Object Management Group

First Needham Place
250 First Avenue, Suite 100
Needham, MA 02494

Telephone: +1-781-444-0404
Facsimile: +1-781-444-0320

Request For Proposal

Architecture-Driven Modernization (ADM): Knowledge Discovery Meta-Model (KDM)

OMG Document: lt/03-11-04

Letters of Intent due: 15th February 2004
Submissions due: 31st May 2004

Objective of this RFP

There is a vast amount of useful, deployed, operational software representing an enormous commercial value. A significant amount of production software often resists evolution because its strategic value and/or its 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, its ability to interoperate, or its dependence on undesired technologies or architectures. There is a need to understand and evolve existing software assets for the purpose of documentation, improvement, modification, interoperability, porting, migrations, reuse, redesign and/or redeployment. These activities are collectively defined as software "modernization".

In addition, there exists a significant gap between tools and methodology support that is available for "green-field" projects and for the projects, which involve maintenance and evolution of existing code bases.

A large industry of software tool vendors and service vendors exists to enable software modernization. There is a need for standardization in software modernization that will enable integration and interoperability between solutions from multiple vendors. Standardization will increase interoperability between different tools by creating an open framework. This will enable a new generation of solutions to benefit the whole industry.

This is the first in the series of Architecture-Driven Modernization (ADM) standards that will allow users to begin modernization projects knowing that there is interoperability between different tool vendors. Standardization will ensure that end users are investing not just in individual tools but also rather into a coordinated strategy. See roadmap within the introduction section for details.

This RFP recognizes the existence of OMG specifications that are related to ADM. The response to this RFP should utilize these specifications or indicate why they are inadequate. Responders to this RFP are encouraged to reuse and possibly extend (if needed) existing OMG standards in order to make them more appropriate to the ADM domain.

This RFP solicits proposals for a meta-model for representing information related to existing software assets and their operational environments (referred to as the *Knowledge Discovery*). One common characteristic of various tools that address the ADM challenge is that they *analyze* the existing software assets (for example, source code modules, database descriptions, build scripts, etc.) to obtain existing systems knowledge. Each tool implements a portion of the knowledge about existing software assets. Such tool-specific knowledge may be implicit ("hard-coded" in the tool), restricted to a particular source language and/or particular transformation and/or operational environment. All the above may hinder interoperability between different tools. The meta-model for

Knowledge Discovery shall provide a common repository structure that will facilitate the exchange of data contained within individual tool models that represent existing software assets. The meta-model shall represent the physical and logical assets at various levels of abstraction. The primary purpose of this meta-model is to promote a common interchange format that will allow interoperability between existing modernization tools, services and their respective models.

For further details see Chapter 6 of this document.

1.0 Introduction

The roadmap for this project involves issuing multiple RFPs for ADM as follows:

A. Knowledge Discovery Meta-Model (KDM)

B. Target Meta-Models (for example, J2EE, Web services, or .NET)

C. Source-to-Target Mapping (STM) Meta-Model

D. Metrics Extension (ME)

The roadmap is subject to change. This RFP requests a specification for the KDM meta-model.

1.1 Goals of OMG

The Object Management Group (OMG) is the world's largest software consortium with an international membership of vendors, developers, and end users. Established in 1989, its mission is to help computer users solve enterprise integration problems by supplying open, vendor-neutral portability, interoperability and reusability specifications based on Model Driven Architecture (MDA). MDA defines an approach to IT system specification that separates the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform, and provides a set of guidelines for structuring specifications expressed as models. OMG has established numerous widely used standards such as OMG IDL[IDL], CORBA[CORBA], Realtime CORBA [CORBA], GIOP/IIOP[CORBA], UML[UML], MOF[MOF], XMI[XMI] and CWM[CWM] to name a few significant ones.

1.2 Organization of this document

The remainder of this document is organized as follows:

Chapter 2 - *Architectural Context* - background information on OMG's Model Driven Architecture.

Chapter 3 - *Adoption Process* - background information on the OMG specification adoption process.

Chapter 4 - *Instructions for Submitters* - explanation of how to make a submission to this RFP.

Chapter 5 - *General Requirements on Proposals* - requirements and evaluation criteria that apply to all proposals submitted to OMG.

Chapter 6 - *Specific Requirements on Proposals* - problem statement, scope of proposals sought, requirements and optional features, issues to be discussed, evaluation criteria, and timetable that apply specifically to this RFP.

Appendix A – References and Glossary Specific to this RFP

Appendix B – General References and Glossary

1.3 Conventions

The key words "**must**", "**must not**", "**required**", "**shall**", "**shall not**", "**should**", "**should not**", "**recommended**", "**may**", and "**optional**" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.4 Contact Information

Questions related to the OMG's technology adoption process may be directed to *omg-process@omg.org*. General questions about this RFP may be sent to *responses@omg.org*.

OMG documents (and information about the OMG in general) can be obtained from the OMG's web site *(http://www.omg.org/)*. OMG documents may also be obtained by contacting OMG at *documents@omg.org*. Templates for RFPs (this document) and other standard OMG documents can be found at the OMG *Template Downloads Page* at *http://www.omg.org/technology/template_download.htm*

2.0 Architectural Context

MDA provides a set of guidelines for structuring specifications expressed as models and the mappings between those models. The MDA initiative and the standards that support it allow the same model specifying business system or application functionality and behavior to be realized on multiple platforms. MDA enables different applications to be integrated by explicitly relating their models; this facilitates integration and interoperability and supports system evolution (deployment choices) as platform technologies change. The three primary goals of MDA are portability, interoperability and reusability.

Portability of any subsystem is relative to the subsystems on which it depends. The collection of subsystems that a given subsystem depends upon is often loosely called the *platform*, which supports that subsystem. Portability – and reusability - of such a subsystem is enabled if all the subsystems that it depends upon use standardized interfaces (APIs) and usage patterns.

MDA provides a pattern comprising a portable subsystem that is able to use any one of multiple specific implementations of a platform. This pattern is repeatedly usable in the specification of systems. The five important concepts related to this pattern are:

1. *Model* - A model is a representation of a part of the function, structure and/or behavior of an application or system. A *representation* is said to be *formal* when it is based on a language that has a well-defined form ("syntax"), meaning ("semantics"), and possibly rules of analysis, inference, or proof for its constructs. The syntax may be graphical or textual. The semantics might be defined, more or less formally, in terms of things observed in the world being described (e.g. message sends and replies, object states and state changes, etc.), or by translating higher-level language constructs into other constructs that have a well-defined meaning. The optional rules of inference define what unstated properties you can deduce from the explicit statements in the model. In MDA, a *representation* that is not *formal* in this sense is not a model. Thus, a diagram with boxes and lines and arrows that is not supported by a definition of the meaning of a box, and the meaning of a line and of an arrow is not a model—it is just an informal diagram.

2. *Platform* – A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented.

3. *Platform Independent Model (PIM)* – A model of a subsystem that contains no information specific to the platform, or the technology that is used to realize it.

4. *Platform Specific Model (PSM)* – A model of a subsystem that includes information about the specific technology that is used in the realization of that subsystem on a specific platform, and hence possibly contains elements that are specific to the platform.

5. *Mapping* – Specification of a mechanism for transforming the elements of a model conforming to a particular metamodel into elements of another model that conforms to another (possibly the same) metamodel. A mapping may be expressed as associations, constraints, rules, templates with parameters that must be assigned during the mapping, or other forms yet to be determined.

For example, in case of CORBA the platform is specified by a set of interfaces and usage patterns that constitute the CORBA Core Specification [CORBA]. The CORBA platform is independent of operating systems and programming languages. The OMG Trading Object Service specification [TOS] (consisting of interface specifications in OMG Interface Definition Language (OMG IDL)) can be considered to be a PIM from the viewpoint of CORBA, because it is independent of operating systems and programming languages. When the IDL to C++ Language Mapping specification is applied to the Trading Service PIM, the C++-specific result can be considered to be a PSM for the Trading Service, where the platform is the C++ language and the C++ ORB implementation. Thus the IDL to C++ Language Mapping specification [IDLC++] determines the mapping from the Trading Service PIM to the Trading Service PSM.

Note that the Trading Service model expressed in IDL is a PSM relative to the CORBA platform too. This highlights the fact that platform-independence and platform-specificity are relative concepts.

The UML Profile for EDOC specification [EDOC] is another example of the application of various aspects of MDA. It defines a set of modeling constructs that are independent of middleware platforms such as EJB [EJB], CCM [CCM], MQSeries [MQS], etc. A PIM based on the EDOC profile uses the middleware-independent constructs defined by the profile and thus is middleware-independent. In addition, the specification defines formal metamodels for some specific middleware platforms such as EJB, supplementing the already-existing OMG metamodel of CCM (CORBA Component Model). The specification also defines mappings from the EDOC profile to the middleware metamodels. For example, it defines a mapping from the EDOC profile to EJB. The mapping specifications facilitate the transformation of any EDOC-based PIM into a corresponding PSM for any of the specific platforms for which a mapping is specified.

Continuing with this example, one of the PSMs corresponding to the EDOC PIM could be for the CORBA platform. This PSM then potentially constitutes a PIM, corresponding to which there would be implementation language specific PSMs derived via the CORBA language mappings, thus illustrating recursive use of the Platform-PIM-PSM-Mapping pattern.

Note that the EDOC profile can also be considered to be a platform in its own right. Thus, a model expressed via the profile is a PSM relative to the EDOC platform.

An analogous set of concepts apply to Interoperability Protocols wherein there is a PIM of the payload data and a PIM of the interactions that cause the data to find its way from one place to another. These then are realized in specific ways for specific platforms in the corresponding PSMs.

Analogously, in case of databases there could be a PIM of the data (say using the Relational Data Model), and corresponding PSMs specifying how the data is actually represented on a storage medium based on some particular data storage paradigm etc., and a mapping from the PIM to each PSM.

OMG adopts standard specifications of models that exploit the MDA pattern to facilitate portability, interoperability and reusability, either through ab initio development of standards or by reference to existing standards. Some examples of OMG adopted specifications are:

1. *Languages* – e.g. IDL for interface specification, UML for model specification, OCL for constraint specification, etc.
2. *Mappings* – e.g. Mapping of OMG IDL to specific implementation languages (CORBA PIM to Implementation Language PSMs), UML

Profile for EDOC (PIM) to CCM (CORBA PSM) and EJB (Java PSM), CORBA (PSM) to COM (PSM) etc.

3. *Services* – e.g. Naming Service [NS], Transaction Service [OTS], Security Service [SEC], Trading Object Service [TOS] etc.
4. *Platforms* – e.g. CORBA [CORBA].
5. *Protocols* – e.g. GIOP/IIOP [CORBA] (both structure and exchange protocol), [XMI] (structure specification usable as payload on multiple exchange protocols).
6. *Domain Specific Standards* – e.g. Data Acquisition from Industrial Systems (Manufacturing) [DAIS], General Ledger Specification (Finance) [GLS], Air Traffic Control (Transportation) [ATC], Gene Expression (Life Science Research) [GE], Personal Identification Service (Healthcare) [PIDS], etc.

For an introduction to MDA, see [MDAa]. For a discourse on the details of MDA please refer to [MDAc]. To see an example of the application of MDA see [MDAb]. For general information on MDA, see [MDAd].

Object Management Architecture (OMA) is a distributed object computing platform architecture within MDA that is related to ISO's Reference Model of Open Distributed Processing RM-ODP[RM-ODP]. CORBA and any extensions to it are based on OMA. For information on OMA see [OMA].

3.0 Adoption Process

3.1 Introduction

OMG adopts specifications by explicit vote on a technology-by-technology basis. The specifications selected each satisfy the architectural vision of MDA. OMG bases its decisions on both business and technical considerations. Once a specification adoption is finalized by OMG, it is made available for use by both OMG members and non-members alike.

Request for Proposals (RFP) are issued by a *Technology Committee* (TC), typically upon the recommendation of a *Task Force* (TF) and duly endorsed by the *Architecture Board* (AB).

Submissions to RFPs are evaluated by the TF that initiated the RFP. Selected specifications are *recommended* to the parent TC after being *reviewed* for technical merit and consistency with MDA and other adopted specifications and *endorsed* by the AB. The parent TC of the initiating TF then votes to *recommend adoption* to the OMG Board of Directors (BoD). The BoD acts on the recommendation to complete the adoption process.

For more detailed information on the adoption process see the *Policies and Procedures of the OMG Technical Process* [P&P] and the *OMG Hitchhiker's Guide* [Guide]. In case of any inconsistency between this document and the [P&P] in all cases the [P&P] shall prevail.

3.2 Steps in the Adoption Process

A TF, its parent TC, the AB and the Board of Directors participate in a collaborative process, which typically takes the following form:

- *Development and Issuance of RFP*

  RFPs are drafted by one or more OMG members who are interested in the adoption of a standard in some specific area. The draft RFP is presented to an appropriate TF, based on its subject area, for approval and recommendation to issue. The TF and the AB provide guidance to the drafters of the RFP. When the TF and the AB are satisfied that the RFP is appropriate and ready for issuance, the TF recommends issuance to its parent TC, and the AB endorses the recommendation. The TC then acts on the recommendation and issues the RFP.

- *Letter of Intent (LOI)*

  A Letter of Intent (LOI) must be submitted to the OMG signed by an officer of the member organization, which intends to respond to the RFP, confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements. (See section 4.3 for more information.). In order to respond to an RFP the respondent must be a member of the TC that issued the RFP.

- *Voter Registration*

  Interested OMG members, other than Trial, Press and Analyst members may participate in specification selection votes in the TF for an RFP. They may need to register to do so, if so stated in the RFP. Registration ends on a specified date, 6 or more weeks after the announcement of the registration period. The registration closure date is typically around the time of initial submissions. Member organizations that have submitted an LOI are automatically registered to vote.

- *Initial Submissions*

  Initial Submissions are due by a specified deadline. Submitters normally present their proposals at the first meeting of the TF after the deadline. Initial Submissions are expected to be complete enough to provide insight on the technical directions and content of the proposals.

- *Revision Phase*

  During this time submitters have the opportunity to revise their Submissions, if they so choose.

- *Revised Submissions*

  Revised Submissions are due by a specified deadline. Submitters again normally present their proposals at the next meeting of the TF after the deadline. (Note that there may be more than one Revised Submission deadline. The decision to extend this deadline is made by the registered voters for that RFP.)

- *Selection Votes*

  When the registered voters for the RFP believe that they sufficiently understand the relative merits of the Revised Submissions, a selection vote is taken. The result of this selection vote is a recommendation for adoption to the TC. The AB reviews the proposal for MDA compliance and technical merit. An endorsement from the AB moves the voting process into the issuing Technology Committee. An eight-week voting period ensues in which the TC votes to recommend adoption to the OMG Board of Directors (BoD). The final vote, the vote to adopt, is taken by the BoD and is based on technical merit as well as business qualifications. The resulting draft standard is called the *Adopted Specification*.

- *Business Committee Questionnaire*

  The submitting members whose proposal is recommended for adoption need to submit their response to the BoD Business Committee Questionnaire [BCQ] detailing how they plan to make use of and/or make the resulting standard available in products. If no organization commits to make use of the standard, then the BoD will typically not act on the recommendation to adopt the standard. So it is very important to fulfill this requirement.

- *Finalization*

  A Finalization Task Force (FTF) is chartered by the TC that issued the RFP, to prepare an *adopted* submission for publishing as a formal, publicly available specification. Its responsibility includes production of one or more prototype implementations and fixing any problems that are discovered in the process. This ensures that the final available standard is actually implementable and has no show-stopping bugs. Upon completion of its activity the FTF recommends adoption of the resulting draft standard called the *Available Specification*. The FTF must also provide evidence of the existence of one or more prototype implementations. The parent TC acts on the recommendation and recommends adoption to the BoD. OMG Technical Editors produce the *Formal Published Specification* document based on this *Available Specification*.

- *Revision*

  A Revision Task Force (RTF) is normally chartered by a TC, after the FTF completes its work, to manage issues filed against the *Available Specification* by implementers and users. The output of the RTF is a revised specification reflecting minor technical changes.

3.3 Goals of the evaluation

The primary goals of the TF evaluation are to:

- Provide a fair and open process
- Facilitate critical review of the submissions by members of OMG
- Provide feedback to submitters enabling them to address concerns in their revised submissions
- Build consensus on acceptable solutions
- Enable voting members to make an informed selection decision

Submitters are expected to actively contribute to the evaluation process.

4.0 Instructions for Submitters

4.1 OMG Membership

To submit to an RFP issued by the Platform Technology Committee the submitter or submitters must be either Platform or Contributing members on the date of the submission deadline, while for Domain Technology RFPs the submitter or submitters must be either Contributing or Domain members. Submitters sometimes choose to name other organizations that support a submission in some way; however, this has no formal status within the OMG process, and for OMG's purposes confers neither duties nor privileges on the organizations thus named.

4.2 Submission Effort

An RFP submission may require significant effort in terms of document preparation, presentations to the issuing TF, and participation in the TF evaluation process. Several staff months of effort might be necessary. OMG is unable to reimburse submitters for any costs in conjunction with their submissions to this RFP.

4.3 Letter of Intent

A Letter of Intent (LOI) must be submitted to the OMG Business Committee signed by an officer of the submitting organization signifying its intent to respond to the RFP and confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements. These terms, conditions, and requirements are defined in the *Business Committee RFP Attachment* and are reproduced verbatim in section 4.4 below.

The LOI should designate a single contact point within the submitting organization for receipt of all subsequent information regarding this RFP and the submission. The name of this contact will be made available to all OMG

members. The LOI is typically due 60 days before the deadline for initial submissions. LOIs must be sent by fax or paper mail to the "RFP Submissions Desk" at the main OMG address shown on the first page of this RFP.

Here is a suggested template for the Letter of Intent:

This letter confirms the intent of <___organization required___> (the organization) to submit a response to the OMG <___RFP name required___> RFP. We will grant OMG and its members the right to copy our response for review purposes as specified in section 4.7 of the RFP. Should our response be adopted by OMG we will comply with the OMG Business Committee terms set out in section 4.4 of the RFP and in document omg/02-04-02.

<____contact name and details required____> will be responsible for liaison with OMG regarding this RFP response.

The signatory below is an officer of the organization and has the approval and authority to make this commitment on behalf of the organization.

<___signature required____>

4.4 Business Committee RFP Attachment

This section contains the text of the Business Committee RFP attachment concerning commercial availability requirements placed on submissions. This attachment is available separately as an OMG document omg/2002-04-02.

Commercial considerations in OMG technology adoption

A1 Introduction

OMG wishes to encourage rapid commercial adoption of the technologies (specifications and support measures) it publishes. To this end, there must be neither technical, legal nor commercial obstacles to their implementation. Freedom from the first is largely judged through technical review by the relevant OMG Technology Committees; the second two are the responsibility of the OMG Business Committee. The BC also looks for evidence of a commitment by a submitter to the commercial success of products based on the submission.

A2 Business Committee evaluation criteria

A2.1 Viable to implement across platforms

While it is understood that final candidate OMG submissions often combine technologies before they have all been implemented in one system, the Business Committee nevertheless wishes to see evidence that each major feature has been implemented, preferably more than once, and by separate organizations. Pre-product implementations are acceptable. Since use of OMG specifications should not be dependent on any one platform, cross-platform availability and interoperability of implementations should be also be demonstrated.

A2.2 Commercial availability

In addition to demonstrating the existence of implementations of the specification, the submitter must also show that products based on the specification are commercially available, or will be within 12 months of the date when the specification was recommended for adoption by the appropriate Task Force. Proof of intent to ship product within 12 months might include:

- *A public product announcement with a shipping date within the time limit.*
- *A prototype implementation and accompanying draft user documentation.*

Alternatively, and at the Business Committee's discretion, submissions may be adopted where the submitter is not a commercial software provider, and therefore will not make implementations commercially available. However, in this case the BC will require concrete evidence of two or more independent implementations of the specification being used by end-user organizations as part of their businesses.
Regardless of which requirement is in use, the submitter must inform the OMG of completion of the implementations when commercially available.

In the case of the proposed adoption of support measures, the BC needs to have proof of the intent to use or recommend such support measures within 12 months of the date when the support measures were recommended for adoption by the appropriate Task Force.

A2.3 Access to Intellectual Property Rights

OMG will not adopt a specification or support measure if OMG is aware of any submitter, member or third party which holds a patent, copyright or other intellectual property right (collectively referred to in this policy statement as "IPR") which might be infringed by implementation or recommendation of such specification or support measure, unless OMG believes that such IPR owner will grant a license to organizations (whether OMG members or not) on non-discriminatory and commercially reasonable terms which wish to make use of the specification or support measure. Accordingly, the

submitter must certify that it is not aware of any claim that the specification or support measure infringes any IPR of a third party or that it is aware and believes that an appropriate non-discriminatory license is available from that third party. Except for this certification, the submitter will not be required to make any other warranty, and specifications will be offered by OMG for use "as is". If the submitter owns IPR to which an use of a specification or support measure based upon its submission would necessarily be subject, it must certify to the Business Committee that it will make a suitable license available to any user on non-discriminatory and commercially reasonable terms, to permit development and commercialization of an implementation that includes such IPR.

It is the goal of the OMG to make all of its technology available with as few impediments and disincentives to adoption as possible, and therefore OMG strongly encourages the submission of technology as to which royalty-free licenses will be available. However, in all events, the submitter shall also certify that any necessary license will be made available on commercially reasonable, non-discriminatory terms. The submitter is responsible for disclosing in detail all known restrictions, placed either by the submitter or, if known, others, on technology necessary for any use of the specification or support measure.

A2.4 Publication of the specification

Should the submission or support measures be adopted, the submitter must grant OMG (and its sublicensees) a worldwide, royalty-free license to edit, store, duplicate and distribute both the specification and works derived from it (such as revisions and teaching materials). This requirement applies only to the written specification, not to any implementation of it.

A2.5 Continuing support

The submitter must show a commitment to continue supporting the technology underlying the specification or support measure after OMG adoption, for instance by showing the BC development plans for future revisions, enhancement or maintenance.

4.5 Responding to RFP items

4.5.1 Complete proposals

A submission must propose full specifications for all of the relevant requirements detailed in Chapter 6 of this RFP. Submissions that do not present complete proposals may be at a disadvantage.

Submitters are highly encouraged to propose solutions to any optional requirements enumerated in Chapter 6.

4.5.2 Additional specifications

Submissions may include additional specifications for items not covered by the RFP that they believe to be necessary and integral to their proposal. Information on these additional items should be clearly distinguished.

Submitters must give a detailed rationale as to why these specifications should also be considered for adoption. However submitters should note that a TF is unlikely to consider additional items that are already on the roadmap of an OMG TF, since this would pre-empt the normal adoption process.

4.5.3 Alternative approaches

Submitters may provide alternative RFP item definitions, categorizations, and groupings so long as the rationale for doing so is clearly stated. Equally, submitters may provide alternative models for how items are provided if there are compelling technological reasons for a different approach.

4.6 Confidential and Proprietary Information

The OMG specification adoption process is an open process. Responses to this RFP become public documents of the OMG and are available to members and non-members alike for perusal. No confidential or proprietary information of any kind will be accepted in a submission to this RFP.

4.7 Copyright Waiver

If a submitted document is copyrighted, a waiver of copyright for unlimited duplication by the OMG is required to be stated in the document. In addition, a limited waiver of copyright is required that allows each OMG member to make up to fifty (50) copies of the document for review purposes only.

4.8 Proof of Concept

Submissions must include a "proof of concept" statement, explaining how the submitted specifications have been demonstrated to be technically viable. The technical viability has to do with the state of development and maturity of the technology on which a submission is based. This is not the same as commercial availability. Proof of concept statements can contain any information deemed relevant by the submitter; for example:

"This specification has completed the design phase and is in the process of being prototyped."

"An implementation of this specification has been in beta-test for 4 months."

"A named product (with a specified customer base) is a realization of this specification."

It is incumbent upon submitters to demonstrate to the satisfaction of the TF managing the evaluation process, the technical viability of their proposal. OMG will favor proposals based on technology for which sufficient relevant experience has been gained.

4.9 Format of RFP Submissions

This section presents the structure of a RFP submission. The *final revised submission* must be in this form before a vote can be taken to recommend its adoption..

4.9.1 General

- Submissions that are concise and easy to read will inevitably receive more consideration.
- Submitted documentation should be confined to that directly relevant to the items requested in the RFP. If this is not practical, submitters must make clear what portion of the documentation pertains directly to the RFP and what portion does not.
- The key words "**must**", "**must not**", "**required**", "**shall**", "**shall not**", "**should**", "**should not**", "**recommended**", "**may**", and "**optional**" shall be used in the submissions with the meanings as described in RFC 2119 [RFC2119].

4.9.2 Required Outline

A three-part structure for submissions is required. Parts I is non-normative, providing information relevant to the evaluation of the proposed specification. Part II is normative, representing the proposed specification. Specific sections like Appendices may be explicitly identified as non-normative in Part II. Part III is normative specifying changes that must be made to previously adopted specifications in order to be able to implement the specification proposed in Part II....

PART I

- Copyright Waiver (see 4.7)
- Submission contact point (see 4.3)
- Overview or guide to the material in the submission
- Overall design rationale (if appropriate)
- Statement of proof of concept (see 4.8)
- Resolution of RFP requirements and requests

Explain how the proposal satisfies the specific requirements and (if applicable) requests stated in Chapter 6. References to supporting material in Part II should be given.

In addition, if the proposal does not satisfy any of the general requirements stated in Chapter 5, provide a detailed rationale.

- Responses to RFP issues to be discussed

  Discuss each of the "Issues To Be Discussed" identified in Chapter 6.

PART II

The contents of this part should be structured based on the template found in [FORMS] and should contain the following elements as per the instructions in the template document cited above:

- Scope of the proposed specification
- Proposed Conformance criteria

  Submissions should propose appropriate conformance criteria for implementations.

- Proposed Normative References

  Submissions should provide a list of the normative references that are used by the proposed specification

- Proposed List of Terms and Definitions

  Submissions should provide a list of terms that are used in the proposed specification with their definitions.

- Proposed List of Symbols

  Submissions should provide a list of special symbols that are used in the proposed specification together with their significance

- Proposed specification.

PART III

- Changes or extensions required to adopted OMG specifications

  Submissions must include a full specification of any changes or extensions required to existing OMG specifications. This should be in a form that enables "mechanical" section-by-section revision of the existing specification.

4.10 How to Submit

Submitters should send an electronic version of their submission to the *RFP Submissions Desk* (*omg-documents@omg.org*) at OMG Headquarters by 5:00

PM U.S. Eastern Standard Time (22:00 GMT) on the day of the Initial and Revised Submission deadlines. Acceptable formats are Postscript, ASCII, PDF, Adobe FrameMaker, Microsoft Word, and WordPerfect. However, it should be noted that a successful (adopted) submission must be supplied to OMG's technical editors in FrameMaker source format, using the most recent available OMG submission template (see [FORMS]). The AB will not endorse adoption of any submission for which appropriately formatted FrameMaker sources are not available; it may therefore be convenient to prepare all stages of a submission using this template.

Submitters should make sure they receive electronic or voice confirmation of the successful receipt of their submission. Submitters should be prepared to send a single hardcopy version of their submission, if requested by OMG staff, to the attention of the "RFP Submissions Desk" at the main OMG address shown on the first page of this RFP.

5.0 General Requirements on Proposals

5.1 Requirements

5.1.1 Submitters are encouraged to express models via OMG modeling languages such as UML, MOF, CWM and SPEM (subject to any further constraints on the types of the models and modeling technologies specified in Chapter 6 of this RFP). Submissions containing models expressed via OMG modeling languages shall be accompanied by an OMG XMI [XMI] representation of the models (including a machine-readable copy). A best effort should be made to provide an OMG XMI representation even in those cases where models are expressed via non-OMG modeling languages.

5.1.2 Chapter 6 of this RFP specifies whether PIM(s), PSM(s), or both are being solicited. If proposals specify a PIM and corresponding PSM(s), then the rules specifying the mapping(s) between the PIM and PSM(s) shall either be identified by reference to a standard mapping or specified in the proposal. In order to allow possible inconsistencies in a proposal to be resolved later, proposals shall identify whether the mapping technique or the resulting PSM(s) are to be considered normative.

5.1.3 Proposals shall be *precise* and *functionally complete*. All relevant assumptions and context required for implementing the specification shall be provided.

5.1.4 Proposals shall specify *conformance criteria* that clearly state what features all implementations must support and which features (if any) may *optionally* be supported.

5.1.5 Proposals shall *reuse* existing OMG and other standard specifications in preference to defining new models to specify similar functionality.

5.1.6 Proposals shall justify and fully specify any *changes or extensions* required to existing OMG specifications. In general, OMG favors proposals that are *upwards compatible* with existing standards and that minimize changes and extensions to existing specifications.

5.1.7 Proposals shall factor out functionality that could be used in different contexts and specify their models, interfaces, etc. separately. Such *minimalism* fosters re-use and avoids functional duplication.

5.1.8 Proposals shall use or depend on other specifications only where it is actually necessary. While re-use of existing specifications to avoid duplication will be encouraged, proposals should avoid gratuitous use.

5.1.9 Proposals shall be *compatible* with and *usable* with existing specifications from OMG and other standards bodies, as appropriate. Separate specifications offering distinct functionality should be usable together where it makes sense to do so.

5.1.10 Proposals shall preserve maximum *implementation flexibility*. Implementation descriptions should not be included and proposals shall not constrain implementations any more than is necessary to promote interoperability.

5.1.11 Proposals shall allow *independent implementations* that are *substitutable* and *interoperable*. An implementation should be replaceable by an alternative implementation without requiring changes to any client.

5.1.12 Proposals shall be compatible with the architecture for system distribution defined in ISO's Reference Model of Open Distributed Processing [RM-ODP]. Where such compatibility is not achieved, or is not appropriate, the response to the RFP must include reasons why compatibility is not appropriate and an outline of any plans to achieve such compatibility in the future.

5.1.13 In order to demonstrate that the specification proposed in response to this RFP can be made secure in environments requiring security, answers to the following questions shall be provided:

- What, if any, are the security sensitive elements that are introduced by the proposal?
- Which accesses to security-sensitive elements must be subject to security policy control?
- Does the proposed service or facility need to be security aware?
- What default policies (e.g., for authentication, audit, authorization, message protection etc.) should be applied to the security sensitive elements introduced by the proposal? Of what security considerations mustthe implementers of your proposal be aware?

5.1.14 The OMG has adopted several specifications, which cover different aspects of security and provide useful resources in formulating responses. [CSIV2] [SEC] [RAD].

5.1.15 Proposals shall specify the degree of internationalization support that they provide. The degrees of support are as follows:

a) Uncategorized: Internationalization has not been considered.

b) Specific to <region name>: The proposal supports the customs of the specified region only, and is not guaranteed to support the customs of any other region. Any fault or error caused by requesting the services outside of a context in which the customs of the specified region are being consistently followed is the responsibility of the requester.

c) Specific to <multiple region names>: The proposal supports the customs of the specified regions only, and is not guaranteed to support the customs of any other regions. Any fault or error caused by requesting the services outside of a context in which the customs of at least one of the specified regions are being consistently followed is the responsibility of the requester.

d) Explicitly not specific to <region(s) name>: The proposal does not support the customs of the specified region(s). Any fault or error caused by requesting the services in a context in which the customs of the specified region(s) are being followed is the responsibility of the requester.

5.2 Evaluation criteria

Although the OMG adopts model-based specifications and not implementations of those specifications, the technical viability of implementations will be taken into account during the evaluation process. The following criteria will be used:

5.2.1 Performance

Potential implementation trade-offs for performance will be considered.

5.2.2 Portability

The ease of implementation on a variety of systems and software platforms will be considered.

5.2.3 Securability

The answer to questions in section 5.1.14 shall be taken into consideration to ascertain that an implementation of the proposal is securable in an environment requiring security.

5.2.4 Conformance: Inspectability and Testability

The adequacy of proposed specifications for the purposes of conformance inspection and testing will be considered. Specifications should provide

sufficient constraints on interfaces and implementation characteristics to ensure that conformance can be unambiguously assessed through both manual inspection and automated testing.

5.2.5 Standardized Metadata

Where proposals incorporate metadata specifications, usage of OMG standard XMI metadata [XMI] representations must be provided as this allows specifications to be easily interchanged between XMI compliant tools and applications. Since use of XML (including XMI and XML/Value [XML/Value]) is evolving rapidly, the use of industry specific XML vocabularies (which may not be XMI compliant) is acceptable where justified

6.0 Specific Requirements on Proposals

The following section outlines the problem statement, scope of proposals sought, relationship to existing OMG specifications, related activities, documents and standards, mandatory and optional requirements, issues to be discussed, evaluation criteria and timetables.

6.1 Problem Statement

There is a vast amount of useful, deployed, operational software representing an enormous commercial value. A significant amount of software resists evolution because its strategic value and/or its 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, its ability to interoperate with other systems or external sources, and its dependence on undesired technologies or architectures. There is a need to understand and evolve existing software assets for the purpose of documentation, improvement, modification, interoperability, porting, migrations, reuse, redesign and/or redeployment.

In addition, there exists a significant gap between tools and methodology support that is available for “green-field” projects and for the projects, which involve maintenance and evolution of existing code bases.

6.1.1 Business Value of ADM

Existing systems, which have evolved to embody conflicting or challenging designs and whose modification requires laborious manual coding, 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. Transformation can help meet these business requirements more quickly at a reduced cost and with better quality.

For a large number of organizations the need for transformation of their enterprise information systems is usually triggered by certain external events such as the need to provide Web-access to an existing application or the need to abandon an unsupported platform. For companies that deliver technology-dependent or software-intensive products, the evolution of existing code is a vital part of day-to-day business requirements.

ADM provides the means to address these requirements using systematic and low risk approaches.

6.1.2 Interoperability Benefits

A large industry of software tool vendors and service vendors exists to enable ADM. There is a need for standardization to enable integration and interoperability among solutions from multiple ADM vendors. Standardization will increase interoperability between different tools by creating an open framework. This can enable a new generation of solutions to benefit the whole industry.

ADM standards will allow users to begin modernization projects knowing that there is interoperability between different tool vendors. Standardization will ensure that end users are investing not just in individual tools but rather into a coordinated strategy.

6.1.3 Long-term Benefits of New Technology Adoption

There are also long-term strategic reasons to work towards standardization of ADM. In the past the forward engineering industry has largely ignored existing software assets, while reverse engineering was in turn ignoring the new forward engineering methods. Over time, this has created a significant gap between tools and methodology support that is available for "green-field" projects and for the projects, which involve maintenance and evolution of existing code bases.

There is a significant gap in the ability of organizations to apply model driven solutions to existing systems due to the nature of those systems. In existing systems, meta-data is not well documented or has been documented in a manner that does not conform to adopted industry standards. Few if any people understand the inherent design that underlies existing systems. In addition, existing systems historically do not lend themselves to being able to extract model driven views from those systems.

Current efforts will help break the barriers between forward engineering tools and reverse engineering tools and thus will 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). ADM also facilitates the evolution of existing systems through the use of forward engineering disciplines and allows the re-deployment of those systems using newer technologies including J2EE, .NET, Web services or CORBA.

6.1.4 Bottom-line Benefits of Standardization

Standardization of ADM meta-models will help industry and individual businesses by reducing the risk of undertaking software improvement initiatives. The ability to share common information across projects that use a variety of tools and processes will lessen the time, risk and cost of software transformations. This will in turn improve the quality of reverse engineering and ADM tools, provide new capabilities, and extend the ROI on software development tools.

6.2 Scope of Proposals Sought

This RFP solicits proposals for a Knowledge Discovery Meta-model (KDM) for exchanging information related to transformation of existing software assets. Specifically, the proposal seeks a common repository structure to represent information about existing software assets and its operating environment.

The KDM represents the structure of the existing software and its related artifacts. It provides the ability to document existing systems, discover reusable components in existing software, support transformations to other languages and to MDA, or enable other potential transformations.

The meta-model will also enable information about existing software artifacts to be exchanged among different tools. This will enable vendors that specialize on certain languages, platforms or types of transformations to deliver customer solutions in conjunction with other vendors.

There are at least four aspects to the knowledge gathered through the ADM process.

- Implementation
- Design
- Architecture
- Business rules

The KDM shall represent the principal artifacts of existing software, which in general terms can be described as the entities (the structural elements, the "things"), their relations, and attributes. The meta-model shall be general enough to address all the above levels of transformations.

The meta-model shall not be restricted to any particular implementation language or platform. This is especially important for addressing the aspects of existing knowledge relevant to the language level. In particular, the meta-model shall be able to represent behavioral programming artifacts down to *but not below a procedure level*. A procedure is, for example, a paragraph within a COBOL program or a method in Java. The reason for this constraint on the scope of the RFP is that entities and their relationship below the procedure level are often too platform- and language-specific.

Entities at the implementation aspect that we do want to include, for example, are methods, source files, classes, screen definitions, data elements, records, tables or transactions. Relationships at the language level include, for example, "function uses a variable", "class inherits from another class", or "file includes a header". Attributes at the language level include, for example, name, access rights, version, source language, last scan date, or type of relationship.

Design aspects of a system vary widely based on methodological approach. In general, design information can include data design, system design, interface

design and program design. Discoverable aspects include, for example, redundancy, cohesion, usability, accessibility, performance and fragmentation. The KDM should facilitate this discovery process.

The architecture aspect is important for ADM. At the architectural level there are several kinds of structures:

- Physical structures (for example, files, directories, relations between files, build dependencies, etc.)
- Logical structures (for example, subsystems, modules, layers, components and their dependencies, various architecture views, etc.)

Physical structures that are related to the build process of the existing software are often critical in ADM projects. Architectural entities include, for example, modules, subsystems, architecture layers, components or libraries. Architectural relationships include, for example, "file is contained in directory", "component provides a method", "component uses API of another component", etc. Architectural attributes include, for example, component name.

In order to be applicable to the ADM on an architectural basis, the KDM shall be able to represent both physical as well as logical structures. The meta-model shall relate the elements of logical structures to physical structures as well as to language-level structures.

As far as the business rules aspect is concerned, knowledge about the functions being performed within a system should be representable within the KDM. A business rule entity provides a functional description of a portion of a system that may be augmented by systems analysts. A relationship from a given business rule may be linked back to one or more physical system aspects. At the business rules aspect, the KDM shall not be specific to a particular domain.

To accommodate these aspects of ADM, the meta-model shall be created in such a way that facilitates its population through either automated or manual means or facilities.

6.2.1 RFP Assumptions

Responders should take note of the following assumptions with respect to this RFP.

- "Platform" as used in this RFP does not refer exclusively to middleware but refers to hardware and/or software environments that span mainframe, mid-range, network, Web-based as well as other environments. "Platform-independent", therefore, means that a given

representation is independent of any specific hardware and/or software environment.

- All terms not defined herein, this RFP defers to OMG documents listed in sections 6.3 and 6.4.

The RFP assumes that submitters will use MOF. One of the mandatory requirements is of the proposed meta-model to be *at least* EMOF compliant. This should not restrict submitters from using CMOF.

6.3 Relationship to Existing OMG Specifications

This RFP relates to several OMG specifications. These include:

- MOF (ptc/03-10-14): including
  - Facilities to define meta-models
  - Exchange formats (XMI, etc.)
  - QVT (transformation between two models based on a definition in terms of elements from the meta-models)
- UML (ptc/03-08-02; ptc/03-09-15; ptc/03-10-14)
- SPEM (formal/2002-11-14)
- CWM (formal/2003-03-02)
- EDOC (includes a meta-model for Java) (ptc/02-02-05)
- EAI (includes detailed meta-models for programming languages calls, including COBOL) (ptc/02-02-02)

Submitters should reuse the relevant parts of these specifications by referencing them, should use terminology consistently with these specifications and should otherwise mention why these specifications or parts thereof are not relevant for the purpose of this RFP.

6.4 Related Activities, Documents and Standards

Related documents, URLs and standards relevant to this RFP include:

- Systems Engineering RFP (in particular the requirement to describe an installation of an existing system for maintenance purposes) RFP ad/2003-03-41
- Business Semantics of Business Rules RFP br/03-06-03
- Reusable Assets Specification (RAS) RFC ad/2003-10-12
- Software Portfolio Management RFP (bom/2000-12-15)

There is no known overlap with specification activities or specifications, competing or complementary, from other standards bodies.

6.5 Mandatory Requirements

This section specifies the mandatory requirements that must be satisfied by any proposal submitted in response to this RFP.

6.5.1 Meta-Model Requirements

The meta-model shall enable the definition of the artifacts of and related to the existing software assets. A proposal for the meta-model shall include the following:

1. The meta-model shall be MOF 2.0 compliant at the EMOF level.
2. UML shall be used to represent meta-model diagrams.
3. The meta-model shall define a single unified terminology for the domain of knowledge discovery about existing software assets.
4. Where appropriate, that terminology shall match the existing UML terminology.
5. The meta-model shall represent the principal artifacts of existing software as entities, their relationships and attributes. For example, entities include data structures, methods, source files, classes, screen definitions, or transactions. For example, relationships include "function uses a variable", "class inherits from another class", or "file includes a header". For example, attributes include name, access rights, version, source language, last scan date, or type of relationship.
6. The meta-model shall represent external environmental artifacts. In other words the meta-model shall not be restricted only to the artifacts of the existing software itself, but shall also represent the things with which the software interacts. For example, environmental artifacts include hardware actuator, sensor, user, input stimulus, output device, operating system, virtual machine, or manual process.
7. The meta-model shall consist of a platform and language independent core.
8. The meta-model shall support a wide range of the major platforms and languages such as COBOL, C, Java, Ada, z OS, Unix, J2EE or .NET.
9. The meta-model shall be able to represent multiple h eterogeneous systems, that is, systems composed of software artifacts based on different platforms and/or languages.
10. The submission shall include a description of how the meta-model can be extended.

11. The ability to describe the physical structure of the existing software, including the build dependencies between physical artifacts, referencing the appropriate portions of existing specifications related to software configuration management.

12. The ability to describe the logical structure that can be used to aggregate and/or modify the physical structure of the software artifacts, such as for refactoring.

13. The ability to trace any entity, relationship, or attribute from a logical structure back to the physical artifacts.

14. The meta-model shall be able to represent behavioral programming artifacts down to but not below a procedure level. For example, the paragraph level within a COBOL program, or a method in Java.

15. The meta-model shall be delivered in a MOF XMI form as well as in the document format.

6.6 Optional Requirements

1. The meta-model may reuse, where possible, and extend, if necessary, the UML Infrastructure Core package, EAI, CWM, EDOC or other applicable meta-models.

2. The meta-model may allow for language-specific or platform-specific extensions in order to capture specialized concepts that are not common.

3. The meta-model may use a single unified terminology that relates to the synonyms used in other existing sources that are known to the responder of the proposal.

6.7 Issues to be discussed

These issues will be considered during submission evaluation. They should not be part of the proposed normative specification.

6.7.1 The proposal shall include information on how the single unified terminology relates to the synonyms used in other existing sources that are known to the responder of the proposal.

6.7.2 The proposal shall provide a list of the languages and platforms that the meta-model is claimed to support.

6.7.3 The proposal shall discuss the level of compliancy with MOF chosen by submitters (for example, EMOF compliant, CMOF compliant).

6.7.4 The proposal shall discuss the relationship between the proposed KDM and the Reusable Asset Specification (RAS).

6.8 Evaluation Criteria

A responder to this RFP has to provide proof of the following:

6.8.1 Demonstrate the capability to map the existing software artifacts in a common implementation language, such as C, or C++ or Java or Ada or COBOL into a repository that can be described by the proposed meta-model.

6.8.2 Demonstrate the ability of the proposed meta-model to support more than one implementation language.

6.8.3 Demonstrate the usability of the proposed meta-model for the purpose of:

- Viewing information about the existing system
- Generating cross-referencing reports (inventory of entities and places in the existing software assets, where these entities are used)
- Manipulating, such as adding, modifying and extending, in particular annotations and relations

6.9 Other information unique to this RFP

6.10 RFP Timetable

The timetable for this RFP is given below. Note that the TF or its parent TC may, in certain circumstances, extend deadlines while the RFP is running, or may elect to have more than one Revised Submission step. The latest timetable can always be found at the OMG *Work In Progress* page at http://www.omg.org/schedules/ under the item identified by the name of this RFP. Note that "<month>" and "<approximate month>" is the name of the month spelled out; e.g., January.

Duration	Event or Activity	Actual Date
	Preparation of RFP by TF	

	RFP placed on OMG document server	*27th October 2003*
	Approval of RFP by Architecture Board *Review by TC*	
0	*TC votes to issue RFP*	*November 2003*
60	*LOI to submit to RFP due*	*15th February 2004*
113	*Initial Submissions due and placed on OMG document server ("Three week rule")*	*31st May 2004*
134	*Voter registration closes*	*15th June 2004*
141	*Initial Submission presentations*	*21st June 2004*
	Preliminary evaluation by TF	
240	*Revised Submissions due and placed on OMG document server ("Three week rule")*	*October 2004*
261	*Revised Submission presentations*	*October 2004*
	Final evaluation and selection by TF *Recommendation to AB and TC*	*January 2005*
	Approval by Architecture Board *Review by TC*	*January 2005*
330	*TC votes to* recommend specification	*March 2005*
360	*BoD votes to adopt specification*	*May 2005*

Appendix A References and Glossary Specific to this RFP

A.1 References Specific to this RFP

A.1.1

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

A.1.2

Comsys Systems Redevelopment Methodology:

www.comsysprojects.com/SystemTransformation/TMethodology.htm

A.2 Glossary Specific to this RFP

The following definitions were taken from various sources as noted. Certain definitions were derived, in part, from the Comsys Systems Redevelopment Methodology – see A.1.2.

Architecture: The way an application system is assembled and functions. This includes functional (how information is processed), data (how data is stored) and technical (how the system is deployed) architectures.

Business Unit: A functional area of an organization.

Business Rule: A combination of conditional and imperative logic that changes the state of an object or data element - depending on the methodology being used. (Source: OMG)

Complexity: Measurement of the difficulty required to maintain or enhance a program or system.

Count: The number of occurrences of a source or target architecture asset or an attribute of that asset.

Cyclomatic Complexity: A term describing the number of basic paths through a program.

Data Definition: The programmatic implementation of data. Think of a data definition as the container that holds the data. The data is the liquid in the cup.

Data Definition Rationalization: Reducing data group definitions to a subset of common names and formats.

Defect: An unplanned and potentially problematic attribute of a system.

Essential Complexity: The count of unstructured constructs in a control flow graph of a program.

Function: A group of business activities, which collectively support one aspect of furthering the mission of the enterprise (source Information Engineering).

I/O Record Group: A record group (see record group) that contains at least one record definition used in an I/O (input / output) transaction.

Knowledge Discovery Meta-Model (KDM): An entity relationship model representing existing system artifacts and the relationships among those components.

Maintenance: Process of applying changes to existing systems that include corrective, adaptive and perfective activities. Large scale architectural change is typically not considered as maintenance.

- **Corrective** maintenance identifies and corrects implementation flaws or design errors.

- **Adaptive** or functional maintenance deals with changes in the data requirements or processing environment.
- **Perfective** maintenance supports performance improvement, cost reduction efforts or required technical improvements.

McCabe Metrics: Metrics described by Thomas McCabe. Basic metric types include cyclomatic complexity which measures the testability of a program and essential complexity which measures the structure of a program. (Also see Cyclomatic Complexity and Essential Complexity and Appendix B - Metric Guide.)

Metrics: A count or derived score quantifying the attributes of a source or target application environment.

Modernization: The process of understanding and evolving existing software assets.

Physical Structure: The physical portions of an executable production system.

Record Group: A group of related data elements (such as a COBOL 01 level, database segment or table) grouped based on the fact that they define related data.

Recursion: A program construct that invokes itself.

Repository: An information storage facility or database that contains all meta-data relevant to a given set of requirements. A transformation repository defines the transformation meta-model.

Reuse: The ability to leverage an occurrence of a design model or software asset in more than one system or implementation.

Structure: To convert unstructured source code to structured source code.

Reverse Engineering: Process of analyzing a system to identify components and interrelationships, and to create representations in another form or at a higher level of abstraction. (Source: IEEE)

Scenario: A project oriented template that defines a set of transformation related steps based on a common project goal.

Score: A resulting number based on a metric calculation.

Static Analysis: Method of analyzing a program by tracking program logic paths without executing the program.

Step: A defined set of activities that provide a succinct deliverable or set of deliverables within a transformation process.

Task: Collection of related steps within a transformation process.