

First International Workshop on Model-Driven Software Migration (MDSM 2011)

March 1, 2011 in Oldenburg, Germany

Satellite Event of IEEE CSMR 2011
March 1-4, 2011

<http://mdsm2011.uni-koblenz.de>

Wilhelm Hasselbring
Software Engineering Group
University of Kiel
Kiel, Germany
wha@informatik.uni-kiel.de

Andreas Fuhr, Volker Riediger
Institute for Software Technology
University of Koblenz-Landau
Koblenz, Germany
afuhr|riediger@uni-koblenz.de

Abstract—The First International Workshop on Model-Driven Software Migration brings together researchers and practitioners in the area of model-driven approaches supporting software migration. State-of-the-art techniques as well as real-world experiences are presented and discussed to stimulate further model-driven migration research activities.

I. BACKGROUND

Model-driven Software Development (MDSM) and Software Migration are two different approaches that had been under research separately. In recent years, researchers found interesting analogies between both fields.

In software engineering, one of the key principles is abstraction, that is, focusing only on the important aspects while fading-out details [1]. Model-driven software development (MDSM) aims at modeling these important aspects at different levels of abstraction. This allows to design software, starting with the "big picture" (abstract level) and approach more concrete levels by adding more details to the models until the system is implemented (concrete level).

Software migration aims at converting an old system (legacy system) into a new technology without changing functionality [2]. This implies understanding, how the legacy system is working. For this purpose, legacy code must be leveraged into a higher level of abstraction in order to focus only on the important aspects.

At this point, model-driven software development and software migration meet. Migration projects can benefit from the vision of MDSM by abstracting legacy systems (reverse engineering), transform them and implement the migrated system (forward engineering).

However, both fields of research are not yet entirely understood. Neither is the combination of both fields examined very well. The MDSM workshop brings together latest research in the field of model-driven software migration approaches.

II. GOALS AND TOPICS

The MDSM workshop brings together researchers and practitioners in the area of model-driven approaches supporting software migration to present and discuss state-of-the-art techniques as well as real-world experiences to stimulate further model-driven migration research activities.

The scope of the MDSM workshop includes, but is not restricted to, the following topics:

- Modeling languages, query languages and transformation languages
- Domain Specific Languages for software migration
- Model-integration in repositories
- Model-driven architecture reconstruction or migration
- Model-driven code migration
- Software migration by transforming legacy code
- Model-driven software renovation
- Tools and methods for model-driven migration
- Design patterns for model-driven software migration
- Experience reports

III. PROGRAM

The MDSM workshop is held during the CSMR 2011 main conference on March 1, 2011. The full-day workshop consist of three thematically grouped sessions:

- one 90 minutes project session, presenting latest research projects in the field of model-driven software migration
- two 90 minutes paper sessions containing paper presentations with plenty of time for discussions.

IV. ACCEPTED PAPERS

The following four of seven submissions have been accepted by the program committee for presentation at the workshop. The proceedings of the workshop are published on CEUR (<http://ceur-ws.org>).

Bartolomei, T., Derakhshanmanesh, M., Fuhr, A., Koch, P., Konrath, M., Lämmel, R., and Winnebeck, H., “An integrated repository for API migration”: We demonstrate that the integration of different dimensions of knowledge about APIs may be usefully leveraged in API migration. Our demonstration targets the two major GUI APIs for the Java platform, and wrapper-based re-implementations that cover both directions of migrating between them. Based on our integrated repository, we assess certain API migration-related qualities of the wrappers (e.g., completeness); we also provide guidance for API migration (e.g., in improving the existing wrappers). Our repository integrates *i*) basic facts about the structural properties and internals of genuine and wrapper-based API implementations, *ii*) the usage of the APIs in open-source projects on SourceForge, and *iii*) ontology-based links between types and methods of different APIs.

Bode, S., Lehnert, S., and Riebisch, M., “Comprehensive Model Integration for Dependency Identification with EMFTrace”: As model-based software development becomes increasingly important, the number of models to express various aspects of software at different levels of abstraction raises. Meanwhile evolutionary development and continuous changes demand for explicit dependencies between involved models to facilitate change impact analysis, software comprehension, or coverage and consistency checks. However, there are no comprehensive approaches supporting models, dependencies, changes, and related information throughout the entire software development process. The approach presented in this paper provides a unified and model-spanning concept with a repository for model integration, model versioning, and dependency identification utilizing traceability techniques, enhanced with analytic capabilities. The identification is based on a rule set to provide high values for precision and recall. The approach is implemented in a tool called EMFTrace, which is based on Eclipse technology and supports different CASE tools for modeling.

Heidenreich, F., Johannes, J., Reimann, J., Seifert, M., Wende, C., Werner, C., Wilke, C., and Assmann, U., “Model-driven Modernisation of Java Programs with JaMoPP”: The history of all programming languages exposes the introduction of new language features. In the case of Java—a widespread general purpose language—multiple language extensions were applied over the last years and new ones are planned for the future. Often, such language extensions provide means to replace complex constructs with more compact ones. To benefit from new language extensions for large bodies of existing source code, a technique is required that performs the modernisation of existing source code automatically. In this paper we demonstrate, how Java programs can be automatically migrated to new versions of the Java language. Using JaMoPP, a tool that can create models from Java source code, we enable the application of model transformations to perform model-driven modernisation of Java programs. Our approach is evaluated by applying two concrete trans-

formations to large open source projects. First, we migrate classical for loops to the new for-each style (introduced in Java 5). Second, we convert anonymous classes to closures (planned for Java 8). Furthermore, we discuss how tracing transformations allows to quantify the impact of planned extensions.

Oldevik, J., K. Olsen, G., Brønner, U., and Bodsberg, N. R., “Model-Driven Migration of Scientific Legacy Systems to Service-Oriented Architectures”: We propose a model-driven and generative approach to specify and generate web services for migrating scientific legacy systems to service-oriented platforms. From a model specification of the system migration, we use code generation to generate web services and automate the legacy integration. We use a case study from an existing oil spill analysis application developed in Fortran and C++ to show the feasibility of the approach.

V. ORGANIZERS

A. Workshop Chairs

- Wilhelm Hasselbring, Christian-Albrechts-Universität zu Kiel, Germany
- Andreas Fuhr, Universität Koblenz-Landau, Germany
- Volker Riediger, Universität Koblenz-Landau, Germany

B. Program Committee

- Andy Schürr, Technische Universität Darmstadt, Germany
- Anthony Cleve, Institut National de Recherche en Informatique et en Automatique (INRIA) Lille, France
- Bernhard Rumpe, Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen, Germany
- Dragang Gasevic, Athabasca University, Canada
- Eleni Stroulia, University of Alberta, Canada
- Filippo Ricca, Università degli Studi di Genova, Italy
- Harry Sneed, Central European University Budapest, Hungary & Universität Passau, Universität Regensburg, Germany
- Heinz Züllighoven, Universität Hamburg, Germany
- Jaques Klein, Université du Luxembourg
- Jorge Ressa, Universität Bern, Switzerland
- Jürgen Ebert, Universität Koblenz-Landau, Germany
- Klaus Krogmann, Karlsruhe Institute of Technology, Germany
- Rainer Gimnich, IBM Frankfurt, Germany
- Rocco Oliveto, Università degli Studi di Salerno, Italy
- Romain Robbes, Universidad de Chile, Chile
- Steffen Becker, Universität Paderborn, Germany
- Tudor Girba, Universität Bern, Switzerland

REFERENCES

- [1] S. Beydeda, M. Book, and V. Gruhn, *Model-driven software development: With 5 tables*. Berlin: Springer, 2005.
- [2] Harry M. Sneed, Ellen Wolf, and Heidi Heilmann, *Softwaremigration in der Praxis: Übertragung alter Softwaresysteme in eine moderne Umgebung*, 1st ed. Heidelberg: dpunkt-Verl., 2010.