Driving business agility with Model Driven Architecture

An emerging approach for cost-effective, reliable and rapid application development

position paper: accelerated development



Model-Driven Architecture (MDA) is currently one of the most exciting approaches for accelerating code development and improving the quality of software in complex systems. MDA combines computer-aided verification and machine intelligence during modeling to discover and remove design bugs before code reviews and testing. The upshot: Companies can save significant production time and costs — and gain the business agility they demand.

EDS

Table of Contents

Author:

Steven Witkop
Information Specialist
Ohio/Pennsylvania Solution Centre
EDS

Introduction

Getting mission-critical enterprise applications up and running fast — and ensuring they do the job they're designed for over the long-term — is key for an enterprise's success. But doing so is often easier said than done. And that's where an evolutionary approach to rapid application development known as Model Driven Architecture (MDA) can pay big dividends. Since MDA keeps business domain code loosely coupled with platform-specific code, it gives enterprises the flexibility and agility to evolve business requirements independently from technology. The approach also helps companies maximize their investments in existing technology.

Let's examine the approach in more detail. After addressing its history from the industry perspective, we'll showcase its usefulness through a practical example. We'll conclude by listing five ways MDA can help enterprises move with agility.

The advent of Model Driven Architecture

Model Driven Architecture (MDA) is about using modeling languages as programming languages. Modeling languages enable us to program systems at a higher level of "abstraction" than is possible by using languages such as Java and XML.

Through the abstraction process, developers hide all but the relevant data in the system's code — thereby reducing its complexity and increasing development productivity. Abstraction also increases the system's longevity because the system's specifications are less tied to the underlying computing environment, which is always in flux.

MDA is part of a broad effort across the computer industry to raise the level of abstraction in how we develop systems. This is not the first time the industry has raised the level of abstraction to improve productivity; it did exactly this during the transition from assembly language to third-generation languages (3GLs).

The change to 3GLs didn't happen overnight, though. The industry experienced some teething problems along the way. At first, 3GL compilers (the programs that turn hand-written source code into machine code that a computer can process and use) didn't produce the machine code very effectively. It was easier and faster for developers to write the code themselves. Over time, however, as computers increased processing speed and compiler technology improved, developers started using compilers and became far more productive. Today, programming business applications in machine code is virtually unheard of because it would be less productive by an order of magnitude.

The role of Object Management Group

Those who design computer systems — whether for banks or battleships — face several choices. When they're interested in protecting investments in existing technology and maximizing flexibility, they choose hardware and software that implements and uses open standards. Open standards enable companies to run or interoperate multiple programs that were originally designed to work only on specific platforms. Given today's rapidly changing, multi-vendor computing environment — evidenced by the adoption of software platforms such as J2EE and .NET — the use of open standards makes sense.

How can organizations ensure their mission-critical information systems are rooted in standards that will adapt to new hardware capabilities and software platforms? The Object Management Group (OMG) addresses this reality with MDA. MDA addresses the challenges of today's highly networked, constantly changing systems environment by providing an architecture that ensures the following:

- Portability — by increasing application reuse and reducing the cost and complexity of application development and management, now and into the future
- Cross-platform interoperability — by using rigorous methods to guarantee that standards based on multiple implementation technologies all implement identical business functions
- Platform independence — by greatly reducing the time, cost and complexity associated with retargeting applications for different platforms (including those yet to be introduced)
- Domain specificity — through domain-specific models that enable rapid implementation of new, industry-specific applications over diverse platforms
- Productivity — by allowing developers, designers and system administrators to use languages and concepts they're comfortable with, while allowing seamless communication and integration across the teams

MDA represents an evolutionary step forward from previous development approaches.

MDA represents an evolutionary step forward from previous development approaches. It's built on the solid foundation of well-established OMG standards, including Unified Modeling Language (UML), the ubiquitous modeling notation used and supported by every major company in the software industry, and XML Metadata Interchange (XMI), the standard for storing and exchanging models using XML.

For more information about Model Driven Architecture, visit the Object Management Group's Web site: http://www.omg.org/mda

Not just another accelerated approach

Most new development approaches today address ways to accelerate application delivery. Yet not all of these approaches are alike. The model-driven approach focuses on developing domain models and using machine intelligence to generate code; other approaches concentrate on complex modeling and rely on platform-specific skills and dexterity. More importantly, model-driven development offers organizations tangible productivity improvements over previous approaches.

The Pet Store reference application

To showcase the practicality of the model-driven approach, let's examine the steps involved in generating a working application.

This example uses OptimalJ by Compuware as the tool for generating the model because OptimalJ supports MDA in its entirety. In addition, the example uses Java Pet Store reference application as the target application to allow for comparisons with manually developed versions.

The Java Pet Store is a sample application developed by Sun to illustrate the guidelines and patterns discussed in its Java Blueprints. The Java Pet Store models a typical e-commerce application (an online pet store). Numerous J2EE application server vendors implement this sample application to prove the compatibility of their products with the J2EE architecture.

Generating the sample application

Creating a J2EE application with a model-driven tool like OptimalJ is a three-step process. Each step follows the MDA approach, as shown in the figure below.

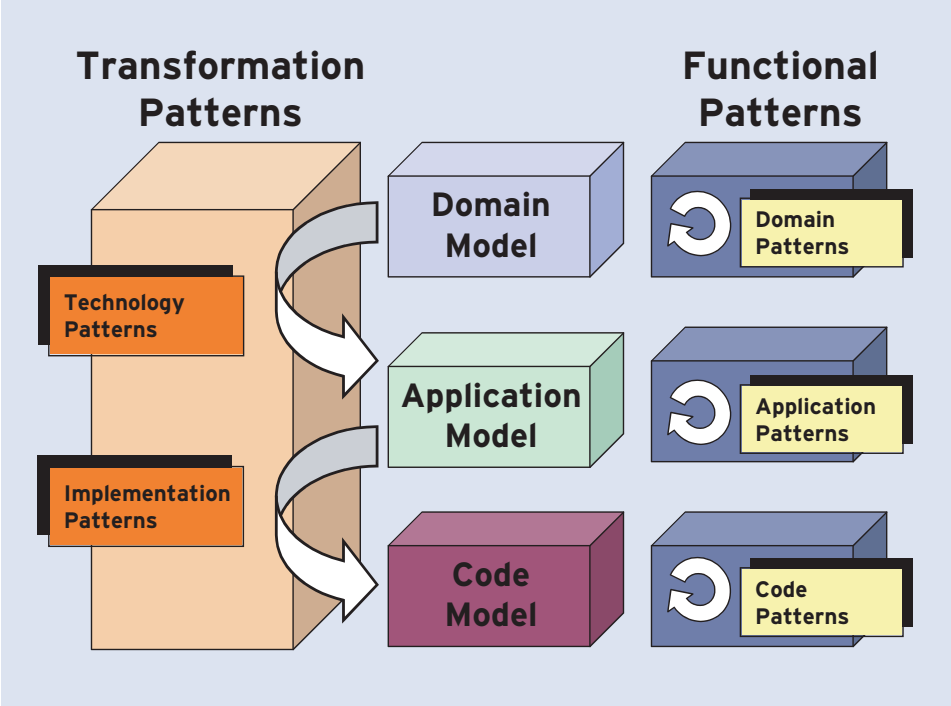

Figure 1: Three-step modeling process using OptimalJ

Step one: Domain model

The first step is to create a business-centric domain model. There are three ways to do so:

- Import an existing (UML) model using the standard XML Metadata Interchange (XMI) file format.
- Reverse-engineer an existing data source to generate the domain model.
- Develop the model from scratch.

At this point, the domain model is void of any technology platform-specific features. The freedom to model without the platform specifics simplifies the modeling process and enhances the future value of the model.

The model is then automatically checked for compliance with Unified Modeling Language (UML) standards.

Step two: Application model

Step two uses the business model from step one and transformation patterns to create a J2EE platform-specific application model. At this point, architects and designers have the opportunity to influence the solution domain by modeling the platform-specific tiers (Web tier, EJB tier, DBMS and integration tiers). For example, they can model complex views of the business as business services. At this point, they also integrate legacy systems and add Web services.

Step three: Code model

Step three uses the platform-specific application model and code model to generate the Java code that will be used to run the application.

Results

After completion of the domain model, it took less than 30 minutes to create the application model, generate the code, compile, deploy and start testing.

The accelerated development process resulted in a working J2EE application that can functionally manage the business domain. From this point forward, developers can use OptimalJ to create, update and remove business domain objects and the relationships between them – all without handcrafting any code.

The generated code for the user interface tier includes functional Java Server Pages (JSP) built using the Struts framework. Struts is a popular presentation framework that has been embraced by the Java development community.

OptimalJ also produced the necessary configuration and deployment files. As a result, hand-coding in effect starts in maintenance mode, instead of from scratch.

Making changes

Changing a J2EE application can be a difficult and cumbersome task. A simple change – such as adding an attribute to an entity – entails multiple different file changes. All the changes can lead to multiple errors. Many of them won't be detected, even at compile time.

With OptimalJ, a developer simply adds the new attribute to the domain model and regenerates the application. OptimalJ sees to it that all the different file changes are done automatically and free of errors. This process ensures that your business rules are intact, easy to validate and easy to test.

After completion of the domain model, it took less than 30 minutes to create the application model, generate the code, compile, deploy and start testing.

Adding the finishing touch

With OptimalJ, developers can use several Web front-end development tools to chisel in the final look. In this example, the developers used Macromedia Dreamweaver to complete the front end of the reference application. In this way, they made the pet store they had created using OptimalJ look and behave the same way as the Java Pet Store created by Sun.

The pie chart depicts the percentage of time developers spent writing different types of code for the application. As shown in Figure 2, almost 90 percent of the effort is in developing the front end. Most back-end development (for EJB Java and configuration) was handled automatically.

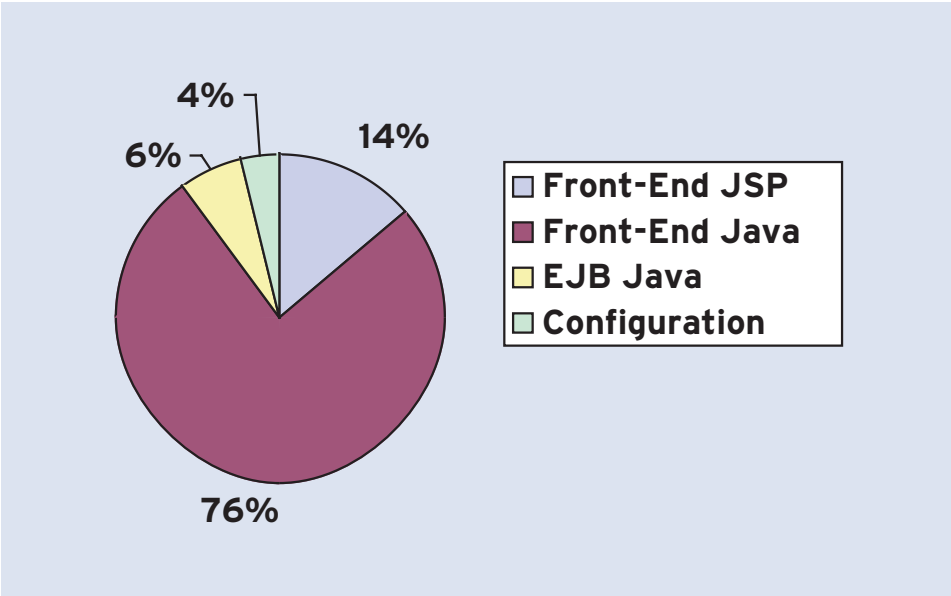

Figure 2: Percentage of time needed for different types of code development

Taking stock of efficiencies

The bar chart below compares the amount of handcrafted code created for two versions of Pet Store reference application and the MDA version generated with OptimalJ (EJB 1.1). For this comparison, the User Interface Tier Java code was counted (not JSP); lines of code for the administration functionality and the mailer application weren't counted – this functionality isn't included.

The chart shows that the MDA approach enables companies to develop complex applications with a very small amount of manual effort. In fact, benchmarking (through MDA scenarios like this one) shows that developers write only one to four lines of Java code manually for each 200 lines generated automatically.

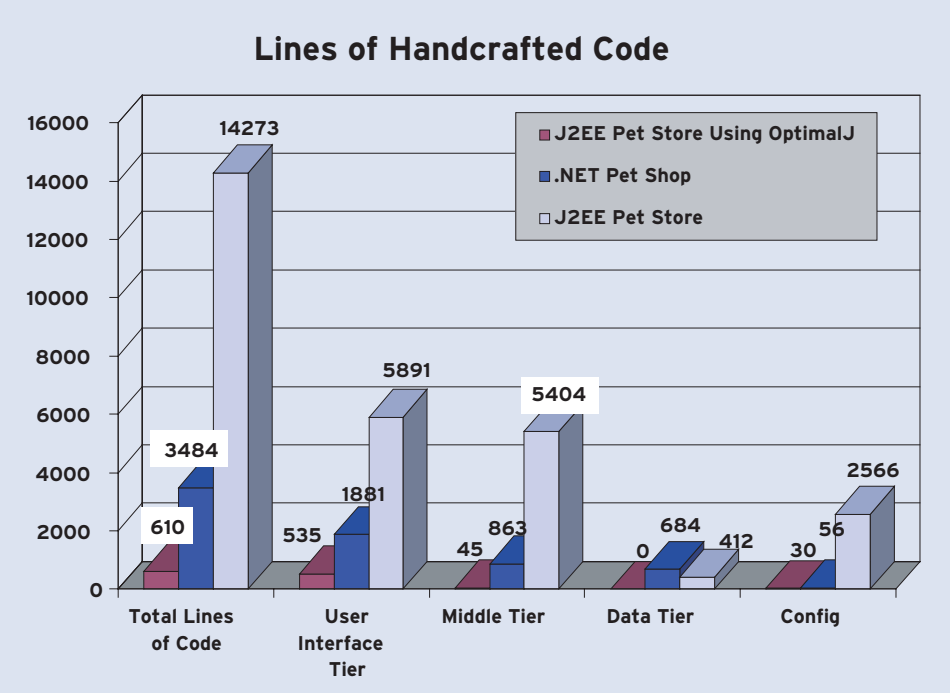

Figure 3: Lines of handcrafted code (application version comparison)

Optimizing today's assets for tomorrow

Model-driven tools are more flexible than code generators of the past. As new technology standards appear, developers and vendors simply create updated templates.

To examine how well OptimalJ handles new standards, developers generated two versions of the sample application. One version used EJB 1.1 API standards. The other version reused the original domain model and used EJB 2.0 API standards.

The result: An upgrade to EJB 2.0 that would have taken several months to do manually took less than 30 minutes with the model-driven approach. This clearly demonstrates how MDA protects existing investments in the domain models while leaving the door open to the benefits of innovative new technologies. The MDA out-of-the-box experience afforded by OptimalJ also proves that it's possible to deliver complex technology solutions using business model abstractions today.

Building agile and adaptable IT – for bottom-line benefits

IT leaders are dealing with the increasing complexity of new technology, legacy integration and changing standards. At the same time, IT managers are facing specific obstacles that keep them from delivering solutions with high velocity. Obstacles include high project startup and collaboration costs, inefficient stakeholder collaboration, constant changes in technology, and other development activities that have nothing to do with quality code.

MDA helps organizations achieve agile and adaptable IT so they can overcome these various business challenges. Through this evolutionary approach, collaborative teams can enjoy a wide range of compelling business benefits:

- Reduced development time for new applications
- Reduced cost throughout the application life cycle
- Improved application quality
- Increased return on technology investments
- Rapid inclusion of emerging technology benefits into existing systems

Deciding whether or not to adopt MDA

Despite all of its potential benefits, MDA isn't for every project. Here are some general considerations to keep in mind:

- The approach works best for large enterprises. (EDS believes that most large enterprises will adopt MDA principles.)
- Early adopters are organizations that have significant experience with UML modeling or that are starting new projects.
- Enterprises that implement MDA must do so only at the beginning of projects – never when they're under way or nearing completion.
- Companies that need or want the agility to evolve their business requirements independently from their IT will also find the approach beneficial.

MDA helps organizations achieve agile and adaptable IT so they can overcome these various business challenges.

Conclusion

If the MDA approach seems right for your enterprise — offering the time and cost savings that you're looking for — then give it careful thought. When applied well, MDA promotes demonstrable improvements in enterprisewide business agility. And that's a real key to success in today's volatile economy.

About the Author

Steven Witkop is a full-time developer for EDS. He received a bachelor's degree in MIS from Temple University in Philadelphia and a master's degree from Walsh University near Detroit. He is a Sun Certified J2EE architect, developer and programmer. Steve works in Detroit, Michigan, and has 12 years of application development experience.

About EDS

EDS, the world's largest independent information technology services company, provides strategy, implementation, business transformation and operational solutions for clients managing the business and technology complexities of the digital economy. EDS brings together the world's best technologies to address critical client business imperatives. It helps clients eliminate boundaries, collaborate in new ways, establish their customers' trust and continuously seek improvement. EDS, with its management consulting subsidiary, A.T. Kearney, serves the world's leading companies and governments in 60 countries. EDS reported revenues of $21.5 billion in 2002. The company's stock is traded on the New York Stock Exchange (NYSE: EDS) and the London Stock Exchange. Learn more at **eds.com**.

Let's begin the conversation

Corporate Headquarters

United States
5400 Legacy Drive
Plano, Texas 75024
1 800 566 9337

Regional Headquarters

Asia Pacific
33rd Floor, Citibank Tower
Citibank Plaza
3 Garden Road
Central
Hong Kong
852 2867 9888

Level 34, 100 Miller Street
North Sydney
New South Wales 2060
Australia
612 9025 0777

South Tower 68-86 Jervois Quay
PO Box 3647
Wellington
New Zealand
64 4 4950400

Canada
33 Yonge Street
Toronto, Ontario
M5E 1G4
Canada
1 416 814 4500
1 800 814 9038

Europe, Middle East and Africa
4 Roundwood Avenue
Stockley Park
Uxbridge
Middlesex UB11 1BQ
United Kingdom
44 20 8848 8989

Latin America
Avenida Presidente Juscelino
Kubitschek, 1830
5th Floor – Tower 4
04543-900
São Paulo
Brazil
55 11 3707 4100