Commercial Tools for Model Driven Architecture Software Development

A. Popandreeva

Key Words: *Software engineering; MDA; modeling; tools.*

Abstract. *MDA based software design approach specifies the system functionality with a platform independent model and then translates this model into platform specific model(s) and executable source code. It defines transformations from platform independent models to platform specific models. Different tools automate different phase of MDA software development. Some of them support model validation and model simulation, other code generation.*

Introduction

The international consortium Object Management Group (OMG) [18] develops standards for a wide range of computer technologies. The OMG's modeling standards, enable powerful visual design, execution and maintenance of software and other processes, including IT Systems Modeling and Business Process Management. Model Driven Architecture (MDA) is one of these standards[1], adopted as OMG specification in September 2001. The more detailed definition of the architecture „MDA Guide Version 1.0.1" is established in 2003. The main goals of MDA are portability, interoperability and reusability. MDA approach uses UML as a modeling language. MDA-based development maximizes software Return On Investment (ROI) integrating applications across the enterprise, and one enterprise with another.

The requirements of a system can be described graphically using UML through its diagram types, graphical and textual elements. The achievement of greater understanding of the requirements and easy transition of the model elements into source code of the implemented system leads to separation of the technical and domain aspects of modeling. The model of business represents accurately the business or user requirements. Thus, any change of the target technology or middleware does not break the model. The model of implementation provides maximum correspondence between model elements and source code generation of the implemented system. The separation of the technical and domain aspects of modeling improves the clarity and reuse opportunities for both these aspects of the design.

1. Model Driven Architecture

Model Driven Architecture (MDA) separates the system abstract architecture from platform architecture (*figure1*, OMG). In this way the application's logic separates from platform technology. The key to the MDA is a construction of the models and automatic transformations among them. Basically, three different types of models are created in the software development process: one for the functional specifications (CIM), one for high level details of the platform (PIM), one for the target platform (PSM). The Computation Independent Model (CIM) is a model based on requirements. The Platform Independent Model (PIM) describes the structure and function of the application or system, but not the specific implementation. It can be transformed through MDA to Platform Specific Model (PSM) which presents an implementation on virtually any platform. A single element of PIM can be converted to multiple PSM(Platform Specific Model) elements across multiple domains.

2. Benefits of Using MDA

There are many benefits of using MDA approaches and some of them are the following:

√ Designers create domain model according desired application's functionality and behavior without thinking how to implement it.

√ Many platform specific models can be generated from one PIM.

√ MDA approach reduces development time and efforts.

√ New MDA templates are designed.

√ An architecture based on the MDA is easily adapted to contemporary platform technology.

√ Provides interoperability capabilities between different technologies.

√ The models at all development levels and transformations between them lead to traceability in the development process.

---

[1] *All of OMG specifications may be downloaded without charge from OMG's website.*

[2] $ROI = \dfrac{(Gain\,from\,Investment - Cost\,of\,Investment)}{Cost\,of\,Investment}$
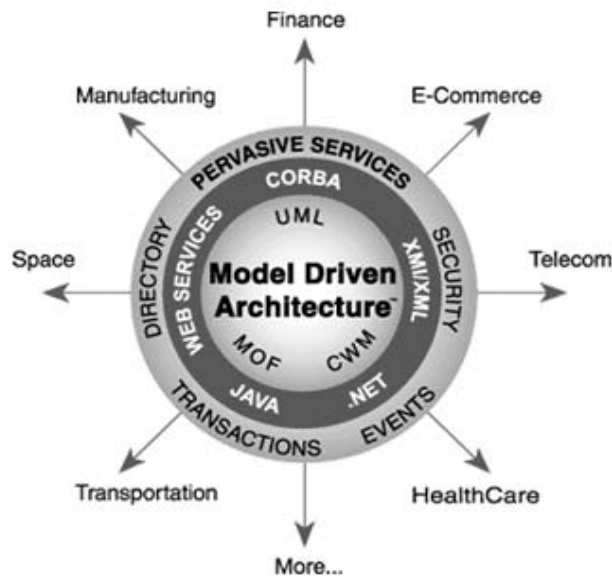
Figure 1. MDA overview

3. MDA Transformations

MDA transformations convert PIM to PSM to implement the developed abstract system across a variety of platforms. The approaches for MDA transformations are the following:

√ **Manual transformation.** The design decisions are taken during transformation from PIM to PSM.

√ **Transforming a PIM using a Profile.** PIM and PSM are UML profiles. The specification of a transformation may be a UML profile.

√ **Transformation using patterns and markings.** Patterns and marks are used for transformations specification.

√ **Automatic transformation**. The developer specifies the required behavior and puts all the information necessary to produce computer program code in the PIM. The automated transformation needs transformation rules, which can be represented as a model – the so-called metamodel.

4. MDA Tools

The automation of the MDA approach requires languages for specification of models, definition of transformations and description of metamodels. Different tools are developed to support MDA development. The evaluation criteria and definition of an ideal MDA tool are defined [7].

The following different kinds of MDA tools can be distinguished as follows:

√ Modelling tools for development CIM, PIM or PSM models.

√ Transformation tools that perform model transformations (e.g. PIM -> PSM) in a flexible and configurable way.

√ Tools that support (xUML) „Executable UML" and convert implementation-independent specifications to executable applications.

4.1. Tools for Analysis, Design and Code Generation

Some of the tools are software modeling and construction tools. Usually they are created as an environment that supports the full life-cycle of system development from requirements gathering, through analysis, modeling, implementation, testing to deployment and maintenance. It is usable with different operating platforms (J2EE, .NET, CORBA, etc.), Web or application servers. Such environment consists of a modeling tool for PIM creation, a transformation engine for transformation PIM into PSM, and a code generator. The modeling tool performs visual modeling with UML and UML Profiles. PIM transforms to multiple PSMs. The code generator produces runable code in most languages (Java, C++, C#, PHP, Delphi, etc.). Such tools are

√ SparxSystems' *Enterprise Architect*[24] which supports XMI import/export to other modeling tools, modeling database systems and generating DLL.

√ Borland *Together* [6] which pioneered the simultaneous bi-directional transformation between PSM and the code model.

√ Consyst's *REP++ Studio* [8] that removes all separations between the data structures, properties and parameters of interfaces, programs and databases. It

uses a unique reflection approach that continuously synchronizes and modifies the application and services according the metadata stored in the repository. It is the perfect complement to the .NET and J2EE™ development platforms

√ IBM *Rational Rhapsody* [11] validates the requirements, design and behavior of the software early in the lifecycle when defects are less costly to fix. It integrates with other IBM products and exists in different editions. It is usable with languages Java, C, and C++ and operating systems Windows and Linux.

√ Softeam's *Objecteering* 6 Enterprise Edition [20] integrates with Eclipse and supports development of information and technical systems. User-friendly graphical editors guide the user through model actions.

√ Kennedy Carter's *iUML* [12] is a multi-user environment for development, integration, validation and simulation xUML models of a C++, Java and embedded C applications. It supports use-case modeling and management textual requirements.

√ Mentor Graphics' *Bridge Point* [16] integrates with Eclipse and supports development of C and C++ real-time and embedded applications.

4.2. Modeling Tools

For development the UML model of business processes and their MDA transformation the following tools can be used:

√ *Ameos/ACD* [2] supports modeling using UML and UML2 Profile. Color can be assigned to UML Profiles and to Model Elements to achieve readable models. The Ameos has an open source version (Open Ameos).

√ Artisan *Studio* [4] supports OMG SysML standard to integrate requirements inside the model and works with requirements not just in the systems analysis phase of a project but throughout the entire project's lifecycle. It supports transferring HOOD design information into UML model without loss of any design data or integrity. The Artisan Studio allows exchange the information with MathWorks Simulink.

√ *Pattern Waver* [10] provided by Founds Tao Inc. is a UML based modeling tool for design with several editions for different purposes (robotics, etc.). Pattern Weaver can be run as an *Applet* and designed diagrams will be published on the Web. Everyone who has an interest in the model can discover and manipulate diagrams without installing any browser plug-ins.

4.3. Transformation and Code Generation Tools

Some of the tools support XMI import/export to other modeling tools, transform PIM to PSM and generate application code.

√ *Accelerated Construction Engine* (ACE) [25] is suitable for development multi-tiered and Web-Transaction applications in the field of Enterprise, E-Business and Telecom. It is usable with languages JAVA, EJB, JSP, XML, and platforms CORBA, SOAP, SNMP, and RDB.

√ realMethods *Framework* [22] uses J2EE patterns and best practices to generate an application code.

√ *PathMATE* [21] is fully integrated with Eclipse and the Rational development software and transforms MDA PIM into C, C++, and Java application code for embedded systems. It supports model-level testing which reduces the coding errors.

√ *BLU AGE* [18] is a generator for UML models' execution and transformation into business applications on JAVA and .NET environments. It offers also extraction of UML models from the source code of existing Java and non Java application (Reverse engineering).

√ LIANTIS's *XCoder* [13] is an Open Source tool for model transformation and code generation for Java, EJB, C++, C# and SQL.

√ *AndroMDA* [3] is an Open Source MDA Generator. Custom transformations for different architectures can be plug-in. They are packaged as the so-called cartridges.AndroMDA comes with an array of ready-made cartridges for common architectures like Spring, EJB, .NET, Hibernate, Struts and even more.

√ Blueprint ME [1] supports the structured software design and code generation from UML diagrams.

4.4. Code Generators

BITPlan *smartGenerator* [5] which integrates with different modeling tools, produces runable code for desired target language, integrates with new technologies.

Domain Solutions' *CodeGenie* [9] translates object-oriented UML analysis application models into code. It is XMI integrated with leading UML editors. It consists of

• Model Reader that reads model content from the major CASE tools into a meta-model.

• Model Browser to browse and display the meta-model content.

• Model Auditor to check executable UML models for completeness and consistency.

• CodeGenie generator to generate the software for application models from an architecture meta-model.

CodeGenie can be fully customised to accommodate the client's chosen software processes, the target software architecture, the software development environment and

the configuration management tools. It is usable with Java and C++ languages.

4.5. Free Access and Some Applications

Free download is available for REP++ Studio, BlueprintSM, BlueprintSM as a part of BlueprintME, a version of Pattern Waver, XCoder. After registration it is possible to free download CodeGenie, PathMATE, iUML.

Ameos/ACD has a lot of applications in various industries automated telecommunication and safety critical application. Artisan Studio is the tool of choice for development technical systems, complex mission-critical systems and software engineering. BITPlan smartGenerator integrates with new technologies such as mobile appliances. IBM Rational Rhapsody is an environment for real-time or embedded systems engineering. PathMATE is suitable for embedded systems development. Bridge Point is suitable for real-time and embedded applications and has many successful applications in the automotive industry.

Conclusion

MDA is a new approach to software development based on building models.   The models present different views of developed software. Different tools   simplify the development of software using a MDA approach. Some of them are only modeling tools, other are code generators. Some of the tools are environments that support the full life-cycle of the software development. Using the tools is possible generation and execution of the program code.

References

1. @-portunity the MDA-enabling company.
<http://www.atportunity.com/>.
2. Ameos – The Next Generation Modeling Tool.
<http://www.aonix.com/ameos.html>.
3. AndroMDA. <http://www.andromda.org/>.
4. ARTiSAN's Real-time Studio.
<http://www.artisansoftwaretools.com/>.
5. BITPlan smartGENERATOR. <http://www.bitplan.com/com/bitplan/web/index.php?topic=products/smartgenerator>.
6. Borland. <http://www.borland.com/>.
7. Calic, T., S. Dascalu, Dw. Egbert. Tools for MDA Software Development: Evaluation Criteria and Set of Desirable Features. Itng Fifth International Conference on Information Technology: New Generations (itng 2008), 2008, 44-50.
8. ConsystSQL. <http://www.consyst-sql. com/a/www/Accueil/Accueil.html>.
9. Domain Solutions Ltd (Code Genie). http://www.domsols.com/ or <http://www.domainsolutions.co.uk/>.
10. Founda,Tao. <http://www.foundatao.com/>.
11. IBM. Rational Rhapsody. <http://www.ibm.com/software/awdtools/rhapsody/>.
12. Kennedy, Carter. <http://www.kc.com/products.php>.
13. LIANTIS, XCoder. <http://www.liantis.com/Model-Driven-Archite.96.0.html>.
14. MDA Guide version 1.0.1 (ed. J. Miller and J.Mukerji), OMG, 2003.
15. MDA Overview. Sparx Systems, 2007,
<http://www.sparxsystems.com>.
16. Mentor Graphics.   <http://www.mentor.com/>.
17. McLaughlin, L. Automated Programming: The Next Wave of Developer Power Tools. – *IEEE Software*, 23, May/June, 2006, 91-93.
18. Netfective Technology Group. BLU AGE Corp.
<http://www.bluage.com/>.
19. Object Management Group (OMG). <http://www.omg.org/>.
20. Objecteering, the Model-driven Development Tool.
<http://www.objecteering.com/>.
21. Pathfinder Solutions. PathMATE.
<http://www.pathfindermda.com/>.
22. RealMethods, realMethods Framework.
<info@realmethods.com>.
23. Softeam Think Object, Objecteering/UML: The First CASE Tool to Completely Support the MDA Approach of the OMG.
<http://www. softeam.com/produits.php>.
24. SparxSystems. <http://www.sparxsystems.com.au/>.
25. TechOne – Building Solutions for Tomorrow.
<http://www.techone.com/>.

Anelia Popandreeva *received a M. Sc degree in Electronics at Technical University of Sofia. She expects to receive her PhD in 2010. Her research interests are in software engineering. She has authored and co-authored more than 90 scientific journal and conference papers. She is a member of IEEE. Anelia works currently as research associate in Institute of Control and System Research, Bulgarian Academy of Sciences.*

Contacts:
ICSR – Bulgarian Academy of Sciences,
Ac. G. Bonchev St., bl. 2,
Sofia 1111, Bulgaria
e-mail : popandreeva@icsr.bas.bg