Science of Computer Programming 85 (2014) 5–40

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Evaluation of model transformation approaches for model refactoring

S. Kolahdouz-Rahimi, K. Lano *, S. Pillay, J. Troya, P. Van Gorp

King's College London, Strand, London, WC2R 2LS, United Kingdom

HIGHLIGHTS

- A measurement-based comparison of leading model transformation approaches on a challenging transformation case study.
- Provides a rigorous method for comparative evaluation of transformation approaches, based on quality characteristics and empirical measurement.
- A wide range of quality characteristics are evaluated, from correctness to complexity, modularity, usability and portability.

ARTICLE INFO

Article history:
Received 24 March 2012
Received in revised form 20 March 2013
Accepted 30 July 2013
Available online 13 August 2013

Keywords:
Model transformation
Measurement
Quality characteristics
Model restructuring

ABSTRACT

This paper provides a systematic evaluation framework for comparing model transformation approaches, based upon the ISO/IEC 9126-1 quality characteristics for software systems. We apply this framework to compare five transformation approaches (QVT-R, ATL, Kermeta, UML-RSDS and GrGen.NET) on a complex model refactoring case study: the amalgamation of apparent attribute clones in a class diagram.

The case study highlights the problems with the specification and design of the refactoring category of model transformations, and provides a challenging example by which model transformation languages and approaches can be compared. We take into account a wide range of evaluation criteria aspects such as correctness, efficiency, flexibility, interoperability, re-usability and robustness, which have not been comprehensively covered by other comparative surveys of transformation approaches.

The results show clear distinctions between the capabilities and suitabilities of different approaches to address the refactoring form of transformation problem.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Model transformations are an essential part of model-driven engineering approaches to software development. Transformations are used to refine models from platform-independent forms to platform-specific, to migrate models in response to metamodel evolution, and generally to translate the semantic content of a model from one language to that of another. Such transformations usually have distinct source and target models and are *exogenous*, with distinct source and target languages [10]. Transformations can also be used to restructure and refactor models, in order to improve the quality of models or to make them conform to standards. Such transformations are often update-in-place, operating on a single model, and are *endogenous*, with the same source and target language. A case study of this kind is used in this paper.

As transformations have become more widely applied, they have also become large and complex software systems in their own right, to which model-driven engineering can be applied [16,25]. A range of model transformation approaches (considered as combinations of a transformation language and a tool for the language) have been developed.

* Corresponding author.
E-mail address: kevin.lano@kcl.ac.uk (K. Lano).

0167-6423/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.scico.2013.07.013

Some approaches emphasise declarative specification, either logic-based or graph-theory based, others are imperative in nature, and others combine declarative and imperative aspects (hybrid approaches).

Factors such as the interoperability of transformation approaches, the efficiency of transformations implemented by means of particular approaches, and the maintainability of transformations specified in particular approaches, have become important factors in selecting an approach for a given transformation problem category.

Therefore, a suitable broad-based evaluation framework is needed to compare and assess the benefits and disadvantages of particular transformation approaches for specific categories of transformation problems. In this paper we introduce an evaluation framework based upon the software quality characteristics defined in the ISO/IEC 9126-1 and 25010 standards [7,18,19]. For each such characteristic, we use its subcharacteristics defined by [18] as *external* measures in the sense of the Goal-Question-Metric paradigm [6], and these measures will in turn be evaluated based upon quantitative *internal* measures or attributes of the transformation specification language and transformation implementation tool of each approach. For example, a subcharacteristic of functionality is *suitability*, which has quantitative measures including size, complexity, effectiveness and development effort.

As an example of applying this framework, we compare five established model transformation approaches from different language categories (GrGen: graph transformation [20], Kermeta: imperative [11], QVT-R: declarative [36], ATL: hybrid [21], UML-RSDS: general purpose MDE tool [27]) upon a transformation problem which is typical of model refactoring transformations.

Our comparison is based on the characteristics of *Functionality*, *Reliability*, *Usability*, *Efficiency*, *Maintainability* and *Portability* from ISO/IEC 9126-1. By systematically comparing and evaluating the selected transformation approaches on the case study, according to the ISO/IEC 9126-1 quality model, we can provide clear guidelines for the appropriateness of different types of transformation approaches for refactoring transformations, and the specific advantages and disadvantages of particular approaches for this type of transformation problem.

Section 2 places this research in the context of previous surveys of transformation approaches. Section 3 defines the transformation case study in detail. Section 4 defines the evaluation framework, and presents test cases. Sections 5, 6, 7, 8, 9 present the individual solutions to the problem. Section 10 compares the different solutions on the relative values of their characteristics. Section 11 gives conclusions.

2. Related work

There have been a number of publications comparing model transformation approaches on different case studies. This previous research can be divided into (i) work defining classifications for transformation approaches; (ii) work comparing approaches using subjective measures; (iii) work comparing approaches using objective measures.

A classification of model transformation approaches based on features is given in [10], which defines a general terminology for describing model transformation approaches. They cover a broad range of classification factors and present two examples of transformation.

Mens and Van Gorp in [31] applied a multi-dimensional taxonomy to categorise tools, techniques or formalisms for model transformation based on their common qualities. A number of functional and non-functional requirements for model transformations are specified.

Taentzer et al. [43] generates a taxonomy for graph transformation tools by focusing on AGG, AToM, VIATRA2 and VMTS, using the commonly-used example of transforming from class diagrams to relational databases, but without considering any quality attributes. Mohagheghi and Dehlen [34] provided an initial framework for defining and evaluating quality across different categories of model-driven engineering. This study also addressed the adaptation of the quality framework to model transformations.

In [32], quality requirements are formulated for graph transformation tools and these are analysed without focusing on a specific case study. The paper [41] also highlights several desirable features of model transformation approaches without focusing on any specific case.

In [39], Rose et al. describe the result of a migration case study at the Transformation Tool Contest 2010 workshop, with nine graph and model transformation tools applied to a model migration problem (the mapping of UML state machines to activity diagrams). All solution experts perform a peer review of the other solutions and the results are analysed statistically. Afterwards, the statistics are investigated critically by experts. The paper considers correctness and tool maturity as the most important evaluation criteria; however, these are evaluated subjectively. Their comparison is based on participant opinions and not on empirical evidence. Furthermore, important characteristics such as efficiency, complexity and modularity are not considered in this research.

A further paper based on subjective evaluations is [45], which shows the result of an earlier Transformation Tool Contest in 2007. 11 participants using graph-based tools attended the contest to perform a transformation from UML activity diagrams to formal CSP processes.

Our paper described in [28] evaluates Kermeta, Viatra, QVT-R, UML-RSDS and ATL on three case studies, a re-expression, a refinement, and the first rule of the case study considered in this paper. Subjective evaluations are given on the appropriateness of the approaches for each category of transformation.

In [1,2] some key factors that influence the internal quality of model transformations are presented and assessed by a specialised set of metrics on size, functionality, modularity and consistency.

Download English Version:

https://daneshyari.com/en/article/433358

Download Persian Version:

https://daneshyari.com/article/433358

Daneshyari.com