various architectural aspects of existing application environments. This does not preclude source-to-source migrations (where appropriate), but encourages user organizations to consider modernization from an analysis and design based perspective. In doing so, project teams will ensure that obsolete concepts or designs are not propagated into modern languages and platforms. The bottom-line results deliver modernized systems that conform more effectively to current business practices and strategic requirements.

Business value of modernization

Existing software systems, which tend to reflect conflicting or confusing designs and laborious manual coding methods, can significantly hinder the business agility of an organization. Companies needing to contend with mergers and acquisitions, new product deployment, changing market requirements, governmental regulations and other business demands require nimble software infrastructures to compete – or even survive. Modernization can also be the key to reducing the operational cost.

¹ William M. Ulrich, “Legacy Systems: Transformation Strategies”, Prentice Hall, 2002

For a large number of organizations, the need to modernize enterprise information systems is typically triggered by certain external events (e.g., the need to provide Web-access to an existing application or the need to port away from an unsupported platform). However, for an enterprise in a technology-intensive environment or industry, the evolution of the existing software base is a vital part of day-to-day requirements.

ADM provides the means to address these requirements using controlled, sustainable and low risk approaches. The main benefits to be achieved as part of the ADM efforts include:

- Enabling business agility by creating software agility
- Improving ROI in existing software initiatives
 - Improving productivity of software development
 - Reducing maintenance effort and cost
- Extending the useful life of existing applications

Interoperability benefits from standardization

A strong cross-section of software tool vendors and service providers has emerged to enable the modernization of existing systems. Unfortunately, users and vendors have been working in isolation, often reinventing the wheel. Standardization of modernization results and processes build on prior experiences and best practices. ADM specifications and corresponding tools will enable projects with large and/or aging software portfolios to become more agile.

There is a need for standardization in modernization projects to enable integration and interoperability among solutions from multiple vendors. Standardization will increase interoperability between different tools and manual processes by creating an open framework. This will enable a new generation of solutions to benefit the whole industry and encourage collaboration among complementary vendors.

ADM standards will allow users to begin modernization projects knowing that there is interoperability among different tool vendors. In other words, standardization will ensure that end users are investing not just in individual tools but rather in a coordinated modernization strategy.

Long-term benefits: Adoption of new technologies, practices and paradigms

There are also long-term strategic reasons to work towards standardization of the modernization of existing systems. In the past, the forward engineering industry has largely ignored existing software assets, while reverse engineering was in turn ignoring new forward engineering methods. Over time, this has created a significant gap between modernization tools and methodology support for “green-field” projects and projects involving maintenance and evolution of existing systems. ADM will help break the barriers between forward engineering tools and reverse engineering tools and help organizations with existing software assets realize the benefits of modern forward engineering technologies, such as the Unified Modeling Language (UML) and the Model-Driven Architecture (MDA).

MDA provides an open, vendor-neutral approach to the challenge of interoperability, building upon and leveraging the value of established modeling standards. MDA aims to separate business or application logic from underlying platform technology. MDA advocates a new way of developing software in which the platform-independent model becomes the long-term maintainable artifact. As a result, platform-independent applications can be deployed on a range of platforms including CORBA, J2EE, .NET, and Web Services using automated, model-based compilers. As new platforms and technologies emerge, MDA ensures “future-proofing” of development investments. In this way, MDA goes beyond middleware to provide a comprehensive, structured solution for application interoperability and long-term portability.

Platform-independent MDA models are an important target for ADM. For organizations, which have adopted MDA, the modernization process is no longer just moving from one language to another, or from one platform to another. ADM, in the context of MDA, creates the model of the code for the purpose of importing it into an MDA-enabled development environment. ADM related standards become essential in achieving successful adoption of modern forward engineering technologies, such as MDA and UML, across large-scale software environments.

Bottom-line benefits of standardization

Standardization of the software modernization process will help businesses reduce the risk of undertaking software improvement initiatives by lessening the time, risk and cost of software transformations, improving the effectiveness of reverse engineering and redeployment tools, and extending the ROI on software development projects.

Why now?

- The current economy requires organizations to be smarter with existing resources, including the reuse of existing software assets.
- “From scratch” development, non-invasive integration, traditional maintenance and off-the-shelf packages have limits on their ability to deliver strategic value to a business.
- Commoditization of middleware and hardware is driving the interest for porting to new platforms.
- The industry is generally accepting UML and MDA as development paradigms and this provides a target for ADM projects.
- Technologies for understanding, reusing and modernizing existing software are maturing and reaching a critical mass as far as vendor solutions.
- There has been a lag between techniques and methodologies that requires companies to reexamine current solutions and practices (e.g. evolution of systems that have grown from initial prototypes, mergers or acquisitions).
- Industry is seeing a growing need for interoperability of enterprise software that cannot be achieved through middleware solutions.

Why OMG?

The Object Management Group (OMG) is an open membership, not-for-profit consortium dedicated to producing and maintaining specifications for interoperable enterprise applications. OMG membership roster includes many of the most successful and innovative companies in the computer industry, as well as those at the forefront of using technology to gain a competitive edge in their business. Participants have made the commitment actively to participate in shaping the future of enterprise, Internet, real-time and embedded systems.

OMG has developed some of the industry's best-known and most influential specifications, including CORBA, OMG IDL, IIOP, UML, MOF, XMI, CWM, the OMA and Domain Facilities in industries such as healthcare, manufacturing, telecommunications and others. The Model Driven Architecture (MDA) builds on these successes, providing a comprehensive interoperability framework for defining the interconnected systems of tomorrow.

Mission statement

The mission of the ADM Task Force is to create specifications and promote industry consensus on the modernization of existing applications. The main goals of the ADM Task Force include:

- Revitalizing of existing applications
- Making existing applications more agile

- Leveraging existing OMG modeling standards and MDA
- Consolidating best practices leading to successful modernization projects

The central part of the specification is a standard repository structure (meta-model) for representing existing software assets. This meta-model will allow organizations to quickly assimilate vendor captured information, extracted from existing software assets across a variety of platforms, into a cohesive platform independent view. This information may be augmented by business-process tools or analyst input, transformed into more reusable or understandable forms, and/or ported into forward engineering paradigms.

The specification will also associate key metrics with the repository and highlight best practices of the ADM process in such a way that the user can select an appropriate path that suites the particular project needs. The specification will additionally support mapping transformations from existing to target architectures.

The proposed standardization initiative will leverage OMG's existing standards and the Model Driven Architecture approach for:

- Model(s) to support the ADM process
 - Model to represent software assets at different levels of abstraction
 - Model to represent target architectures
 - Model to represent transformations to improve and target architectures
- The modernization process and required steps
 - Knowledge discovery at various levels:
 - Language syntax
 - Design
 - Architecture
 - Functionality (Business logic, scenarios, use cases)
 - Identifying anomalies in existing application
 - Improving existing systems
 - Migrating

Timeline & Next Steps

Come join us in this effort. This Task Force issued its first RFP in London. Ongoing work will continue throughout 2004 as vendors and user organizations collaborate on this important effort.