Chapter III
Examples and Evidence

Sowmya Karunakaran
MDA Research Initiative, Chennai, India

ABSTRACT

This chapter aims at highlighting the increased development productivity and quality that can be achieved by Model Driven Software Development (MDSD). The above statement is substantiated by discussing many experiments and case studies in the field of Model Driven development. The chapter will contain the study of various cases in which the Object Management Group's (OMG) Model Driven Architecture (MDA) has been used as a framework to build different applications. The reader will be provided with an overview of how the MDA paradigm greatly expedites application development with the proper tool support. The main emphasis will be on providing case studies for the measurement of the quality of the models.

INTRODUCTION

The software industry remains reliant on the craftsmanship of skilled individuals engaged in labor intensive manual tasks. However, growing pressure to reduce cost and time to market, and to improve software quality, may catalyze a transition to more automated methods. Evaluation of these methods will require more experiments and practical experience. Establishing of a comprehensive collection of benchmark problems would be a valuable next step in that direction. This chapter will serve as:

- A practical guide for software architects and developers as it is peppered with practical examples and extensive case studies.
- An Enchiridion for Model driven software quality assurance
- A rich resource, containing prominent examples for constructive Quality Assurance (QA) of MDSD
- A handbook of various techniques that can be employed for identifying the quality of models and model transformations.
- A trend analyzer, as it will give a broad overview on the tools and methods used by various researchers and professionals.
BACKGROUND

MDA has enjoyed high visibility since its formal announcement by the OMG in March 2001. In three short years, well over 40 companies have come forward with software products said to implement MDA; while a smaller, but significant, number of success stories demonstrate that there really is something to this striking new concept.

It is no exaggeration to say that MDA has the potential to revolutionize the way we create and maintain software. Since MDA is becoming so popular, it is important to understand clearly what it is—and what it is not, but more importantly it is necessary to analyze the quality benefits it offers.

Let us see what the industry experts have to say about MDA:

MDA offers organizations several distinct advantages ... One of them which is, Interoperability and portability. The platform independence of the first stages of MDA development makes it easier to interoperate with, or even move to, different middleware. Given that the middleware space is crowded with Enterprise Java Beans, CORBA, Web Services, SOAP, Ch#, .NET and others, this represents a huge savings in time and cost. (Grady Booch)

Customer projects that previously took six months to complete are taking four using MDA. (PFPC Inc. in Wilmington, Del.)

Sophisticated organizations see the benefits of a model-driven approach, which is the future of effective software design. (Peter Young, Vice President, SUN ONE Studio Tools, Sun Microsystems)

MDA is not about generating complete applications from diagrams; it’s about generating all the linkages to integrate applications from Unified Modeling Language (UML) diagrams. It’s also about having a common, high-level UML model of integration that can generate whatever proxies, bridges, and protocols are required to integrate a new application with those already in existence. (Wells Fargo Bank; IBM)

In the context of MDSD, the creation of models and model transformations is a central task that requires a mature development environment based on the best practices of software engineering principles. In a comprehensive approach to MDSD, models and model transformations must be designed, analyzed, synthesized, tested, maintained and subjected to configuration management to ensure their quality. Working with multiple, interrelated models that describe a software system require significant effort to ensure their overall consistency. It follows that automating the task of model consistency checking and synchronization would greatly improve the productivity of developers and the quality of the models.

This chapter will give the reader an insight on how model driven architecture has been used to build different kind of applications that run on disparate platforms. It will also highlight the benefits like increased productivity, improved software quality and reduced complexity that were obtained by means of following a MDA approach.

The software industry is often ready to profess that MDA methodology and tools can produce code more quickly than manual hand-coding (the traditional way of software development) can. But unfortunately the industry is less ready to accept that these code generators can produce good quality code. The biggest concern is that these generators will turn out to be tumescent, one-size-fits all and on the whole produce impractical code.

Interestingly, the concerns that the software industry currently has parallel those that programmers/ software developers had years ago when “third generation languages” (3GLs) such as CO-