Full Model-Driven Practice: From Requirements to Code Generation

Óscar Pastor and Sergio España

PROS Research Centre, Universitat Politècnica de València
{opastor,sergio.espana}@pros.upv.es

Abstract. A crucial success factor in information systems development is the alignment of the system with business goals, business semantics and business processes. Developers should be freed from programming concerns and be able to concentrate on these alignment problems. Model-driven system development (MDD) does not only provide a structured and systematic approach to systems development, but also offers developers the possibility of using model-transformation technologies to derive models of a lower abstraction level that can be further refined, and even generating software code automatically. This tutorial shows how to successfully integrate business process modelling (BPM), requirements engineering (RE) and object-oriented conceptual modelling with the objective of leveraging MDD capabilities. Participants work with state of the art modelling methods and code generation tools to explore different ways to match an information system with business requirements.

Keywords: model-driven development, requirements engineering, business process modelling, model transformation, model-driven architecture.

1 Introduction

Model-driven development (MDD) is, no doubt, an active area of research and innovation nowadays. Within the information systems research community, MDD methods for software development are being proposed. MDD first covered the (platform independent and platform specific) conceptual-modelling stage, paving the way to industrial tools that support modelling and code generation (e.g. Integranova [1]). Recently, (computation independent) model-driven requirements engineering (RE) academic approaches start to appear. Although, for the present, not many of them are mature enough to be applied under conditions of practice, it is a matter of time that many industrial methods and tools are available for model-driven RE.

2 Overview of the Tutorial

The tutorial presents the principles, concepts and common practices of MDD, introduces an MDD strategy covering the full cycle from requirements engineering to model compilation.

The goals of the tutorial are the following:

- To be able to elicit and specify the requirements of an information system, using MDD-suitable modelling languages.

J. Ralyté et al. (Eds.): CAiSE 2012, LNCS 7328, pp. 701–702, 2012.

- To learn to create the object-oriented conceptual model of the computerised information system, including the following abilities:
  - To systematically derive an initial platform-independent conceptual model from the computation-independent requirements model.
  - To complete the conceptual model in order to specify the software system considering both its static and dynamic aspects.
- To compile the conceptual model using state-of-the-art technology in order to automatically generate fully-functional source code.

The concepts and strategies are general enough to be able to apply them in many MDD scenarios. However, to operationalise them into a concrete approach, a specific method is applied: the integration of Communication Analysis and the OO-Method.

Communication Analysis is a business process modelling and RE method that proposes undertaking information system analysis from a communicational perspective [2]. As a highlighted feature, business process models are complemented with message structures that describe the new and meaningful information that is conveyed to the information system in each business activity (referred to as communicative events) [3].

Recently, Communication Analysis has been integrated into the OO-Method, an object-oriented MDD framework [4] that is UML-compliant. The OO-Method conceptual model consists of four complementary models that cover all the relevant aspects of the information system so as to allow for automatic code generation.

A model transformation strategy has been defined to derive, from Communication Analysis requirements models, initial versions of OO-Method conceptual models that can already be compiled to automatically generate software code [5,6]. This way, the MDD method that results from the bottom-up integration of CA and the OO Method covers the software development life-cycle from RE to code generation.

In short, this tutorial intends to provide insights on MDD that are useful to both researchers (so they can apply them in their proposals) and practitioners (so they are aware of what is coming and can anticipate the evolution of MDD technologies).

References

1. CARE Technologies. IntegranovaModel Execution System, `http://www.care-t.com`
2. España, S., González, A., Pastor, Ó.: Communication Analysis: A Requirements Engineering Method for Information Systems. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 530–545. Springer, Heidelberg (2009)
3. González, A., Ruiz, M., España, S., Pastor, Ó.: Message Structures: a modelling technique for information systems analysis and design. In: WER 2011 (2011), `http://arxiv.org/abs/1101.5341`
4. Pastor, O., Molina, J.C.: Model-Driven Architecture in practice: a software production environment based on conceptual modeling. Springer, New York (2007)
5. González, A., España, S., Ruiz, M., Pastor, Ó.: Systematic Derivation of Class Diagrams from Communication-Oriented Business Process Models. In: Halpin, T., Nurcan, S., Krogstie, J., Soffer, P., Proper, E., Schmidt, R., Bider, I. (eds.) BPMDS 2011 and EMMSAD 2011. LNBIP, vol. 81, pp. 246–260. Springer, Heidelberg (2011)
6. España, S.: Methodological integration of Communication Analysis into a Model-Driven software development framework. PhD at Universitat Politècnica de València, Spain (2011)