A Status on OMG Architecture-Driven Modernization Task Force

William Ulrich
President, Tactical Strategy Group, Inc.
wmmulrich@cs.com

Abstract

This paper provides an overview of existing systems modernization and discusses what the Architecture-Driven Modernization task force is doing to facilitate application meta-data interchange standards, as well as promote modernization in general. Multiple meta-models are being developed to represent application meta-data from multiple perspectives. This paper also communicates the overall roadmap that the ADM task force is using to guide development of a series of modernization standards.

Introduction

The Object Management Group (OMG) Architecture-Driven Modernization (ADM) task force is currently engaged in a series of standards development activities. The mission of this task force is to further the ability of organizations to successfully modernize existing information systems. Existing systems modernization is the process of understanding and evolving existing software assets to further one or more business or organizational objectives.

Existing Systems Modernization

Modernization is an IT augmentation strategy and typically begins where existing practices fall short. Existing practices that either have or are beginning to fall short include ongoing maintenance and outsourcing, new development as a wholesale replacement strategy, commercial off-the-shelf (COTS) package options and application integration via middleware wrappering. Modernization examines, exposes and facilitates the refactoring, redesign and redeployment of core application architectures with the intent of meeting critical business requirements in a way that lowers risks, costs and delivery timeframes.

Modernization supports a series of initiatives (i.e., scenarios) that include, but are not limited to, portfolio management, application quality improvement (i.e., standardization and modularization), source-to-source conversion, platform migration, systems consolidation, service oriented architecture (SOA) migration, model-driven architecture (MDA) migration, application integration and data re-architecting.

There are a wide variety of possible tasks that fall under the modernization umbrella that are used to enable various modernization scenarios. These tasks can generally be categorized into assessment, stabilization and transformation tasks.

Assessment tasks include the analysis of business processes and their impact on systems, exposing application architectures and interrelationships, detailed analysis of source code including processes flows, structure, complexity and anomalies, data and data architecture analysis and examination of semantic business rules buried within systems. Additional assessment tasks include the mapping of existing business, data, application and technical architectures to target architectures and business requirements.

Stabilization and standardization tasks make existing systems more reliable and adaptable. Such tasks are used to rationalize, structure, realign, modularize or otherwise refactor existing applications. The value of such tasks can help companies streamline systems in preparation for business rule extraction and reuse, application consolidation, migration to SOA or MDA, or other transformation scenarios as required.

Transformation tasks include the extraction of existing data definitions, data and business rules, visualization of existing system artifacts for purposes of redesign and the reuse and importation of data and business rules into a target architecture. Target architectures may vary, but include MDA, SOA, .NET or J2EE, or other target options that may emerge.

ADM Task Force Strategy

The ADM task force is being driven by various vendor and user organizations seeking to share tool and analyst captured meta-data from existing systems environments.

This requirement is driven by the fact that many companies and systems tend to use multiple languages and technologies. For example, a company may have systems deployed in COBOL, C, C++, Easytrieve and Assembler. That company would need to use 2 or 3 tools to parse and capture information about those systems. Each tool, however, would individually not be able to integrate captured meta-data in common view or format.

Through the use of a common meta-model, however, each tool would be able to exchange common views across platforms and languages for the purpose of analyzing, standardizing and transforming existing systems.

The ADM task force has termed this meta-model the Knowledge Discovery Meta-model (KDM). The KDM is the first standard being requested via a request for proposal (RFP) by the task force. A second RFP will request an extension to the KDM that will facilitate the exchange of procedural meta-data at the statement / action level.

An ADM roadmap was developed by the task force to outline a series of RFPs that organizations respond to in order to create a particular set of standards. These RFPs address the creation, evolution and use of the KDM.

RFP One: KDM Creation

The first RFP requests a KDM that contains meta-data describing the structural and data representations of the as-is system architecture along with information that can be derived from an as-is view. The resulting standard will include a number of key requirements based on the RFP.

1. KDM shall be MOF 2.0 compliant, represented by UML diagrams and delivered in an XMI and a document format.
2. KDM shall represent principal artifacts of existing software as entities, relationships and attributes. The KDM shall not be restricted to the artifacts of the existing software itself, but should also be able to represent external artifacts with which the software interacts.
3. KDM shall consist of a platform and language independent core and define a single unified terminology for the domain of knowledge discovery about existing software assets.
4. KDM shall support a wide range of the major platforms and languages such as COBOL, C, Java, Ada, z OS, Unix, J2EE or .NET, be able to represent multiple heterogeneous systems composed of software artifacts based on different platforms and/or languages, and be extensible by the user.
5. KDM shall describe the physical structure of the existing software and the logical structure that can be used to aggregate and/or modify the physical structure of the software artifacts, such refactoring. It should also facilitate the tracing of any entity, relationship or attribute from a logical structure back to physical artifacts.
6. A restriction on the first requirement is that KDM shall be able to represent behavioral programming artifacts down to, but not below the procedural level.

In meeting these requirements, a responder to RFP #1 must demonstrate the capability to map the existing software artifacts in a common implementation language, such as C, C++, Java, Ada or COBOL, into a repository that can be described by the KDM.

Responders must further demonstrate the ability of the KDM to support more than one implementation language and show the usability of the proposed meta-model for the purpose of viewing system meta-data, generating informational reports and manipulating meta-data within the repository.

The decision to limit the KDM, as stated in item six above, was due to pragmatic reasons. The ADM task force wanted to phase the creation of the KDM to facilitate the delivery of an initial set of standards in a timely manner.

In order to balance the need to deliver an initial set of standards with the agreed upon need to be able to represent actions being taken by a system below the procedure level, the task force is rapidly moving forward with a second RFP to address a fully detailed KDM. This is discussed in the section that follows below.

There are several responses designed to meet the above requirements to RFP #1 that are currently under review. Each response addresses these requirements in various ways. In general, these responses have created several meta-models that address the above requirements through a tiered, multi-view approach.

For example, one response provides a KDM view from an “execution” perspective as well as from a “build” perspective. The execution perspective exposes

software artifacts from a run time view so a user can see how the system flows, how transactions are triggered and how execution artifacts interact. The build perspective, on the other hand, defines and depicts system artifacts from a source, executable and library standpoint.

Providing multiple views serves to simplify the overall KDM for purposes of usability and understandability. Other views include design views, conceptual views, security views, data views and scenario views.

Some tools may not be able to populate certain views within the KDM repository and need to be augmented by other tools. For example, a give tool may not be able to fully assess the execution flow of an application while another tool may provide this information. One tool may be limited to certain platforms or languages, while a second tool may not.

Used collectively, a set of tools can provide a much broader set of application meta-data than any given tool working alone. The KDM, therefore, offers a vehicle for users to create a more complete, more comprehensive view of one or more applications for the purposes of understanding, refactoring and / or redesigning and redeploying those applications.

One major challenge involves storing a variety of languages, database structures, middleware, teleprocessing monitors and platforms in a common repository. The task force understood that this would be difficult, yet established a requirement to create a core set of entities to represent multifaceted applications in a common view. Where this is not possible, a responder can extend the KDM to include language or platform specific entities and relationships.

Efforts are currently underway to identify common and unique concepts across the response documents and determine if these responses can be consolidated into a common standards submission. This effort will continue into the latter part of 2004.

RFP Two: KDM Expansion

The second RFP, which is currently in development, expands KDM definitions to the statement level. These extensions will facilitate derivation of processing rules and state transitions. This RFP addresses the challenge of creating a common view of procedural statements that modify data, direct processing flow or perform some other program level task.

Requirements include the need to find a common representation for a variety of languages that include variables, action statements, condition statements, data transference, external communications or other actions. Responding to this RFP may require implementation specific extensions, as are accommodated in the initial ADM RFP.

An additional challenge associated with this second RFP involves the creation of a generic view of granular programming statements that could dramatically increase the number of artifacts to be stored within the KDM. Once the KDM has been expanded to incorporate this more granular view of application artifacts, additional RFPs are envisioned to further facilitate modernization initiatives.

Future RFPs

Subsequent RFPs have not been fully defined or vetted by the task force, but discussions have included a KDM refactoring ability, the derivation of KDM metrics and forward mapping to target models and platforms.

KDM factoring is the concept of taking a populated KDM repository and manipulating artifacts in a way that clarifies how one or more applications might be modularized, restructured, rationalized, consolidated or otherwise retooled. The purpose of these activities is to refactor one or more applications into an application that can more readily support changeability, adaptability or subsequent migration requirements.

For example, if an organization wishes to migrate to a services oriented architecture, older application architectures do not lend themselves to such an effort because business logic is intertwined with user interface and data access logic. The KDM, coupled with various tools, facilitates the analysis required to determine how this refactoring can occur.

In addition, because the KDM represents physical and logical assets at various levels of abstraction and offers a common interchange format for modernization tools, refactoring and redesign tools can access the KDM to recreate a more streamlined set of application artifacts.

In addition to creation and expansion of the KDM, a subsequent RFP envisions metric derivation from the KDM. There are a limited number of metrics currently derivable from existing application environments. This includes, for example, program level complexity and structural metrics.

A wealth of additional metrics could be derived from existing application environments, however, to provide summary level insights into the overall difficulty a

modernization team would encounter in various types of projects. These metrics include system level metrics, data level metrics, execution metrics, functional conformance and quality metrics, and architectural metrics.[1]

For example, if there are 80 hard-coded definitions of a record layout for a file, and there should only be one reusable set of definitions for that file, analysts, programmers and modernization planners would want to know this.

The implication of this degree of redundancy is that adding a data element to a file or database table or segment would entail much more work if it had to be replicated 80 times. It would also increase the likelihood of introducing problems or inconsistencies, particularly if a record layout is not consistently defined across its various incarnations.

In addition, a flat file to relational data migration may encounter significant roadblocks if these 80 definitions are conflicted across applications and business areas. A simple definitional redundancy metric could highlight the difficulties and risks in dealing with this data. There are many such metrics that can be derived from the KDM, yet most tools have not incorporated such metrics.

The most significant ADM RFP envisioned over the long-term involves the mapping of existing system artifacts in the KDM to target architectures and environments. These targets can vary but would necessarily include model-driven architecture (MDA), UML models or other design paradigms as appropriate based on target environments.

These mappings imply a transformational process that would allow one to move from the current data, application and technical architecture to a redesigned data, application and technical architecture. This is the ultimate goals of the ADM task force, although great value will be derived from the exposure of existing physical and logical architectures via the initial KDM.

Related ADM Initiatives

One initiative being pursued by the ADM task force that is not directly tied to the creation of interoperability standards involves the identification of various modernization scenarios and how these scenarios can utilize the KDM. A second initiative includes the identification of certain modernization processes that facilitate the use of the meta-data in the KDM for various modernization scenarios.

The scenarios discussed earlier in this paper, along with their relationship to the KDM, are being outlined in a white paper. Scenario definition will help the ADM task force ensure that the KDM specification addresses the practical objectives for various modernization projects in a variety of industries. Scenario topics will likely be added over time.

A second initiative involves the identification of certain modernization processes that can provide guidance to project teams attempting to analyze, improve and transform existing application and data architectures. While creating processes or methods is not under the direct charter of the OMG, the ADM task force believes that pointing to best practices or processes for modernization projects will greatly improve the usability of the KDM on actual projects.

Conclusion

In conclusion, existing systems modernization is becoming more and more vital to corporations and government organizations worldwide. As a result, the tools and processes used to facilitate modernization projects should be governed by standards.

The ADM task force developing interoperability standards through the KDM that allow existing application architectures to be exposed, refactored and transformed to new target architectures. .

Bio:

William Ulrich is President of Tactical Strategy Group, Inc., a management consulting firm established in 1990. His specialty involves planning and mentoring systems modernization projects. Past clients include major corporations and government institutions. As developer of The Systems Redevelopment Methodology (TSRM/USRM), Mr. Ulrich defined many of the standards and metrics being used in legacy modernization and transformation projects today.

Mr. Ulrich served on the faculty of Northeastern Illinois University and has facilitated numerous workshops on software reengineering. He has also written hundreds of articles and several books, including *Legacy Systems: Transformation Strategies* (Prentice Hall). Mr. Ulrich co-chairs the OMG Architecture-Driven Modernization task force. Prior to founding TSG, he served in a senior management capacity at KPMG and held various other positions in the IT industry.

[1] *The Metrics Guide*, http://www.comsysprojects.com/SystemTransformation/tmmetricguide.htm