

Developing Methods for Using Model-Driven Architecture to Develop Quality Software Products at Low Cost Entirely by Re-Use of Existing Components Ministry of Science, Technology and Innovation, Malaysia, project 01-01-06-SF0431 April 2009 to September 2010

Project summary

Model-driven engineering is the development of engineering products primarily by describing their specifications in abstract modeling languages. These languages either support modeling at the application or business level, or describe a rich set of structured parameters for standard platforms for particular implementation technologies. The idea is to describe the problem using an application level modeling language, then to map the problem specification onto the parameters of an implementation platform, so that the engineering product can be produced without any detailed development. The end result will be to greatly reduce the cost of and time needed for creating software, and also greatly increase the quality of the product.

The best-known and most-developed model-driven engineering initiative in the computer software field is Model-Driven Architecture (MDA), an initiative of the Object Management Group (OMG), a software engineering standards body supported by most of the major players in the industry worldwide. Much of the work of the OMG since 2001 has been in support of MDA.

There is an enormous amount of conceptual infrastructure needed to make MDA a practicable technology. There are a number of products available implementing the various MDA-related specifications. Further, since MDA is a very immature technology, there is an open source platform called Eclipse supported by IBM, together with a number of open source tools, including the Eclipse Modeling Framework, contributed by various IBM laboratories and other organizations.

The research problem is that nearly all the effort so far in MDA has been in developing specifications and components of the infrastructure. Not much work has been done in how to organize these components to provide complete software solutions for problems in particular domains. This project will focus on complete solutions, developing methods for integrating the components.

Literature review summary

Model-Driven Architecture is a technology with a long history, going back at least as far as the invention of COBOL (Common Business-Oriented Language) in the 1950s. (It was first thought that COBOL code could be written by business rather than information technology specialists.) The history includes compilers for a wide variety of programming languages; a wide variety of modeling tools; the development of software libraries in statistics, numerical computation, and many other fields; the development of enormous bodies of reusable code in the form of operating systems and database management systems; many application-specific languages in many different fields; and many fourth-generation languages mostly used to develop information systems with very little low-level programming.

Code re-use has been a preoccupation of many software engineers since the 1980s. (Code re-use efforts have been focused on code that is hard to re-use, and tends to ignore the vast body of re-usable code noted in the previous paragraph.) Object-oriented programming was introduced in the 1980s as a way to facilitate code reuse. By the mid-1990s, it had become clear that developing good libraries of re-usable programs was very much like developing new mathematics, and was very difficult. Recent work is on a much larger scale, with coordinated collections of software modules aimed at particular kinds of applications, called Product Line Architecture. [Jarzabek]

Model-Driven Architecture can be seen as contributing a uniform way of representing the different modeling systems, program interfaces, and the auxiliary structures needed to model and implement applications (using a subset of UML [UML2] called Meta-Object Facility or MOF [MOF]). UML itself is a collection of related modeling tools, supported by other

specifications like the constraint language OCL [OCL], an XML-based interchange format XMI [MOF XMI] and the mapping language QVT [MOF QVT]. This uniform representation makes possible the development of generic tools like the Rational suite of software [Rose].

Model-Driven Architecture is based on concepts and terminology published by the OMG [MDA], and there is a considerable body of material published by the OMG and various member organizations [OMG].

There is a body of work concentrating on end-to-end application development in a model-driven environment, called Domain-Specific Modeling Languages, from which useful insights can be drawn. This work tends to concentrate more on lower-level programming language implementation than does MDA. [ISIS]

References

- [ISIS] <http://www.isis.vanderbilt.edu/projects/gme>
- [Jarzabek] Jarzabek, Stanislaw (2007) Effective Software Maintenance and Evolution: A Reuse-Based Approach, CRC Taylor and Francis.
- [MDA] MDA Guide Version 1.0.1 document omg/2003-06-01, 12th June 2003
- [MOF] Meta Object Facility (MOF) Core Specification, Version 2.0. OMG Specification, formal/06-01-01. Latest version is available at <http://www.omg.org/docs/formal/06-01-01.pdf>.
- [MOF QVT] MOF QVT Final Adopted Specification, ptc/05-11-01. Latest version is available at <http://www.omg.org/docs/ptc/05-11-01.pdf>.
- [MOF XMI] MOF 2.0/XMI (XML Metadata Interchange) Mapping Specification, v2.1. OMG Specification, formal/05-09-01. Latest version is available at <http://www.omg.org/docs/ptc/05-09-01.pdf>.
- [OCL] OCL 2.0 Specification, Version 2.0. OMG Available Specification (OCL 2.0 FTF convenience document), ptc/2005-06-06. Latest version is available at <http://www.omg.org/docs/ptc/05-06-06.pdf>.
- [OMG] <http://www.omg.org/mda/presentations.htm>
- [Rose] IBM Rational Rose, <http://www-130.ibm.com/developerworks/rational/products/rose>
- [UML2] Unified Modeling Language: Superstructure, version 2.0. OMG Specification, formal/05-07-04. Latest version is available at <http://www.omg.org/docs/formal/05-07-04.pdf>.
- [UML Infra] Unified Modeling Language: Infrastructure, version 2.0. OMG Specification, formal/05-07-05. Latest version is available at <http://www.omg.org/docs/formal/05-07-05.pdf>