

Chapter 104

Model–Driven Software Modernization

Liliana Maria Favre

Universidad Nacional Del Centro De La Provincia De Buenos Aires, Argentina

Liliana Martinez

Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina

Claudia Teresa Pereira

Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina

ABSTRACT

Modernization of legacy systems is a new research area in the software industry intended to provide support for transforming an existing software system to a new one that satisfies new demands. This chapter analyzes software modernization based on the architecture-driven modernization (ADM). In this context, software modernization is supported by metamodels to describe existing systems, discoverers to automatically create models of these systems, and tools to understand and transform complex models created out of existing systems. This chapter provides an overview of the-state-of-the-art in ADM-based software modernization techniques. Taxonomy of different techniques is described. A description of how traditional techniques such as static and dynamic analysis can be integrated with ADM standards is presented. This chapter also analyzes current challenges and strategic directions in software modernization.

INTRODUCTION

Most vital software systems in companies have been developed several years ago with technology that today can be considered obsolete and unaligned with their current strategic objectives. These legacy systems play a central role in the company's information system. They are often large business-critical applications that have involved the investment of money, time and other resources through the years. Therefore, companies are facing the problematic of having to modernize or replace their legacy software systems.

Unlike the maintenance, software modernization involves significant changes in the structure and functionality of a legacy system with the aim of increasing its strategic value. Software modernization, understood as technological and functional evolution of legacy systems, provides principles, methods,

DOI: 10.4018/978-1-5225-7598-6.ch104

techniques and tools to support the transformation from an existing software system to a new one that satisfies new requirements. It is related to reverse engineering, restructuring, forward engineering and reengineering processes. The definitions given by Chikofsky and Cross (1990) for these processes are still in force beyond the technological evolution:

- *Reverse engineering* is the process of analyzing a subject system to identify the system's components and their interrelationships and, create representations of the system in another form or at a higher level of abstraction.
- *Restructuring* is the transformation from one representation form to another at the same relative abstraction level while preserving the subject system's external behavior (functionality and semantic).
- *Forward engineering* is the traditional process of moving from high-level abstractions and logical, implementation-independent designs to the physical implementation of a system.
- *Reengineering* is the examination and alteration of a subject system to reconstitute it in a new form and the subsequent implementation of the new form.

The term *Refactoring*, introduced by Martin Fowler, defines a special kind of restructuring in object-oriented code (Fowler, 1999). One of the advantages of software modernization versus traditional software reengineering is that the former allows discovering and refactoring object-oriented models at a high level of abstraction, such as business process models.

Software modernization faces many challenges due to the proliferation of new technologies such as mobile computing and cloud computing. Thus, novel technical frameworks for information integration and tool interoperability are needed. The *Model Driven Development* (MDD) appears to be an interesting approach to address these challenges since it provides principles and techniques to represent software systems through models at different levels of abstraction. MDD refers to a range of development approaches based on the use of software models as first class entities. The most well-known realization of MDD is the OMG standard Model Driven Architecture (MDA) (MDA, 2016). The outstanding ideas behind MDA are separating the specification of the system functionality from its implementation on specific platforms, managing the software evolution from abstract models to implementations increasing the degree of automation of model transformations and achieving interoperability with multiple platforms, programming languages and formal languages. The essence of MDA is the Meta Object Facility Metamodel (MOF) that allows different kinds of software artifacts to be used together in a single project (MOF, 2015). Models play a major role in MDA which distinguishes Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM). Some authors also distinguish Implementation Specific Model (ISM) as a description (specification) of the system in source code. To express transformations, OMG defined the MOF 2.0 Query, View, Transformation (QVT) metamodel (QVT, 2015).

OMG Architecture-Driven Modernization Task Force (ADMTF) is developing a set of specifications and promoting industry consensus on modernization of existing applications (ADM, 2016). In the ADM context, software modernization can be summarized as follows. First, information is extracted out of the system artifacts. Second, this information is analyzed in order to take adequate modernization decisions and finally, the information must be transformed to new artifacts. These steps are supported by metamodels to describe existing systems, discoverers to automatically create models of these systems and, tools to understand and transform complex models created out of existing systems. The *Knowledge*

11 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/model-driven-software-modernization/214710

Related Content

The Past, Present, and Future of UML

Rebecca Platt and Nik Thompson (2019). *Advanced Methodologies and Technologies in Network Architecture, Mobile Computing, and Data Analytics* (pp. 1452-1460).

www.irma-international.org/chapter/the-past-present-and-future-of-uml/214713

A Research Approach to Detect Unreliable Information in Online Professional Social Networks: Using LinkedIn Mobile as an Example

Nan Jing, Mengdi Li and Su Zhang (2015). *International Journal of Handheld Computing Research* (pp. 39-56).

www.irma-international.org/article/a-research-approach-to-detect-unreliable-information-in-online-professional-social-networks/148288

Distributed Mobile Services and Interfaces for People Suffering from Cognitive Deficits

Sylvain Giroux, Hélène Pigot, Jean-François Moreau and Jean-Pierre Savary (2009). *Mobile Computing: Concepts, Methodologies, Tools, and Applications* (pp. 1069-1079).

www.irma-international.org/chapter/distributed-mobile-services-interfaces-people/26570

Analysis of the Realistic Resolution with Angle of Arrival for Indoor Positioning

A. Van Nieuwenhuyse, L. De Strycker, N. Stevens, J.-P. Goemaere and B. Nauwelaers (2013). *International Journal of Handheld Computing Research* (pp. 1-16).

www.irma-international.org/article/analysis-of-the-realistic-resolution-with-angle-of-arrival-for-indoor-positioning/79956

Multi-Agent Reinforcement Learning-Based Resource Management for V2X Communication

Nan Zhao, Jiaye Wang, Bo Jin, Ru Wang, Minghu Wu, Yu Liu and Lufeng Zheng (2023). *International Journal of Mobile Computing and Multimedia Communications* (pp. 1-17).

www.irma-international.org/article/multi-agent-reinforcement-learning-based-resource-management-for-v2x-communication/320190