Model Driven Architecture: what is and where is it going

Prof. Dr. Petraq Papajorgji

Model-driven development (MDD) represents the approach in software engineering that pushes for the use of graphical models and pre-built application components so that users can visually construct complex applications in a short time. Although currently not being a standard in software development, Interest for this approach continues to increase as individuals experience the ease and speed with which technical and business users orchestrate and deploy business applications. The right model-driven development platform provides significant productivity advantages over traditional development methods and enables project delivery by smaller teams [2].

At the heart of this approach stand two important concepts: abstraction and automation. Abstraction is used to build models using graphic notations such as the Unified Modeling Language (UML). These models represent a high level of abstraction as they represent the concepts of the business domain and their connections to be used for the solution of the problem. These models are transformed to working software applications using automated transformations to execute the model or interpret it. Several approaches following the MDD principles have resulted to nice and very good tools that help enormously during the process of design and implementation of the software.

Currently, there are no strict standards defining what the MDD should follow. Many companies provide their own version of their MDD tool but reading someone’s model using another tool, is really a challenge. The Object Management Group (OMG) provided the principles of Model Driven Architecture (MDA), its view of the Model Driven Development approach. Model Driven Architecture supports evolving standards in applications as diverse as enterprise resource planning, air traffic control and human genome research; standards that are tailored to the needs of these diverse organizations, yet need to survive changes in technology and the proliferation of different kinds of middleware [1]. MDA is built using OMG standards, including: Unified Modeling Language™ (UML®), the ubiquitous modeling notation used and supported by every major company in the software industry; XML Metadata Interchange™ (XMI®), the standard for storing and exchanging models using XML™; and CORBA®, the most popular open middleware standard [1].

In 2014, OMG adopted the revised “MDA Guide Revision 2.0” as a more detailed definition of the architecture. The primary feature of MDA is defining the structure, semantics, and notations of models using industry standards. MDA models can then be used for the production of documentation, acquisition specifications, system specifications, technology artifacts (e.g. “source code”) and executable systems [1].

The MDA Guide, Revision 2.0, states that “The Meta Object Facility (MOF) provides a key foundation for OMG’s Model-Driven Architecture, which unifies every step of development and integration from business modeling, through architectural and application modeling, to development, deployment, maintenance, and evolution. MOF uses ‘meta-models’ specified in UML® to describe modeling languages as inter-related objects [1].

Another important basis for the MDA approach is The Unified Modeling Language (UML). As for each solution to a problem are two levels of models: Platform-Independent Model (PIM), and one or more corresponding Platform-Specific Models (PSM). Usually, these models will be defined in UML, making OMG’s standard modeling language a foundation of the MDA. (Use of UML, although common, is not a requirement; MOF is the mandatory modeling foundation for MDA) [6].

UML 2.0 is provided with the Object Constraint Language (OCL) in order to make models more precise [8-9]. The use of OCL facilitates enormously software development as makes modeling more abstract and safer.

Although UML is well established as the modeling language, its use in all areas of activity would be difficult as each area or domain has its own partic-
Model Driven Architecture: what is and where is it going

Model Driven Architecture: what is and where is it going

The UML is designed to have profiles; a profile is a set of notations for a particular field of activity. Profiles are official OMG specifications [1]. The current suite of profiles includes:

- **The UML Profile for CORBA™**, which defines the mapping from a PIM to a CORBA-specific PSM.
- **The UML Profile for CCM™** (the CORBA Component Model), OMG’s contribution to component-based programming. Enterprise JavaBeans (EJBs) are the Java mapping of CCM; an initial take on a profile for EJB appears as an appendix of the UML 2.0 Superstructure specification, linked above.
- **The UML Profile for EDOC™** is used to build PIMs of enterprise applications. It defines representations for entities, events, process, relationships, patterns, and an Enterprise Collaboration Architecture. As a PIM profile, it needs mappings to platform-specific profiles. A mapping to Web Services is underway now; additional mappings will follow.
- **The UML Profile for EAI™** defines a profile for loosely-coupled systems - that is, those that communicate using either asynchronous or messaging-based methods. These modes are typically used in Enterprise Application Integration, but are used elsewhere as well.
- **The UML Profile for Quality of Service (QoS) and Fault Tolerance™** defines frameworks for Real-time and high-assurance environments.
- **The UML Profile for Schedulability, Performance, and Time™** supports precise modeling of predictable - that is, real-time - systems, precisely enough to enable quantitative analysis of their schedulability, performance, and timeliness characteristics.
- **The UML Testing Profile™ (UTP™)** provides important support for automated testing in MDA-based development environments.

Another important part of the MDA definitions are XML METADATA INTERCHANGE (XMI). XMI defines an XML-based interchange format for UML and other MOF-based meta-models and models (since a meta-model is just a special case of a model), by standardizing XML document formats, DTDs, and schemas. In so doing, it also defines a mapping from UML to XML. Because one of OMG’s XMI updates reflects the incorporation of XML Schemas, while MOF point updates were made periodically through OMG’s established maintenance process, numbering of XMI and MOF versions diverged [1].

A number of companies in the software industry are using the MDA approach such as U.S. Government Intelligence (https://www.omg.org/mda/mda_files/popkinconquest.htm), Deutsche Bank Bauspar AG (https://www.omg.org/mda/mda_files/SuccessStory_DBB_4pages.pdf), Lockheed Martin (https://www.omg.org/mda/mda_files/LockheedMartin.pdf) etc.

Although the number of companies using the MDA approach is increasing, it is hard to say that the MDA approach has been accepted by the mainstream audience in the software industry. A number of scientific studies have been undertaken to understand the hesitation of the software industry vis-à-vis the MDA approach. Kos & Mernik [4-5] have pointed out that most of the MDA tools come without a debugging or refactoring capabilities. A tool that lacks debugging and refactoring abilities is hard to be used and certainly, will not be well accepted by the software industry. Inadequate support by Integrated Development Environments, and poor interoperability with mainstream languages, can be contributing factors for the resistances of MDDE within the software industry [6].

It is a fact that the MDA approach provides a lot of advantages for the software development. Then why this approach is not largely used? The answer to this question may stand in the universities hands as universities should provide more adequate courses to introduce the MDA to students. Another obstacle of the use of MDA is the inertia in companies that hesitate to embrace new technologies. It is known that the use of new technologies comes with a price and this often pushes companies to postpone the decision of accepting new technologies.
References

[1]. Object Management Group, OMG.org
[2]. Mendix.com
[7]. The MDA foundation model. ormsc/2010-06-01.