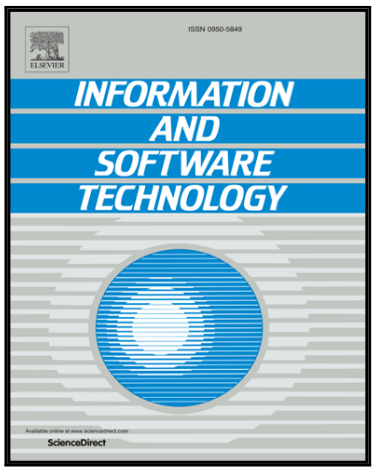

Reusable and Generic Design Decisions for Developing UML-based Domain-specific Languages

Bernhard Hoisl, Stefan Sobernig, Mark Strembeck

PII: S0950-5849(17)30453-6
DOI: 10.1016/j.infsof.2017.07.008
Reference: INFSOF 5851

To appear in: *Information and Software Technology*

Received date: 12 August 2016
Revised date: 12 July 2017
Accepted date: 13 July 2017

Please cite this article as: Bernhard Hoisl, Stefan Sobernig, Mark Strembeck, Reusable and Generic Design Decisions for Developing UML-based Domain-specific Languages, *Information and Software Technology* (2017), doi: 10.1016/j.infsof.2017.07.008

Reusable and Generic Design Decisions for Developing UML-based Domain-specific Languages

Bernhard Hoisl[a], Stefan Sobernig[a,*], Mark Strembeck[a,b,c]

[a]*Vienna University of Economics and Business (WU), Welthandelsplatz 1, 1020 Vienna, Austria*
[b]*Secure Business Austria (SBA) Research gGmbH, Favoritenstraße 16, 1040 Vienna, Austria*
[c]*Complexity Science Hub Vienna (CSH), Josefstädter Straße 39, 1080 Vienna, Austria*

Abstract

Context: In recent years, UML-based domain-specific model languages (DSMLs) have become a popular option in model-driven development projects. However, making informed design decisions for such DSMLs involves a large number of non-trivial and inter-related options. These options concern the language-model specification, UML extension techniques, concrete-syntax language design, and modeling-tool support.
Objective: In order to make the corresponding knowledge on design decisions reusable, proven design rationale from existing DSML projects must be collected, systematized, and documented using an agreed upon documentation format.
Method: We applied a sequential multi-method approach to identify and to document reusable design decisions for UML-based DSMLs. The approach included a Web-based survey with 80 participants. Moreover, 80 DSML projects[☆], which have been identified through a prior systematic literature review, were analyzed in detail in order to identify reusable design decisions for such DSMLs.
Results: We present insights on the current state of practice in documenting UML-based DSMLs (e.g., perceived barriers, documentation techniques, reuse potential) and a publicly available collection of reusable design decisions, including 35 decision options on different DSML development concerns (especially concerning the language model, concrete-syntax language design, and modeling tools). The reusable design decisions are documented using a structured documentation format (*decision record*).
Conclusion: Our results are both, scientifically relevant (e.g. for design-space analyses or for creating classification schemas for further research on UML-based DSML development) *and* important for actual software engineering projects (e.g. by providing best-practice guidelines and pointers to common pitfalls).

Keywords: model-driven software development, domain-specific language, design decision, design rationale, Unified Modeling Language, survey

1. Introduction

In model-driven development (MDD), a domain-specific modeling language (DSML) is a domain-specific language (DSL) for specifying design-level and platform-independent concerns in the target domain, rather than implementation-level concerns (see, e.g., [1]). In this context, DSMLs typically provide (but are not limited to) a graphical concrete syntax. A DSML is built on top of a tailored *abstract syntax* (i.e. the core language model) which is typically defined using metamodeling techniques. In addition to a DSML's abstract syntax (metamodel), DSML developers often use formal textual specification techniques to express the DSML's structural and behavioral semantics [2]. Once the abstract syntax and a corresponding concrete syntax are specified, a DSML is typically integrated into an MDD tool chain, such as the Eclipse Modeling Framework (EMF).

In recent years, the development of DSMLs based on the Unified Modeling Language (UML [3]) and/or on the Meta Object Facility (MOF [4]) has become a popular choice among software engineers: In a related survey, we found that more than 50% of the participating MDD researchers and practitioners have contributed to at least one UML-based DSML between 2000 and 2015 [5]. In addition to our own findings, the UML's relevance for DSML development is also reported in numerous other contributions (see, e.g., [6, 7, 8, 9, 10]). On the one hand, this momentum is due to a general trend towards the usage of DSLs in MDD [11]. On the other hand, the UML and the MOF provide native extension techniques for a) developing fully customized modeling languages (e.g., new diagram types) and b) for adapting the UML to domain-specific purposes while reusing UML features. Examples of such techniques include UML profiles [9, 12], pruning/reduction [13], metamodel slic-

[☆]Note that it is pure coincidence that there were 80 participants in the survey and that 80 DSML projects were reviewed.

[*]Corresponding author
Email addresses: bernhard.hoisl@wu.ac.at (Bernhard Hoisl), stefan.sobernig@wu.ac.at (Stefan Sobernig), mark.strembeck@wu.ac.at (Mark Strembeck)