Workshop Preview of the 15th Workshop on Domain Specific Modeling (DSM 2015)

Jeff Gray
University of Alabama
Box 870290
Tuscaloosa, AL 35487, USA
gray@cs.ua.edu

Jonathan Sprinkle
University of Arizona
1230 E. Speedway Blvd.,
Tucson, AZ, USA
sprinkle@ECE.Arizona.Edu

Juha-Pekka Tolvanen
MetaCase
Ylistönmäentie 31, FI-40500
Jyväskylä, Finland
jpt@metacase.com

Matti Rossi
Aalto University School of Economics
Runeberginkatu 22-24
FI-00101 Helsinki, Finland
matti.rossi@aalto.fi

Abstract

Domain-specific languages provide a viable and time-tested solution for continuing to raise the level of abstraction, and thus productivity, beyond coding, making systems development faster and easier. When accompanied with suitable automated modeling tools and generators it delivers to the promises of continuous delivery and devops. In domain-specific modeling (DSM) the models are constructed using concepts that represent things in the application domain, not concepts of a given programming language. The modeling language follows the domain abstractions and semantics, allowing developers to perceive themselves as working directly with domain concepts. Together with frameworks and platforms, DSM can automate a large portion of software production.
This paper introduces Domain-Specific Modeling and describes the SPLASH 2015 workshop, to be held on 27th of October in Pittsburgh, PA, which is the 15th anniversary of the event.

Categories and Subject Descriptors D 3.2 [**Languages**]: Specialized application languages, very high-level languages; D 2.2 [**Design Tools and Techniques**]: *Computer-aided software engineering* (CASE)

Keywords Modeling Languages; Metamodeling; Domain-Specific Languages; Code Generation

1. Introduction

Domain-Specific Modeling (DSM) provides a modern solution to demands for higher productivity by constricting the gap between problem and solution modeling. In the past, productivity gains have been sought from new programming languages. Domain-specific modeling languages provide a viable solution for continuing to raise the level of abstraction beyond coding, making development faster and easier.

SPLASH Companion'15, October 25–30, 2015, Pittsburgh, PA, USA
ACM. 978-1-4503-3722-9/15/10
http://dx.doi.org/10.1145/2814189.2833204

In DSM, the models are constructed using concepts that represent things in the application domain, not concepts of a given programming language. The modeling language follows the domain abstractions and semantics, allowing developers to perceive themselves as working directly with domain concepts. The models represent simultaneously the design, implementation and documentation of the system. In a number of cases, the final products can be automatically generated from these high-level specifications with domain-specific code generators. This automation is possible because of domain-specificity: both the modeling language and code generators fit to the requirements of a narrowly defined domain, usually inside one organization only.

Metamodeling and metamodel-based tools significantly ease the support and production of domain-specific development environments. They provide support for experimentation with the language as it is built, and remove the burden of tool creation and maintenance from the language creator.

2. Defining and using DSMLs

Three things are necessary to achieve full automatic code generation from domain modeling: firstly, a modeling tool supporting a domain-specific modeling language; secondly, a code generator; and lastly, a domain-specific framework. Figure 1 shows these three elements at two levels: the definition level and the use level.

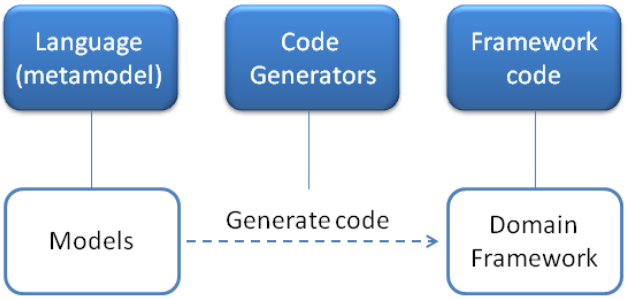

Figure 1. Framework for domain-specific modeling

The top-level (representing the definition) is made once by the organization for a given domain. Normally, one or two experts will define the modeling language (i.e., a

metamodel) and related code generation, normally with a metamodeling tool. The metamodel is the implementation of the DSML, and includes the concepts and rules directly from the domain. Developers will often create the framework code in earlier projects in the domain, with some being added or modified specifically for the DSM creation project.

The bottom-level process represents the use of a DSML and code generator. This level is performed many times, once for each product, by developers. Reusing parts of the model that are common to several products can often further reduce development time. The code generation and use of a domain framework or platform services require no effort by the developer. Together, these savings form the primary payback of the DSM approach amortized over each use.

In DSM, the specification models are built from instances of the domain concepts. The code generator walks through the model and transforms the concept structures into code. In some cases, the code will be fully self-contained; more often, significant parts of the code will be calls to reusable components and the domain framework. Because the code is generated, syntax and logic errors do not normally occur, and the resultant improvement in quality forms a significant secondary payback of the DSM approach [1].

3. Workshop Focus and Topics

The goals of the workshop are to share experiences and demonstrate the DSM solutions that have been developed by both researchers and practitioners, identify research questions and continuing to build the community. We aim to investigate in detail the DSM solutions that have been investigated by members in the modeling community. We received 17 papers, of which we accepted 12. The DSM workshop is one of longest running series of workshops at SPLASH/OOPSLA, this being the 15th anniversary of the series and we plan a little celebration the workshop.

This workshop will provide ample time for paper presentations, language demonstrations and interactive workgroups. Traditionally there have been lively discussions on different approaches and tools. The workshop also aims to collect and exchange experiences related to building and using DSMs; continue building and extending the DSM community; and address in workgroups the issues raised at previous workshops, which triggered interest that led to publication of three special journal issues on DSM topics and a book on DSM [2]. The results of the group work sessions, along with presentation slides, will be made available on the workshop website [3] together with the papers. The topics addressed in the workshop include:

- Industry/academic experience reports describing success/failure in implementing and using DSM
- Novel features in language workbenches/DSM tools
- Approaches to implement metamodel-based languages
- Metamodeling frameworks and languages
- Modularization technologies for DSM
- Novel approaches for code generation from DSM
- Issues of support/maintenance for DSM-based systems
- Evolution of languages along with their domain
- Organizational and process issues in DSM
- Demonstrations of working, or in-progress, DSM solutions (languages, generators, frameworks, tools)

References

[1] Gray, J., Tolvanen, J.-P., Kelly, S. Gokhale, A., Neema, S., and Sprinkle, J,, "Domain-Specific Modeling," *CRC Handbook on Dynamic System Modeling*, (Paul Fishwick, ed.), CRC Press, 2007.

[2] Kelly, S., and Tolvanen, J-P, *Domain-Specific Modeling*, Wiley, 2008.

[3] Workshop on Domain-Specific Modeling (DSM'15), http://www.dsmforum.org/events/DSM