A Domain Specific Modeling Language for Enterprise Application Development

Bahman Zamani, MDSE Research Group, Dept. of Software Engineering, University of Isfahan, Isfahan, Iran
Shiva Rasoulzadeh, MDSE Research Group, Dept. of Software Engineering, University of Isfahan, Isfahan, Iran

ABSTRACT

This article describes how experience in domain specific modeling can be captured and abstracted in a domain specific modeling language (DSML). Modeling with a DSML results in quality models. Patterns of enterprise application architecture (PofEAA) is a rich set of patterns that can be used by designers when designing (modeling) web-based enterprise applications. This article aims at defining a DSML based on PofEAA patterns, as well as providing tool support for designing web-based enterprise applications that use these patterns. The authors have built a DSML using the profile extension mechanism of UML, by defining stereotypes. In addition to the proposed profile, this article has implemented the structure and behavior of PofEAA patterns in Rational Software Architecture (RSA) which is resulted in a tool that facilitates the design of software for designers. To show the usefulness of the tool, it is used for modeling two small systems based on the PofEAA patterns. The results show that many of the design is automated and the modeling speed is increased.

KEYWORDS

DesignPattern, DSML, UMLProfile, Web-Based Enterprise Applications

1. INTRODUCTION

Due to the complexity of modern software systems, it is de rigueur to build a model of the system prior to implementation. The Unified Modeling Language (UML) is a de facto standard for modeling software systems and is well-known for being a general-purpose modeling language (Booch, Jacobson, & Rumbaugh, 1998). Alongside the increasing variety in software projects comes the need for specialization within specific domains. Hence, in spite of numerous advantages of general purpose modeling languages, when the aim is to create a model for a specific domain, restrictions may arise for designers that reduce the speed of modeling (Fuentes-Fernandez & VALLECILO-MORENO, 2004). Software engineers have a special interest in utilizing domain-specific concepts when building their models (Selic, 2007). Models are the primary artifacts in model driven engineering (MDE). As such, the quality assessment of models is an important issue in MDE (Zamani & Butler, 2013). Additionally, the emergence of the MDE paradigm has increased interest in domain specific modeling languages (DSMLs). Since DSMLs are more expressive, solve complexity better, and render modeling simpler, they are becoming more common (Tolvanen, 2006). In this regard, in the general-purpose language of UML, a method is provided for extending this language to different domains, i.e., via the option of defining a “UML profile”. A UML profile consists of stereotypes, tagged values, and constraints. By using this option, one can extend UML to support the concepts of a particular domain (OMG, Unified Modeling Language Specification Version 2.5, 2015).

DOI: 10.4018/IJITSA.2018070104

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
In a bid to produce quality models, designers apply best practices that are introduced by the experts as patterns. “A pattern is a solution to a problem in a context” (Zamani & Butler, 2013). As a useful set of patterns, Martin Fowler’s Patterns of Enterprise Application Architecture (PofEAA) (Fowler, 2002) is used for modeling enterprise applications. When using patterns for designing a large software system, a developer is required to have significant experience working with patterns. The problem is that, the beginners and novice designers in this area may make mistakes in selecting the right pattern and in applying the pattern correctly, which will lead to low quality software. Therefore, it will be helpful to establish an approach that will make the process of designing a system using the relationships between patterns easier. One way of doing so will be to have a DSML for the patterns that assist the designer through the process of modeling the system, which is a DSML that supports selected patterns and the relationships between these patterns.

In this work, we considered PofEAA patterns as our domain concepts. We organize our defined DSML as a UML 2.0 Profile (OMG, Unified Modeling Language Specification Version 2.5, 2015). A UML 2.0 profile provides language constructs such as stereotypes and tagged values, which allow for attaching domain specific semantic, thus enabling us to customize UML for our domain and apply domain specific modeling (DSM). However, we discovered that simply defining a UML profile was not adequate when using patterns as concepts, e.g., we were unable to create a class, attribute, or relationship with constraints or tagged values, and could not express other behaviors with object constraint language (OCL) or tagged values. Similarly, in the case of using patterns of PofEAA as concepts of our domain, we need to define the behavior of these pattern in order to use them correctly. Due to the fact that Rational Software Architecture (RSA) has included GoF design patterns, and designers can use them for their modeling, hence, we try to add PofEAA patterns to this tool as well, to help designers in using them correctly. We have implemented PofEAA patterns in RSA; also we have defined a UML profile in RSA that presents PofEAA patterns concepts. Profile can be used by designers while modeling the system; it is also used for implementing patterns completely. As our case study, we show that how the proposed tool can be used in designing an online shopping and hotel management system.

To summarize, the contribution of this paper is as follows:

- A UML profile which is a DSML to present concepts of the Patterns of Enterprise Application Architecture (PofEAA).
- A plugin with pattern templates that demonstrates pattern’s structure and behavior and helps designer to design enterprise systems with PofEAA.

This paper is organized as follows. Section 2 introduces related work regarding different UML profiles and patterns. Section 3 introduces PofEAA patterns briefly and shows their whole architecture, and describes two patterns in terms of their structure and behavior. Section 4 describes our work and contains the definition of UML profile, pattern authoring process, and how to implement the behavior of patterns. Section 5 contains the case study in which we design online shopping and hotel management models with our tool. Section 6 concludes the paper.

2. RELATED WORK

There exist a plethora of research that is conducted on the DSMLs, UML profiles, and design patterns. Among them, in the following, we discuss the most related research that is performed in the area of profiles and patterns.

Rekhis et al. (Rekhis, Bouassida, & Bouaziz, 2010) argue that the management of the huge volume of data that each real-time application has to deal with is a problem since real-time databases must handle time-constrained data and time-constrained transactions. They have addressed this problem.
Reassessing Software Quality Performance: The Role of Knowledge Management
www.igi-global.com/article/reassessing-software-quality-performance/112066?camid=4v1a

Performance Implications of the Knowledge Chain
www.igi-global.com/article/performance-implications-knowledge-chain/2669?camid=4v1a