Knowledge Industry Survival Strategy (KISS)

Fundamental Principles and Interoperability Requirements for Domain Specific Modeling Languages

Jorn Bettin
Sofismo
Switzerland
jbe@sofismo.ch

William Cook
University of Texas at Austin
United States
wcook@cs.utexas.edu

Tony Clark
Thames Valley University
United Kingdom
tony.clark@tvu.ac.uk

Steven Kelly
MetaCase
Finland
stevek@metacase.com

Abstract

Domain Specific Languages are raising the level of abstraction of software specifications and of knowledge representation in general. When DSLs are used to formalize the results of domain analysis, the result is a clean separation of concerns in the problem space. This is a major advance over aspect oriented programming, where separation of concerns is only achieved in the solution space. However, the level of interoperability between current DSL tools is comparable to the level of interoperability between CASE tools in the 90s. To increase the popularity of DSL based approaches, this needs to change. Software development has become highly decentralized, and an assumption that all parties in a global software supply chain will use identical tooling is simply not realistic. As a result today's software supply chains are much less automated than supply chains in other, more mature industries. The KISS series of workshops is used to incrementally establish a consensus on the fundamental principles that underpin the use of DSLs, and to improve DSL tool interoperability.

Categories and Subject Descriptors D.2.11 [**Software Engineering**]: Software Architectures – data abstraction, domain-specific architectures, languages (e.g., description, interconnection, definition).

General Terms Documentation, Design, Economics, Reliability, Human Factors, Standardization, Languages, Verification.

Keywords Domain specific language; interoperability; modularity; modeling language design; model driven software.

1. Objectives

Meeting DSL interoperability challenges requires a significant increase in the use of formal specifications in the wider context of developing software intensive systems, and advanced tooling for creating, managing, and exploiting formal specifications. KISS [1] [2] is an industry led initiative with two goals:

1. Reaching a strong consensus on fundamental values and principles for designing and using DSLs
   - The substrate on top of which practical interoperability can be achieved
2. Progress towards interoperability between tools
   - Not aiming too high, but instead achieving practical results

2. Results Achieved To Date

Building software intensive systems has taught us that domain specific languages represent the key to capturing, preserving, and exploiting knowledge in virtually all industries. Spurious complexity is our biggest enemy. Hence, when designing software, we keep it sweet & simple (KISS). Most importantly, we are committed to the following values, fundamental principles, and guidelines:

2.1 Values

1. We strive to automate software construction from domain models; therefore we consciously distinguish between building software factories and building software applications

OOPSLA 2009 October 25-29, 2009, Orlando, Florida, USA.
ACM 978-1-60558-768-4/09/10.

2. We work with domain-specific assets, which can be anything from models, components, frameworks, generators, to languages and techniques

3. We support the emergence of supply chains for software services, which implies domain-specific specialization and enables mass customization

4. We see Open standards, as well as Open Source test beds and reference implementations as driving interoperability

5. We use methodologies that conform with the values of the Agile Manifesto

2.2 Fundamental principles for designing domain specific languages

1. There must be an economic imperative for the development of a DSL

2. The DSL must be meaningful to users of the DSL

3. The DSL must be appropriate for the intended processing

4. The DSL definition must always be available when processing a model

5. The DSL should be cognitively efficient for the users of the DSL

6. The DSL should have multiple notations where necessary

7. DSLs should be kept small through modularization and integration

8. DSLs should offer mechanisms for modularizing and integrating models

9. The DSL should be supported by appropriate tooling for DSL users

2.3 Guidelines

1. DSL development must assemble sufficient expertise in:

   - Problem domain; solution domain; designing DSLs; language implementation technologies

2. DSL development must anticipate evolution of the DSL

3. DSL modularization should take into account DSL user roles

4. DSL design and validation should use an iterative approach

5. DSL design should involve the construction of concrete examples

6. DSL notations should take into account established domain notations

— Peter Bell, Jorn Bettin, Tony Clark, Keith Duddy, Scott Finnie, Matthew Fowler, Steven Kelly, Jack Kennedy, Frank Sauer, Laurence Tratt, Markus Voelter, Jos Warmer

These results reflect the 2nd KISS workshop on 16 June 2009 at the Code Generation conference [3]. Material related to the fundamental principles was first presented at the 1st KISS workshop on 14 April 2009 at the ASWEC conference [4]. The core values were originally articulated at a birds-of-a-feather session at OOPSLA 2003.

The specific goal for the OOPSLA workshop is the consolidation of KISS results achieved in 2009, and publication of these results in an Industrialized Software Manifesto.

3. Format and Activities

The workshop will be highly interactive, making use of the Open Space Technology format.

Jorn Bettin will introduce the workshop with an assessment of the readiness of software organizations to adopt the use of domain specific languages as part of model driven product line approaches. The talk will underscore the need for interoperability between domain specific language design/implementation tools.

4. Publication of Results

The organizers will consider the best papers for publication in the theme issue of the SoSyM journal on Model Based Interoperability [5] (submission deadline 1 November 2009).

References

[1] KISS signatories "Knowledge Industry Survival Strategy" (2009), www.industrialized-software.org/kiss-initiative.

[2] Tony Clark, Jorn Bettin "Knowledge Industry Survival Strategy" (2009), itcentre.tvu.ac.uk/~clark/Papers/KISS.pdf.

[3] KISS signatories "KISS Workshop at Code Generation 2009, Cambridge, United Kingdom", www.industrialized-software.org/kiss-cg-2009.

[4] KISS signatories "KISS Workshop at ASWEC 2009, Gold Coast, Australia", www.industrialized-software.org/kiss-aswec-2009.

[5] Tony Clark, Jorn Bettin (ed.) "Software and Systems Modeling, Theme Section on Model Based Interoperability" (2009), http://sosym.se-rwth.de/theme_issues/cfp-MBI.pdf.