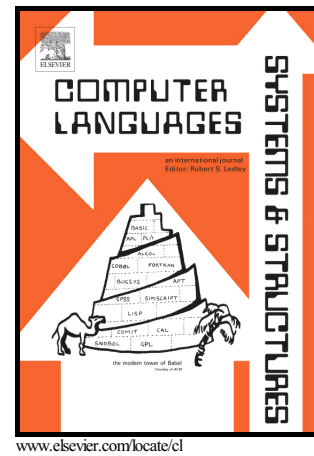


Author's Accepted Manuscript

Safe model polymorphism for flexible modeling

Thomas Degueule, Benoit Combemale, Arnaud Blouin, Olivier Barais, Jean-Marc Jézéquel



PII: S1477-8424(16)30026-4
DOI: <http://dx.doi.org/10.1016/j.cl.2016.09.001>
Reference: COMLAN235

To appear in: *Computer Language*

Received date: 9 February 2016
Revised date: 5 September 2016
Accepted date: 11 September 2016

Cite this article as: Thomas Degueule, Benoit Combemale, Arnaud Blouin Olivier Barais and Jean-Marc Jézéquel, Safe model polymorphism for flexible modeling, *Computer Language*, <http://dx.doi.org/10.1016/j.cl.2016.09.001>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting galley proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain

Safe Model Polymorphism for Flexible Modeling

Thomas Degueule^a, Benoit Combemale^a, Arnaud Blouin^b, Olivier Barais^c,
Jean-Marc Jzquel^c

^a*Inria, France*

^b*INSA Rennes, France*

^c*University of Rennes 1, France*

Abstract

Domain-Specific Languages (DSLs) are increasingly used by domain experts to handle various concerns in systems and software development. To support this trend, the Model-Driven Engineering (MDE) community has developed advanced techniques for designing new DSLs. However, the widespread use of independently developed, and constantly evolving DSLs is hampered by the rigidity imposed to the language users by the DSLs and their tooling, e.g., for manipulating a model through various similar DSLs or successive versions of a given DSL. In this paper, we propose a disciplined approach that leverages type groups' polymorphism to provide an advanced type system for manipulating models, in a polymorphic way, through different DSL interfaces. A DSL interface, aka. *model type*, specifies a set of features, or services, available on the model it types, and subtyping relations among these model types define the safe substitutions. This type system complements the Melange language workbench and is seamlessly integrated into the Eclipse Modeling Framework (EMF), hence providing structural interoperability and compatibility of models between EMF-based tools. We illustrate the validity and practicability of our approach by bridging safe interoperability between different semantic and syntactic variation points of a finite-state machine (FSM) language, as well as between successive versions of the Unified Modeling Language (UML).

Keywords: Metamodeling, Model typing, Type groups polymorphism

1. Introduction

Extending the time-honored practice of separation of concerns, Domain-Specific (*modeling*) Languages (DSLs) are increasingly used to handle different, complex problem or solution concerns in software-intensive system development [43]. In particular, problem-based DSLs target experts who focus on building parts of software or systems that address specific concerns (e.g., security policies or variability management). These DSLs help narrow the gap between problem and implementation concepts by providing experts with problem-level abstractions and tools to transform these abstractions into executable artifacts.

Download English Version:

<https://daneshyari.com/en/article/4949431>

Download Persian Version:

<https://daneshyari.com/article/4949431>

[Daneshyari.com](https://daneshyari.com)