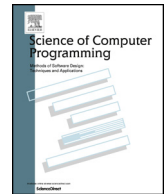




ELSEVIER

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico


A survey and comparison of transformation tools based on the transformation tool contest



Edgar Jakumeit^{*}, Sebastian Buchwald^a, Dennis Wagelaar^b, Li Dan^c,
 Ábel Hegedüs^d, Markus Herrmannsdörfer^e, Tassilo Horn^f, Elina Kalnina^g,
 Christian Krause^h, Kevin Lanoⁱ, Markus Lepper, Arend Rensink^j, Louis Rose^k,
 Sebastian Wätzoldt^h, Steffen Mazanek

HIGHLIGHTS

- Many state-of-the-art graph rewriting and model transformation tools competed at the Transformation Tool Contest 2011.
- We give the most encompassing survey of transformation tools so far, based on an illustrative Hello World case.
- We compare the tools and their languages in detail, based on an elaborate taxonomy.
- Researchers gain an overview of the field, prospective users get help in choosing in between the tools.
- All tools can be tested online with a SHARE virtual machine.

ARTICLE INFO

Article history:

Received 24 March 2012

Received in revised form 12 October 2013

Accepted 23 October 2013

Available online 8 November 2013

Keywords:

Graph rewriting

Model transformation

Tool survey

Transformation tool contest

ABSTRACT

Model transformation is one of the key tasks in model-driven engineering and relies on the efficient matching and modification of graph-based data structures; its sibling graph rewriting has been used to successfully model problems in a variety of domains. Over the last years, a wide range of graph and model transformation tools have been developed – all of them with their own particular strengths and typical application domains. In this paper, we give a survey and a comparison of the model and graph transformation tools that participated at the Transformation Tool Contest 2011. The reader gains an overview of the field and its tools, based on the illustrative solutions submitted to a Hello World task, and a comparison alongside a detailed taxonomy. The article is of interest to researchers in the field of model and graph transformation, as well as to software engineers with a transformation task at hand who have to choose a tool fitting to their needs. All solutions referenced in this article provide a SHARE demo. It supported the peer-review process for the contest, and now allows the reader to test the tools online.

© 2013 Elsevier B.V. All rights reserved.

* Corresponding author.

E-mail addresses: eja@ipd.info.uni-karlsruhe.de (E. Jakumeit), post@markuslepper.eu (M. Lepper), steffen.mazanek@gmail.com (S. Mazanek).

^a Karlsruher Institut für Technologie, Germany.

^b Vrije Universiteit Brussel, Brussels, Belgium.

^c Faculty of Science and Technology, University of Macau, China.

^d Budapest University of Technology and Economics, Hungary.

^e Institut für Informatik, Technische Universität München, Germany.

^f Institute for Software Technology, University Koblenz-Landau, Germany.

^g Institute of Mathematics and Computer Science, University of Latvia, Latvia.

^h Hasso-Plattner-Institut, Universität Potsdam, Germany.

ⁱ Dept. of Informatics, King's College London, Strand, London, UK.

^j Department of Computer Science, University of Twente, The Netherlands.

^k Department of Computer Science, University of York, UK.

1. Introduction

The work in this article is part of the special issue on Experimental Software Engineering in the Cloud of the journal Science of Computer Programming, which collates articles that compare software engineering tools in a pragmatic manner, for example by having tool developers solve a shared case study. The focus of this special issue is on reproducibility and accessibility, to be satisfied with virtual machine demos.

1.1. Background

The Transformation Tool Contest [98] held 2011 in Zurich, Switzerland, was a research workshop where providers of various transformation tools from the worlds of model transformation and graph rewriting gathered to compare their tools alongside a number of selected case studies. At this workshop, SHARE [25] virtual machines were employed to support the peer-review process.

The compared solutions were implemented with complex software toolkits, often requiring a substantial amount of time and knowledge to install and set up. The SHARE cloud allowed all participants to start up the pre-installed software environment of their competitors including their solutions with just one click, drastically lowering the barrier for comparison and evaluation.

While being meant originally for peer-based evaluation, the SHARE images of this scientific workshop were kept available for further empirical evaluation by prospective tool users from industry, who can explore the effect of inputs and parameters better suited to their needs.

While held as a scientific workshop for comparing transformation tools alongside common case studies, one of the case studies consisting of small and illustrative tasks resulted in instructive solutions that are simple enough to be understood with limited knowledge and in a very limited amount of time. Based on those solutions, an introduction into the tools and their approaches can be given.

We take the chance offered by those facts and summarize the results and findings of the Transformation Tool Contest 2011 in the following not only for the tool building community, but also for a prospective user on the search for the right graph or model transformation tool to employ.

1.2. Motivation

We aim at assisting software engineers that are facing a specific model transformation or graph rewriting problem in choosing a tool that is well-suited to their needs. This statement gives rise to the questions: What is a *model transformation* or a *graph rewrite problem*? And: What does *well-suited to someone's needs* mean? In Appendix A we investigate these questions and their answers in detail, together with a discussion of the advantages and disadvantages of using those tools; here, we give a brief summary.

In a nutshell, we are confronted with a graph rewriting or model transformation problem if the most adequate representation of the data at hand is a mesh of objects, and we need to change its structure or map it to another mesh of objects.

The transformation tools introduced in this article offer *faster* development of solutions compared to manual coding in a traditional programming language, by *reusing* existing functionality. The transformation languages offered by the tools are typically of higher *expressiveness* than general-purpose programming languages for tasks of the transformation domain, leading to more *concise* solutions, thus *lowering* maintenance *costs*. They are *declarative*, i.e., they allow you to concentrate on the *what* instead of on the *how*. Often, these tools offer a *visual* style of specification and debugging (or simulation).

1.3. Survey and comparison

The problem of *reuse* is: Are the available features really the ones that are needed? Using a caliper when a hammer is needed would not offer the expected benefits. You must *choose* according to *your needs*.

In order to assist you in choosing a promising tool we present different kinds of information in the following sections, at an increasing level of detail. They allow to incrementally reduce the large set of potential candidates to a small set of the most promising candidates, which can then be evaluated in depth, based on their article in the proceedings of the workshop [98], but especially by having a look at their SHARE images.

Please note that our focus is on assisting you in *choosing* a tool, under the assumption that model transformation or graph rewriting are not foreign words to you. For a tutorial *introduction* into the field please have a look at Graph Transformation in a Nutshell [28] and Model-Driven Software Engineering in Practice [9] or MDA Explained – The Model Driven Architecture: Practice and Promise [54].

Hello World. We start with a description of the Hello World case [69], which was posed at the Transformation Tool Contest 2011 [98]. The Hello World case is a mini-benchmark consisting of several prototypical tasks of the domain (and is thus a lot more revealing than the Hello World program from traditional programming [53], which only requires to print a greeting message to the console). Those tasks are *simple* yet highly *illustrative*. The tasks as well as the solutions can be read and

Download English Version:

<https://daneshyari.com/en/article/433359>

Download Persian Version:

<https://daneshyari.com/article/433359>

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