Model-driven architecture
White paper
June 2009

Rational. software

# Model-driven architecture, embedded developers and IBM Rational Rhapsody.

*Rick Boldt*

**Contents**



## Executive summary

Model-driven architecture (MDA) is a software development approach that uses models to perform tasks such as specification writing and application development. The approach enables developers to separate system functionality and behavior from implementation details.

With MDA, developers first create one or more platform independent models (PIM) which are later translated into one or more platform-specific models (PSM). This approach enables applications to be more easily ported from one environment to another.

The MDA approach is evolving to include a broad range of design concepts that may be applied to numerous kinds of software projects spanning many industries, including electronic commerce, financial services, healthcare, aerospace and transportation. Embedded developers must tailor their solutions to help maximize the benefits of MDA while meeting the special needs of their environment such as: realtime performance, compact code, safety, reliability and specialized hardware control.

IBM® Rational® Rhapsody® software's key enabling technologies focus on the needs of embedded developers. Rhapsody software enables embedded developers to easily separate functionality and behavior from implementation detail in order to get more out of the MDA process.

**Highlights**

***MDA enables developers to create platform-independent models (PIM) that later may be easily translated to various platform-specific models (PSM).***

## MDA overview

In the development world today, there is a proliferation of component and distribution infrastructure environments, an ongoing evolution to new source-level programming languages and different modeling standards.

Operating in such a diverse and rapidly-changing environment creates a multitude of problems: How can you design a system that can successfully integrate disparate technologies? How can you build a system that will be robust and stable today and in the future when new technologies comes into use?

MDA exists to answer these questions through the application of modeling technology. The MDA is an approach aimed at developing applications that integrate today and in the future. In MDA, you can develop a platform-independent Unified Modeling Language (UML) model of your application. This PIM is then mapped to one or a set of appropriate infrastructure and implementation environments, such as Common Object Requesting Broker Architecture (CORBA), Component Object Model (COM) or the realtime operating system (RTOS) to create a PSM.

MDA is based on the sound principle that separating the specification of the systems operation from the details of how it will use its platform enables the portability, interoperability and reusability of the initial application. This architectural separation is achieved using the following approach:

- *Create a system specification (PIM) without knowledge of the intended execution platform.*
- *Define the platform to be used.*
- *Transform the system specification into one for the selected platform (PSM).*

By first creating a model that focuses on the functionality and the behavior of the application and then later translating that model into one that contains the details of the target, realtime operating system (RTOS), middleware and communication mechanisms, the original model may be more easily targeted for different environments. This is the promise of MDA.

**Highlights**

***IBM Rational Rhapsody helps companies create robust, integrated applications that can adapt to changing technology and infrastructure requirements.***

### Rhapsody software's key technologies for embedded developers

IBM Rational Rhapsody is a UML 2.1 model-driven development (MDD) solution built around a series of enabling technologies that can provide you with highly effective means of producing systems and software-intensive designs. Rhapsody software focuses on the needs of the embedded developer and supports the concepts of MDA. The robust support of UML 2.0 within IBM Rational Rhapsody is particularly suited to bridging the functional and object-oriented gap in one environment, thereby facilitating an extremely flexible design approach. For effective MDD, Rhapsody software creates a design environment that helps keep you in constant communication with the system's behavior through execution and validation based on the graphical design.

### Visual modeling with UML 2.0

The model-driven environment in IBM Rational Rhapsody is based upon standard UML 2.0 implementation combined with structural modeling such as block diagramming, creating a comprehensive systems and software environment.

***With its support of UML 2.1, Rhapsody software offers developers an effective tool for creating, testing, debugging and managing a wide variety of complex models.***

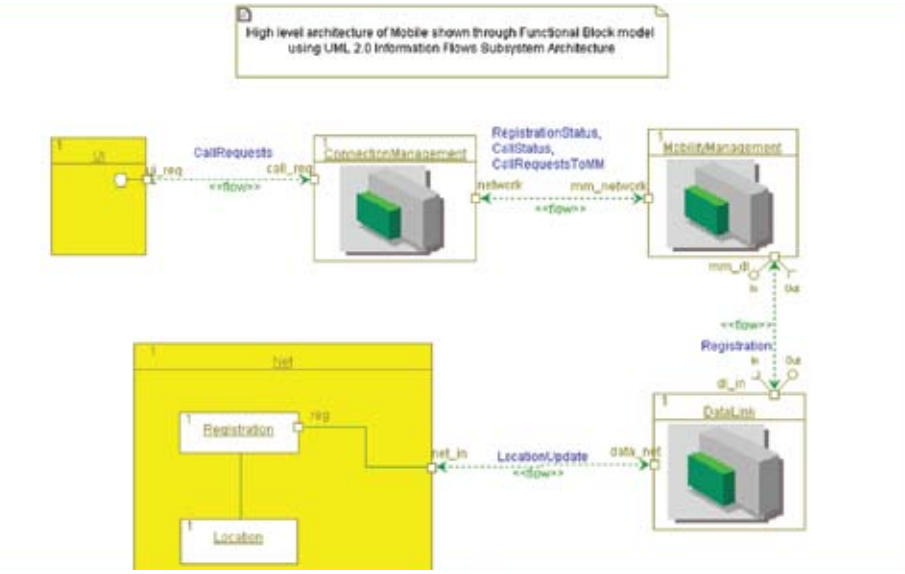


*Figure 1: UML 2.0 architectural views*

**Highlights**

***The key development technologies in Rhapsody software offer visual, executable and dynamic model creation capabilities as well as tools to enable change management automation, targeting, testing and team collaboration.***

*Executable models*

Execution of the graphical environment helps give you a visualization and understanding of how the system behaves at run time, even on the target.

*Dynamic model/code associativity*

Dynamic model/code associativity (DMCA) can allow you to freely work within the source files, and helps ensure that changes made at the source level are dynamically updated in the model, so the model and code can stay in sync.

*Automatic application synthesis*

Create a build environment for virtually any RTOS right inside the model, including the generated code, legacy code, third-party libraries, other model components and so on. A key component of this functionality is the realtime framework that allows a Rhapsody model to be automatically targeted to the platform that the realtime framework has been ported to.

*Design-level debug and test*

Exercise the model in a stepwise fashion and view the behavior to see if it conforms to the specifications, as well as automatically test and validate your system with the requirements.

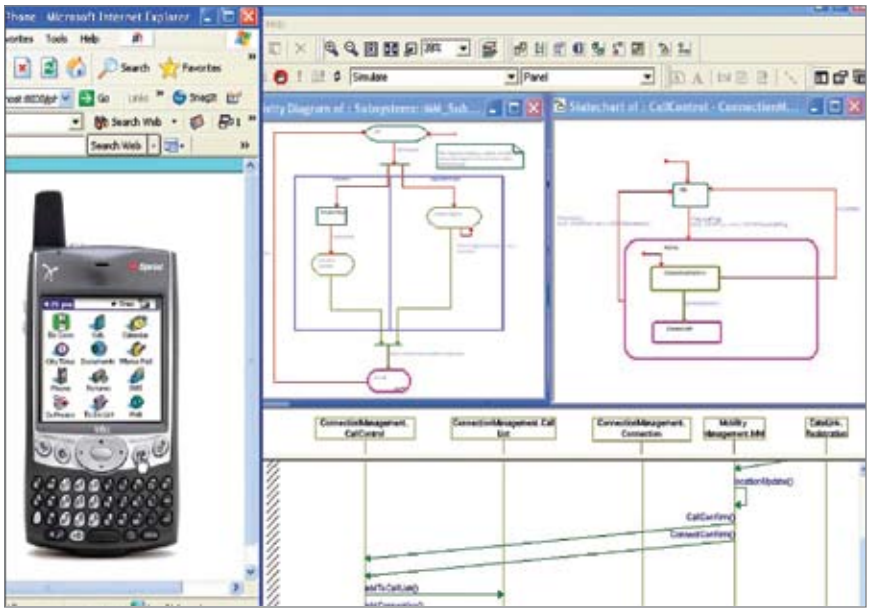

*Figure 2: Design-level debugging through realistic panel*

**Highlights**

***Rhapsody software works with UML 2.1 to create platform-independent models with extended functionality such as the ability to create enhanced sequence diagrams, information flows and other components critical to embedded design.***

*Scalable team collaboration*

Connect with nearly any configuration management infrastructure to help small and large teams work seamlessly together on projects, whether the team is local or distributed throughout the world. Take advantage of advanced model-level differencing and merging capability.

### The MDA and IBM Rational Rhapsody

The usual starting point within an MDA-based process is to create a UML model of your application that is platform independent. This model is referred to as a PIM and is meant to allow the designer to focus on the functionality and behavior of the application without having to worry about the implementation details such as what RTOS, hardware and middleware will be used.

Rhapsody software uses mechanisms supported by the MDA Guide Version 1.0.1 available from the Object Management Group (OMG) to help achieve platform independence with its realtime framework. You can create a PIM using UML and then simply select the desired implementation of the realtime framework for your target and transform the PIM into a PSM.

Many solutions are currently available that will enable you to capture a UML model. However, not all solutions are created equally. To help ensure you can achieve the full benefits of MDA, make sure you know how each potential solution measures up against the following key elements: amount of UML support, XML Metadata Interchange (XMI) support, textual language choice and the ability to debug and test the behavior and functionality of the PIM.

UML has become a de facto standard for modeling software applications and is widely considered to be the modeling language to use in an MDA process. A major focus of OMG's newly revised and enhanced UML version 2.1 is the ability to create a systems model. A UML 2.0 systems model maps well to an MDA PIM model. UML 2.1 provides several enhancements that both aid in creating models that are platform independent and are especially important when creating embedded designs. These include enhanced sequence diagrams, ports, tags, enhanced activity diagrams, information flows, statechart inheritance and structured classes.

Ports can be used to define the essential interfaces in the system. Providing well-defined interfaces is key to enabling the reuse of system elements, both within the current project and throughout the company for other projects. Sequence diagram enhancements enable lifeline decomposition and the referencing of scenarios within a sequence diagram. This capability allows you to capture more complex scenarios within your system or application by making them more readable. Using flows, you can explicitly show what pieces of information move between elements. This information enables you to create a comprehensive picture of the interfaces between elements.

Tags, and the ability to create profiles, allow you to assign values such as weight, reliability, size, material, security level and cost to system elements. This enables you to describe the elements of a system beyond the architectural, structural, behavioral and functional definitions that the graphics provide using items such as: activity diagrams, which support unrestricted concurrency; statechart inheritance, which enables easy reuse of the behavioral parts of the model; and composite classes with parts, providing for easy hierarchal decomposition. IBM Rational Rhapsody supports all of these important UML 2.1 concepts that aid in creating a robust, comprehensive and efficient PIM.

**Highlights**

***Solutions that enable developers to use standard programming languages to define PIM model triggers, methods and algorithms offer greater flexibility and portability than those that require proprietary language use.***

*Textual language considerations*

The graphics of UML are not enough to completely capture the functionality and behavior of the PIM model. A textual language is also needed to define, at a minimum, the triggers and actions that will occur. The textual language may also be used to define methods and algorithms within the model. One important requirement is that the textual language must contain semantics that are formal enough to translate into a PSM. UML 2.1 has defined semantics for this language, but it does not specify the syntax of this language.

Solution vendors offer two different choices when it comes to this textual language. Some solutions have their own proprietary language, while others allow developers to use a standard programming language. The potential benefits of using a standard language include: a shorter learning curve, greater portability from one user to another, increased language flexibility, and the abundance of third-party solutions that may be used to augment the modeling solution's capabilities.

The benefits of using a proprietary language are somewhat ambiguous. Some vendors will claim that a proprietary language enables the simulation of the model. While it may be true that solutions with proprietary languages enable simulation, the same results can be achieved using a standard programming language. Another argument is that the model is not truly platform independent unless implementations can be generated in different languages. This is a false argument for two reasons. First, it doesn't matter if you use a standard text language or a proprietary text language; you can generate an implementation in a different language using rules-based code generation technology. Second, even the MDA guide does not necessarily consider the programming language to be platform specific.

**Highlights**

IBM Rational Rhapsody enables you to use either the C++, C, Ada or Java™ language as the textual language. In each case, you can simulate the model as a PIM to help ensure that the functionality and behavior are correct at the model or specification level. You also have the option to use rules-based code generation to generate an implementation with a different programming language, in the rare case that this is desired.

***Don't be locked into a solution that requires you use a specific tool or vendor. Rhapsody software supports PIM model development using C++, C, Ada or Java as the textual language.***

The key principle in creating a PIM is that it be independent of any platform, including any solutions that you may use to create the model. In fact, what good is a PIM if it traps you into using a specific tool or vendor? XMI is the XML-based standard used to exchange model information between UML solutions. Rhapsody software supports the XMI standard, which enables you to easily export Rhapsody software models and import them from other solutions, facilitating comprehensive PIM development.

*Testing and debugging capabilities*

Creating a model is important, but it's even more important in a model-driven development process to ensure the model correctly depicts the intended functionality and behavior of the application. In order to do this, you must thoroughly test and debug the model.

***Rational Rhapsody supports a newer testing and debugging methodology that tests the actual code generated instead of relying on model interpretation. This method provides better insight into application functionality and behavior.***

Tool vendors often present two choices when it comes to debugging the PIM. The older method is based on interpreting the model, whereas the newer method allows you to debug the model using the actual code generated. Though the older method allows you to debug the entire model, including the graphics, there is no guarantee that the actual generated code will act in the same way as that in the interpreted model.

The newer method enables you to debug the actual code in conjunction with the model. This method is more powerful, especially when doing embedded development, because it allows you to test the algorithms as they will be deployed.

**Highlights**

Another factor that can increase the accuracy of debugging is the ability to include the same services and scheduling that the RTOS, middleware and hardware will provide in the simulation.

To help achieve platform independence, Rhapsody follows the common technique to target a system model for a technology-neutral virtual machine. It uses this approach with its realtime framework and model execution technology, helping enable the PIM behavior and functionality to be debugged on the host platform.

*Defining the target environment*

The PIM is the starting point of the MDA process, but the translation process and the ability to define the target environment or platform are also critical to the process' success. When doing an embedded design, these tasks are especially important.

***With the flexibility to easily target and retarget PIMs to PSMs, Rhapsody software helps companies stay agile in today's complex ever-changing development environment.***

With IBM Rational Rhapsody, you can define the target environment using the realtime framework. This capability is extremely powerful because it enables you to easily retarget the PIM to a different PSM. Additionally, whether you have a commercial RTOS or your own scheduling environment, Rhapsody software helps test and debug the application at the design or PIM level while still taking into account the standard scheduling concepts used in typical embedded designs.

## Conclusion

The MDA initiative responds to the burgeoning complexity of today's systems and system environments. It helps answer questions about how to best protect and reuse intellectual property in the face of shifting infrastructure and evolving language technology. Using standardized infrastructures to implement PIMs created in UML enables you to migrate your systems to new technology as it becomes available. It also enables you to integrate systems constructed with widely divergent technology, including today's complex, component-based distributed systems. IBM Rational Rhapsody offers robust PIM application generation and testing capabilities to help meet your needs today and into the future.

**For more information**

To learn more about IBM Rational Rhapsody, contact your IBM representative or IBM Business Partner, or visit:

**ibm.com**/software/rational



IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
June 2009


RAW14072-USEN-01