Chapter 5
Not Ready for Prime Time: A Survey on Security in Model Driven Development

Jostein Jensen
Norwegian University of Science and Technology, Norway

Martin Gilje Jaatun
SINTEF, Norway

ABSTRACT

Model Driven Development (MDD) is by many considered a promising approach for software development. This article reports the results of a systematic survey to identify the state-of-the-art within the topic of security in model driven development, with a special focus on finding empirical studies. The authors provide an introduction to the major secure MDD initiatives, but the survey shows that there is a lack of empirical work on the topic. The authors conclude that better standardization initiatives and more empirical research in the field is necessary before it can be considered mature.

1. INTRODUCTION

Model Driven Development (MDD) has been considered a promising approach to software development since its introduction about a decade ago. The Object Management Group (OMG, 2010) is the most prominent standardization body within the MDD domain, and has developed a framework for model driven development called Model Driven Architecture (MDA). MDA is a framework for developing applications and writing specifications, where improved portability, platform independence and cross-platform interoperability are among keywords used by OMG to describe the benefits of using this framework.

Kleppe et al. (2003) present the MDA development lifecycle. The basis for development is platform independent models (PIM), which

DOI: 10.4018/978-1-4666-2482-5.ch005

specify functionality and behavior. These models are abstracted away from the technology that will be used to realize the system. PIMs can then be transformed into platform specific models (PSM), adding technology specific details to the PIM. PSM again can then be transformed into code. Kleppe and colleagues also mention a third model type used during the requirements and analysis phase of development, called computational independent model (CIM).

Figure 1 shows the MDA software development lifecycle as it is depicted by Kleppe et al. (2003). The ovals to the left represent generic software development phases, while the squares to the right represent artifacts produced in an MDA context. Artifacts developed during the requirements phase and used for analysis are often referred to as Computational Independent Models (CIM). Platform independent models (PIM) are abstract representations of the system to be built, and independent of any implementation technology. PIMs are transformed, preferably automatically using tool support, to Platform Specific Models. These are specific to the technology that will be used to realize future systems. Continuing the MDA lifecycle, PSMs are transformed into code. Since PSMs are close to the technology, this transformation is by some considered to be straightforward (Kleppe et al., 2003).

Note that real life seldom has a perfect match for theoretical frameworks such as the MDA lifecycle presented in Figure 1. Thus, in concrete examples one will not always find that all the models such as CIM, PIM and PSM are actually used in practice, and in such cases one must modify the map to fit the terrain.

PIMs form the basis for low-level system designs and as such constitute an important part of a system's documentation (while still providing important abstractions). The layering between platform independent models, platform specific models and code are the key to solve problems related to portability, platform independence and interoperability. Developers are mainly supposed to work with the platform independent models, and since these are platform and technology neutral it should be a relatively simple task to transform them into different platforms and technology solutions.

In traditional software development, security aspects are often considered late in the development lifecycle, if they are considered at all (Wyk & McGraw, 2005). However, the cost of eliminating

Figure 1. MDA Software development lifecycle