Chapter 75
Teaching Model-Driven Engineering in a Master's Program:
Three Editions on a PBL-Based Experience

Alexandre Bragança
Polytechnic Institute of Porto, Portugal

Isabel Azevedo
Polytechnic Institute of Porto, Portugal

Nuno Bettencourt
https://orcid.org/0000-0003-1767-8240
Polytechnic Institute of Porto, Portugal

ABSTRACT

Model-driven engineering (MDE) is an approach to software engineering that adopts models as the central artefact. Although the approach is promising in addressing major issues in software development, particularly in dealing with software complexity, and there are several success cases in the industry as well as growing interest in the research community, it seems that it has been hard to generalize its gains among software professionals. To address this issue, MDE must be taught at a higher-education level. This chapter presents a three-year experience in teaching MDE in a course of a master program in informatics engineering. The chapter provides details on how a project-based learning approach was adopted and evolved along three editions of the course. Results of a student survey are discussed and compared to those from another course. In addition, several other similar teaching experiences are analyzed.

DOI: 10.4018/978-1-6684-3702-5.ch075

INTRODUCTION

During their education, engineers learn about the relevant models in their areas and how to further apply them. One of the capabilities that students should acquire in programs that qualify for building systems, where software is a key and intense part, is "create and use models in system development" (Landwehr et al., 2017).

More intensive use of models has also been adopted for software engineering. Among them is Model-Driven Engineering (MDE), which promises several ways to address well-known problems (Somers, 2017), including software increasing complexity (Whittle, Hutchinson, & Rouncefield, 2014). Moreover, it is in line with the usual start of designing complex systems with some level of abstraction provided by models in traditional engineering disciplines.

In software product lines, substantial gains can be achieved, even for quality assurance, when the effort is put in the domain engineering instead of solely in the application engineering. In fact, MDE has been applied successfully in the industry but essentially in large corporations that can afford the inherent costs (Baker, Loh, & Weil, 2005; Burden, Heldal, & Whittle, 2014; Hossler, Born, & Saito, 2006).

However, it seems that MDE's advantages have been hard to generalize in a way that makes it available for the common developer (Haan, 2008). Also, companies that already design and use models dedicated to a particular domain may probably use MDE more than others that develop generic software (Whittle et al., 2014). Whittle et al. (2014) mentioned an organization that had to train hundreds of developers with difficulties in abstract thinking when MDE was adopted.

Multiple factors hinder organizations from embracing MDE, and its acceptance clearly requires technical changes, but also the overcoming of human attitudes when facing new techniques and the need to use new tools (Brambilla, Cabot, & Wimmer, 2012; Whittle et al., 2017). This aspect was highlighted in general some years ago (Glass, 2011) with the recognition that there is a learning curve with an initial low productivity that is acceptable when people realize the value of their adoption. In a Model-Driven Development (MDD) – which is essentially MDE focused on software development – survey, it has been found out that "in most cases, the use of the MDD in organisations depends only on the interest of people to use it" (Parviainen, Takalo, Teppola, & Tihinen, 2009) and people may only be appealed to use what they have heard about. Nevertheless, new competencies are needed, and their lack can compromise MDE appropriateness (Christensen & Ellingsen, 2016). Thus, pedagogical and training issues cannot be ignored (Goulão, Amaral, & Mernik, 2016).

In this context the authors share a three-year experience in teaching Model-Driven Engineering in a course of a master program in Informatics Engineering, a total of three editions of the course. Each one can be seen as action research (Lewin, 1946) iteration (see Figure 1) that aimed to analyse if it is possible to promote MDE subjects using a Project-Based Learning approach. This research question also reflects the desire to maintain the highest quality standards of the program, which is accredited by many international bodies. For instance, the American Board for Engineering and Technology (ABET) accreditation emphasizes the need of continuous improvement of programs. In fact, the reflection on the appropriateness of the provided learning experiences and pedagogical models has been institutionally reinforced. The continuous improvement of the course and, consequently, of the program has been the main motivation for this work. Our expectation is that the introduction of courses devoted to MDE may have the same impact on software development as Unified Modelling Language (UML) had in the past.