



Reviewing MDA Work Products

Version 3.0
May 25, 2004

PathMATE™ Series

Pathfinder Solutions LLC
33 Commercial Street, Suite 2
Foxboro, MA 02035 USA
www.PathfinderMDA.com
888-543-7222

Table of Contents

PathMATE Overviewiii

1. Introduction 1

2. Goals 1

3. Reviews in the Overall MDA Process 2

4. Individual Review Points..... 5

 System-Level Requirements Document5

 Domain Model.....6

 Domain Requirements Matrix7

 Information Model8

 Scenario Models 10

 State Model 11

 Action Model..... 12

5. Summary..... 14

6. References 14

PathMATE Overview

This overview introduces Model Driven Architecture (MDA) and the PathMATE™ tools that make MDA work. MDA and PathMATE move you from writing and debugging code to developing and testing the logic of a high performance system. Over years of rigorous refinement in several industries, PathMATE tools have proven their value in rapid and effective software systems development.

PathMATE Toolset

The PathMATE Model Automation and Transformation Environment includes all the tools required to transform your MDA models into high-performance systems (Figure 1).

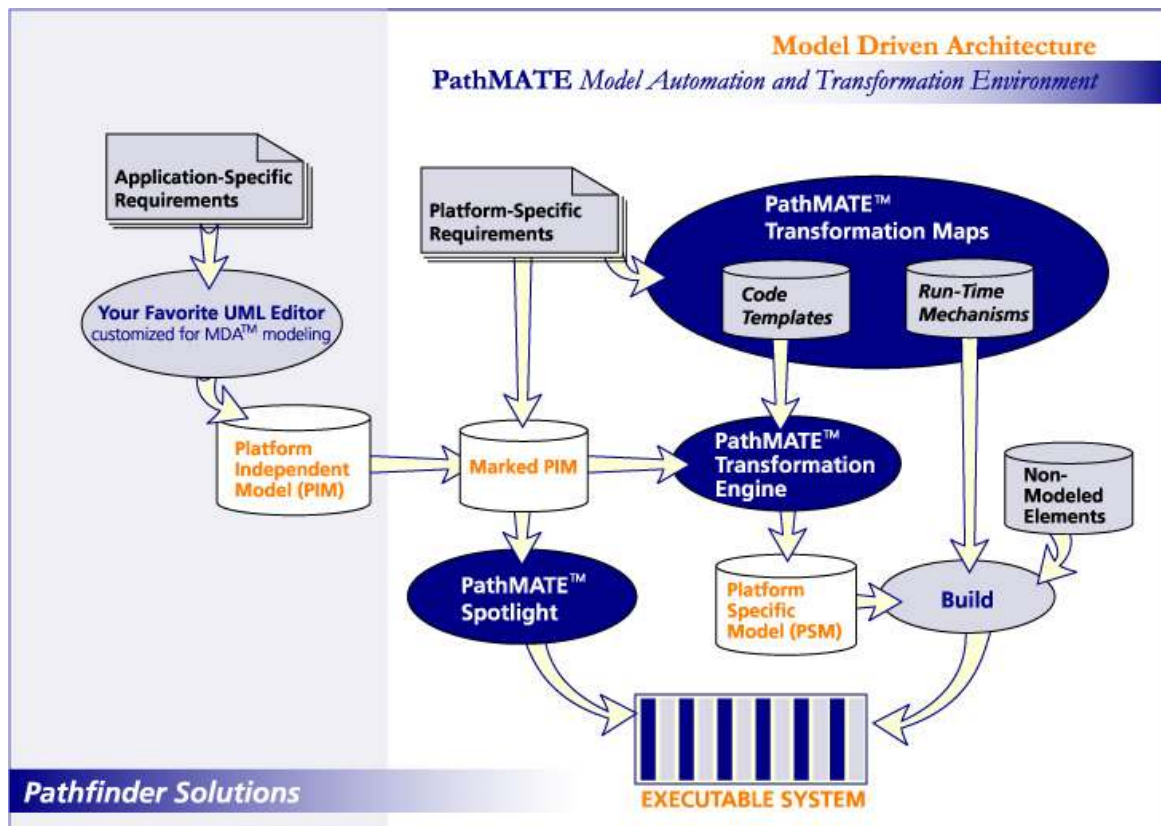


Figure 1. PathMATE Toolset

The three parts of the PathMATE toolset cooperate to turn your models into executable systems:

- *Transformation Maps* – Generate C, C++, or Java software with off-the-shelf Transformation Maps, or create custom maps to drive output for other languages or specific platforms.
- *Transformation Engine* – The Engine transforms platform-independent models into working, embedded software applications.
- *Spotlight* – Verify and debug your application logic with Spotlight, the most advanced model testing environment available.

No other MDA transformation environment offers a more open or configurable set of development tools, designed to meet the requirements of systems engineers.

How PathMATE Works

Use Model Driven Architecture to build complex embedded systems that meet rigorous standards for speed and reliability. MDA works because it separates what the system does from its deployment on a particular platform. PathMATE adds these advantages:

- *Greatest architectural control* – A highly configurable Transformation Engine enables you to optimize output for resource-constrained platforms.
- *Clean separation of model and code* – Conforming to the MDA paradigm, PathMATE models contain no implementation code. That gives you fast and flexible deployment and migration capabilities.
- *Configurable, target-based model execution and testing* – Preemptively eliminate platform-specific bugs, minimize quality assurance resources, and accelerate development.
- *Lowest cost of ownership* – Integrate PathMATE with your existing UML editor. Build on your previous investment in training and software.
- *Speed* – Even large transformations take just seconds with PathMATE. That enables highly iterative model development, and rapid transformation and test cycles.

Try the demonstration software available at www.PathfinderMDA.com to get started quickly and easily.

1. Introduction

Reviewing MDA Work Products is a companion to the paper, *Model Based Software Engineering: Rigorous Software Development with Domain Modeling*. It covers the nature, scope, and frequency of MDA work product reviews. The discussion centers primarily on how and when to conduct work product reviews. We also discuss how the review process fits into the overall MDA process, and differences between internal and external reviews.

2. Goals

The goals of this MDA work product review guide are to:

- Establish the overall review process
- Provide suggestions on how and when to conduct working reviews, internal reviews, and formal (external) reviews
- Understand the goals and documentation needs of each review point
- Provide review techniques for each work product

3. Reviews in the Overall MDA Process

Review points vary with the scope of each MDA phase, and each phase has different goals and techniques for work reviews or sessions, internal reviews, and external reviews.

Theory of Operation

The successful establishment of review culture can make this form of meeting a very productive form of team interaction. This requires an investment in preparation, discipline and respect. Some important general guidelines for successful reviews are:

- Make material available early enough: five days for external reviews, varies for internal (by volume) but at least one full day before the actual review meeting.
- Publish and follow an agenda.
- Use a facilitator to keep the review on track.
- For anything more formal than a working review, assign a recorder to keep track of important discussion points and all action items.
- Maintain respect - speak concisely on a topic and within the agenda, listen attentively and don't interrupt.

Working Review

A working review is simply a gathering of the team members responsible for a single work product, such as the Detailed System-Level Requirements, or the Class Diagram for a domain. These creative working sessions have few guidelines, but do keep the overhead low: don't use formal action items, and don't invite non-technical people. Also, avoid doing individual tasks in a group setting, such as reading the review material for the first time, or fixing typos.

Goal

The goal for all working reviews is the same: complete the overall task at hand as quickly and effectively as possible. This can be helped by ensuring each individual clearly understands their overall responsibilities and exactly which assignments are due at the next working review session.

Frequency

The frequency of the working reviews should be tuned to the needs of the team: more frequent in the first part of an effort to ensure coordination, less frequent as people tackle the bulk of their assignments, then more frequent again as individual contributions are completed and integrated.

It is important to provide adequate time between working review sessions to allow individuals to accomplish their assignments, and to be sure momentum is not lost by spacing working reviews too far apart.

It is strongly recommended that at least 2 people be assigned to any individual set of work products. This helps increase overall system quality and team productivity. In the case that only one person can be assigned to an activity, then this person must find some other person or team to serve as a sounding board. The individual must strike a balance between the frequency of dedicated team Working Reviews and the Internal Review.

Internal Review

In contrast to the more informal working review, the reviewers for internal reviews are typically members of the team not directly responsible for the work product under review. Depending on the work product, this may include representatives from the other subsystems in the domain as well as team members working on clients of or servers to this domain. Internal reviews should be viewed as a working event conducted in a cooperative setting. While somewhat informal it is still important to have a review agenda and to record comments and action items as they arise. Internal review meetings should be limited to no more than 2 hours. Sticking to the agenda and covering only non-trivial comments is imperative.

Frequency

Typically, an internal review will be conducted once for a given set of work products. However, if a significant number or scope of issues are identified during the first Internal Review, another follow-up may be required. Internal reviews should be scheduled by the work product producers and interpreted as a statement that they believe they are ready to move on to the next stage in the MDA process.

External Review

The reviewers for an external review are from outside the project team. These reviewers could be customer representatives or outside consultants. In this setting it is even more important to have and to stick to a clear agenda. Due to the typically more survey-level review characteristic of an external review, all individual contributors probably need not attend. To get the most out of the external review it is crucial that the review results are summarized, that action items are clearly identified and assigned (with expected completion dates), and that meeting minutes are published including the action items and conclusions reached at the review.

Frequency

Generally, a single External Review will be conducted once for a given set of high-level work products. Follow-up reviews should only be

conducted where significant rework is required due to major issues or requirement changes.

4. Individual Review Points

This section discusses review of selected individual work products from the System-Level Requirements document through Process Modeling.

System-Level Requirements Document

Timely completion and approval of the System Requirements Document can be the first, decisive step in a successful project. Alternatively, inability to focus on this work product can provide the first in a long series of sliding and ambiguous efforts. Proper review structure can provide the crisp starting step needed to kick things off properly.

Internal Review Goals

- determine that document is complete and clear enough to begin development
- provide a forum for team to integrate individual efforts
- identify and record issues and questions for external resolution

External Review Goals

- achieve sign-off for detailed description of system functionality OR identify specific issues requiring resolution
- freeze requirements - provide foundation for development and setting against which future requirements change requests are evaluated.
- provide a public forum to demonstrate early progress
- for change reviews: provide forum in which to publish and JUSTIFY new schedule.

Working Review Frequency

The scope of this effort varies widely based on system scope, subject matter complexity, etc. Typically the review frequency will increase over effort duration, as the research and writing assignments complete.

Documentation Needs

- the document itself with an agenda is all that is needed
- for external reviews, an approval sign-off sheet may help focus the participation of external entities.
- for post-freeze external change reviews, a new schedule also accompanies

Techniques

A thorough understanding of an effective detailed requirement can eliminate some subjectivity from the review process:

- Overall: verify that the document adequately addresses product specification features/requirements
- Individual requirements: make sure that requirements are understandable, have an external (system) perspective, are implementation-free, and testable

Domain Model

The domain model is the first set of MDA analysis work products, and in many ways the most crucial analysis element. A well conceived and understood domain model will pay large dividends throughout subsequent analysis steps.

Internal Review Goals

- determine that domain breakout is complete and clear enough to begin further analysis
- identify and record issues and questions for external resolution

External Review Goals

- achieve sign-off for domain chart OR identify specific issues requiring resolution
- provide visibility to external entities on domain chart content and layout
- provide a public forum to demonstrate early progress

Working Review Frequency

The domain model is usually a collaborative effort of a small number of key technical contributors - where working "reviews" are group creative sessions. For the initial domain modeling effort, working sessions should be conducted daily unless some significant research, interviewing, etc. must be conducted. While the preservation of momentum is important for all activities, it is critical for the domain model. The domain modeling effort should be short and intense - not a drawn-out background effort.

Documentation Needs

- domain chart with domain and bridge descriptions
- system requirements document

Techniques

Determining what a "good" domain model is relatively more difficult than for some of the lower level analysis models where the techniques tend to be better understood. However, there are several characteristics of a well structured domain model including:

- clear subject matter division between domains
- domain breakout not simply by platform or program
- domain complexity level is not excessive - look for opportunities for delegation to new server domains
- a top level application domain that represents the main system concepts
- for maximum reuse possibilities and subject matter purity, server domain naming should not be from perspective of clients
- common abstractions do not show up in multiple domains
- bridges represent flow of requirements only - NOT data flow or flow of control
- high cohesion within domains
- low coupling between domains

Domain Requirements Matrix

The domain requirements matrix is a simple partitioning of the system level requirements to the domains that fulfill them and can be a tremendous help in clarifying the domain's system role and responsibilities as well as avoiding duplication of effort in different domains. It is also critical in providing a forum in which the analyst can record and resolve issues and assumptions.

Internal Review Goals

- determine that system requirements allocation to domains is complete and clear enough to continue further analysis
- identify and record issues and questions for external resolution

External Review Goals

An external domain requirements matrix review is probably unnecessary.

Working Review Frequency

The domain requirements document is typically an individual effort by the domain owner. For large domains, help may be needed to partition a large volume of system requirements. Working Reviews should be tailored to meet the needs of each effort.

Documentation Needs

- domain requirements matrix document
- system requirements document
- domain model with descriptions
- a list of this domain's published services

Techniques

Document assessment includes:

- complete coverage - all appropriate requirements have been mapped from system requirements document and/or appropriate product specifications
- requirements mapped to a domain are appropriate for domain
- system requirements that do not map to a single domain have been clearly split and allocated to the appropriate domains
- all services expected by clients are covered

Information Model

Internal Review Goals

- verify all abstractions are appropriate for domain
- determine that the Class Diagram adequately supports all needs identified by the requirements matrix
- verify that good UML Analysis principles and syntax have been followed
- identify and record issues and questions that require external resolution

External Review Goals

- provide a forum for detailed information dissemination, feedback and possibly validation from technically capable external audiences
- conduct detailed review of product requirements and detailed system requirements in a structured conceptual context
- provide another public forum to demonstrate early progress

Working Review Frequency

An individual analyst should schedule a group review of his/her work on a bi-weekly basis (assuming a dedicated focus on modeling).

For large domain teams (maybe 4 or more analysts), individual analysts may be paired to provide frequent (every days or so) review

of each other's work without involving the entire team. In this context, the entire team should conduct Working Reviews work every week.

Documentation Needs

- domain mission and bridge descriptions
- Class Diagram, including object, attribute, and relationship descriptions
- domain requirements matrix
- "client references" report
- (for external): product description and detailed system-level requirements

Techniques

Reviewing a Class Diagram should be done from two different perspectives. The *MDA perspective* looks primarily at modeling techniques and MDA syntax while the *subject matter perspective* is more interested in how well the Class Diagram captures the domain requirements. It is important to avoid focusing completely on one perspective and ignoring the other.

Analysis Perspective

At a minimum, the following Class Diagram characteristics should be examined:

- Abstraction relevance - the object, attribute and, relationship abstractions are appropriate to this domain
- Object and attribute descriptions -- describe the model abstractions rather than real world entities that they may share a name with
- Attribute atomicity - attributes represent atomic data elements within this domain
- Relationship naming and descriptions -- clearly describe the relationship's meaning and the "why" behind the multiplicity and conditionality
- Model conciseness - is there over-abstraction (too many objects with not much to do) i.e. could the model complexity be reduced?

Subject Matter Perspective

While the subject matter perspective requires an understanding of how to read a Class Diagram, the reviewer is less focused on the model constructs in of themselves. Instead, the reviewer verifies that the domain requirements are adequately addressed by the Class Diagram's objects and the relationships between them.

Scenario Models

Scenario Models are the highest level analysis of the dynamic within a domain, and perform two important functions. First, they focus attention back on the current domain's Class Diagram. Difficulty in determining which objects should be involved in system scenarios and/or excessive event traffic between objects is probably an indication that the Class Diagram needs some rework. Second, the behavioral details identified on the Scenario Models provide a kick-start for the developing the state models and services.

Internal Review Goals

- ensure the subset of scenarios initially chosen represent the majority of core processing for the domain
- verify the object behavior will adequately and appropriately support the domain requirements
- identify and record issues and questions for external resolution

External Review Goals

An external review of the Scenario Models is generally inappropriate.

Working Review Frequency

The Scenario Modeling effort is a very creative collaboration of the analysts to develop the behavioral strategy in a domain. Like domain modeling, a short but intensive effort should be structured with frequent working sessions.

Documentation Needs

- Scenario Models
- Class Diagram
- Domain Requirements Matrix
- scenario description documents

Techniques

Scenario Models have two main review characteristics that need to be considered. First, do the scenarios exercise most or all of the important uses of the system? "Important" scenarios could mean frequently performed, performance-critical, or possibly safety-critical depending on the system. Second, given that the scenarios are well chosen, are the event traffic patterns created to satisfy these scenarios adequate and appropriate?

- scenario selection - should be checked against the domain requirements document to see that all important scenarios have been explored.

- activity patterns - some patterns to watch out for include excessive event traffic to an object (may indicate that object is too complex and should delegate some of its responsibilities), more objects than necessary involved in a relatively simple scenario (may indicate over-abstraction of objects), and excessive use of external domain bridge services which could indicate poor subject matter separation.

State Model

The state model review provides a detailed context from which to exercise and verify the high-level behavior of a domain.

Internal Review Goals

- verify object behavior follows the patterns laid out in the Scenario Models
- ensure actions obey the run-time rules of MDA: all attributes are consistent at the end of each action
- determine the robustness of each state model against unexpected events
- verify proper event labeling and state naming
- review the suitability of state action descriptions, ensuring event generation and bridge service essentials are captured, and an appropriate high-level perspective is maintained
- ensure all event, and event data item descriptions are complete and consistent

External Review Goals

An external review of the state models is generally inappropriate.

Working Review Frequency

With a solid plan in the form of the Scenario Models, state models are typically created individually. Working reviews for each state model should be held once the core processing is modeled, and again at the end once the State Transition Table and error analysis are completed.

Documentation Needs

- Class Diagram
- Scenario Models and Scenario Descriptions
- State Models, State Transition Tables, and a report with event and event parameter descriptions

Techniques

State models have several important characteristics that should be checked by the reviewer:

- abstraction relevance - Just as creating state models usually affects the first cut Class Diagram, the state model reviewer should reflect on the Class Diagram abstractions given the state models. Overly complex state models may indicate that an object or set of objects may need to be repartitioned. Question the validity of overly simple or bureaucratic objects - with actions that do no more than mimic the transitions of other objects and maintain relationships.
- event meanings - accurate description of the request or incident that causes the transition. The event meaning should identify a single point in time.
- state naming - accurate description of the object's condition during the state action, representing some finite span of time. Do not name a state with an event meaning.
- state action text - high-level, concise description of the state action in English-like prose. Only provide enough detail to determine any events that are generated or services invoked during the action.

Action Model

The Action Model is the final step in the analysis process and has the least subjectivity of any of the MDA work products. In addition to any automated syntax checking that is commonly available, we recommend that there be at least selective Action Model review. Similar to peer code reviews conducted on elaborative projects, Action Model review can often save time in pointing out errors prior to translation, compilation and testing.

Internal Review Goals

- uncover incorrectly interpreted state model or service actions,
- verify compliance with Action Modeling rules and conventions,
- verify proper use of design-level features and other server domain services.

External Review Goals

An external review of the Action Models is generally inappropriate.

Working Review Frequency

Action Modeling is an individual activity. A team may elect to review work completed on a per-object basis, or may only review select Action Models, based on a variety of criteria such as apparent difficulty, complexity, participation in a core scenario, use of an

external domain service, etc. While it is important to have at least an analysis partner review all work, the team doesn't have to review each Action Model in a group setting.

Documentation Needs

- State models, service definitions
- Action Models
- Service descriptions of all services invoked within the scope of the review material

Techniques

- syntax - automated static checking
- detailed "penny" simulation - employ desk-checking to execute the models manually verifying that state model action has been accurately interpreted by the analyst
- external domain service verification - spot checking that invoked services have been used as specified by server domain documentation.

5. Summary

One goal of this document is to provide a detailed set of MDA modeling review items and suggest a set of techniques to apply to them. Another goal - perhaps a more important one - is to establish a pattern of review and a general set of guidelines for efficiently and effectively producing creative work products in a team environment.

As with all such recommendations, consider the apparent intent of a suggestion as a higher law than any details of the suggestion itself, and tailor all techniques to the specific requirements of your project, organization and culture.

6. References

For more information about PathMATE, please call Pathfinder Solutions at 508-384-1392, e-mail us at info@pathfindermda.com, or visit us at www.pathfindermda.com. You may wish to refer to the following sources:

On PathMATE:

Model Based Software Engineering: Rigorous Software Development with Domain Modeling, Pathfinder Solutions, 2004 (this paper is available from www.pathfindermda.com)

On the UML™:

The Unified Modeling Language User Guide, Grady Booch, James Rumbaugh, Ivar Jacobson, Addison Wesley, 1999; ISBN 0-201-57168-4

UML Distilled, Martin Fowler, Addison Wesley, 1997; ISBN 0-201-32563-2

UML Summary Version 1.1, Object Management Group, Inc. 1997 (this paper is available from www.omg.org)

UML™ is a trademark of Object Management Group, Inc in the U.S. and other countries.

Acknowledgments

We would like to thank the people at the Teradyne/ATB Tester Software Group. who participated in the development of this document.