DynaMod Project: Dynamic Analysis for Model-Driven Software Modernization

André van Hoorn*, Sören Frey*, Wolfgang Goerigk†, Wilhelm Hasselbring*, Holger Knoche†, Sönke Köster§, Harald Krause‡, Marcus Porembski§, Thomas Stahl†, Marcus Steinkamp§, and Norman Wittmüss‡

*Software Engineering Group, University of Kiel, Christian-Albrechts-Platz 4, 24098 Kiel
†b+m Informatik AG, Rotenhofer Weg 20, 24109 Melsdorf
‡Dataport, Altenholzer Straße 10-14, 24161 Altenholz
§HSH Nordbank AG, Schloßgarten 14, 24103 Kiel

***Abstract*—Our project DynaMod addresses model-driven modernization of software systems. Key characteristics of the envisioned approach are: (1) combining static and dynamic analysis for extracting models of a legacy system's architecture and usage profile; (2) augmenting these models with information that is relevant to the subsequent architecture-based modernization steps; and (3) automatically generating implementation artifacts and test cases based on the information captured in the models. This paper provides an overview of the DynaMod project.**

I. Introduction

Outdated programming technologies or platforms are typical reasons for long-lived software systems to age. Continuous and sustainable modernization is required for retaining their maintainability.

In our project DynaMod, we will investigate techniques for model-driven modernization (MDM) of software systems. Innovative aspects are the combination of static and dynamic analysis for reverse engineering architectural and usage models, as well as its semantic augmentation by information supporting subsequent generative forward engineering steps, including tests. DynaMod started with the beginning of 2011 and has a two-year funding from the German Federal Ministry of Education and Research (BMBF).

The remainder of this paper provides a brief summary of the envisioned approach (Section II), the consortial partners (Section III), and the working plan (Section IV).

II. Envisioned DynaMod Approach

The envisioned approach can be structured into three phases:

1) Extracting architecture-level models from the outdated system and its usage.
2) Defining architecture-based transformations to models of the desired system.
3) Generating implementation artifacts and tests for the modernized system.

Model extraction will be achieved by combining static and dynamic analysis, as well as additional augmentation. Static analysis of the source code yields a mainly structural view, e.g., including architectural entities and relations. Dynamic analysis, comprising continuous monitoring, adds quantitative runtime behavior information, such as workload characteristics and execution frequencies of code fragments. The models are further refined by manually augmenting them with information that can only be provided by system experts, e.g., mapping code fragments to architectural layers. The conceptual modernization is performed on the architectural level by defining transformations among the extracted models of the outdated system and models of the target architecture. Established generative techniques from model-driven software development (MDSD) [1] are utilized to develop the modernized system. Test cases are generated by including the usage profile information from the dynamic analysis. Further evolution of the modernized systems is based on the MDSD paradigm, keeping the models synchronized with the system implementation.

III. Project Consortium

The DynaMod project consortium consists of the b+m Informatik AG (development partner and consortium leader), the University of Kiel (scientific partner), as well as two associated companies, Dataport and HSH Nordbank AG.

- *b+m Informatik AG:* The b+m group offers IT solutions, including consultancy, software engineering, and maintenance services. Being the initiator of the openArchitectureWare (oAW) framework, b+m is known for its pioneering role in developing and applying MDSD techniques and tools.
- *University of Kiel, Software Engineering Group:* The Software Engineering Group conducts research in the area of software engineering for parallel and distributed systems. One focus is the investigation of model-driven techniques and methods for engineering, operating, as well as evolving software systems, having an emphasis on the consideration of software quality.
- *Dataport:* Dataport provides information and communication technology services for the public administration in the German federal states of Schleswig-Holstein, Hamburg, Bremen, and for the tax administration of the federal states of Mecklenburg-Vorpommern and Niedersachsen.
- *HSH Nordbank AG:* HSH Nordbank is a leading bank for corporate and private clients in northern Germany. As one of the major providers of real estate finance in Germany it focuses on serving commercial clients. In the regionally rooted key industries of shipping, aviation and energy & infrastructure the bank also operates internationally as a top provider of finance solutions.

The DynaMod project is funded by the German Federal Ministry of Education and Research under the grant numbers 01IS10051A and 01IS10051B.

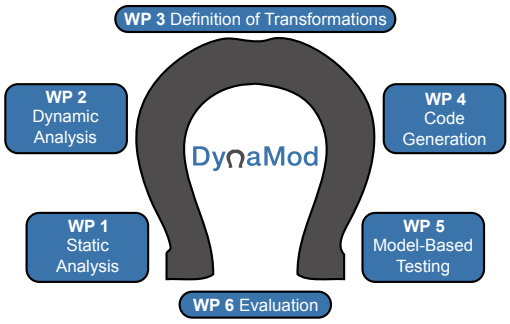

Fig. 1. DynaMod work packages—aligned with the horseshoe model

IV. Working Plan

Based on the envisioned approach outlined in Section II, some more details about our working plan, including involved technologies and desired results, are described in this section. The working plan is structured into six technical work packages (WPs) which are aligned with the horseshoe model for re-engineering, as depicted in Figure 1.

WP1—Static Analysis: This work package investigates methods for the extraction of architectural models utilizing static analysis techniques. Therefore, we will define appropriate meta-models on the basis of the OMG's Architecture-Driven Modernization (ADM) standards, e.g., KDM and SMM. A specific challenge can be seen in providing an adequate representation, abstraction level, and semantic meta-data describing the outdated system to enable smooth integration with information obtained from dynamic analysis (see WP 2). Parsers are required to extract the models from the source code. We will limit ourselves to the programming platforms emerging from our case study scenarios (WP 6).

WP 2—Dynamic Analysis: This work package is concerned with the dynamic analysis of a legacy system's internal behavior and external usage profile while being deployed in its production environment. The instrumentation of systems relying on legacy technology constitutes a technical challenge to be tackled. We plan to extend our monitoring analysis framework Kieker [2], currently restricted to Java-based systems, to support additional programming platforms introduced by the case studies (WP 6). An interesting research question to be addressed here is to identify which information is relevant to the MDM process.

WP 3—Definition of Transformations: The enriched architecture-level models describing the existing system are being translated towards a target architecture employing model-based transformation techniques. Therefore, the corresponding transformation rules are created in this work package. Regarding a first transformation type, KDM may serve as a meta-model for both the source and the target models. A second type transforms these architecture models in code-centric instances. Here, classifying the transformations and exploring transformation patterns can improve the efficiency of the modernization process.

WP 4—Code Generation: Based on MDSD model-to-code transformation techniques, we will use templates to generate implementation artifacts, e.g., SOA wrappers and connectors, from the models of the target architecture (WP 3). The tooling infrastructure will be based on the Eclipse Modeling Project (EMP) which includes the former oAW framework. In this work package, we will focus on the code generation for the case study scenarios (WP 6).

WP 5—Model-Based Testing: Usage models extracted from the legacy system under production workload (WP 2) will be used to generate representative test cases which can be executed automatically by appropriate testing tools. We will focus on tests based on workload generation, allowing to compare quality properties, such as performance and reliability, among the modernized and the outdated system. It is intended to develop transformations from models to test plans to be executed by the load test tool Apache JMeter as well as the extension Markov4JMeter, supporting probabilistic models.

WP 6—Evaluation: In addition to lab studies, the developed methodology and tooling infrastructure will be evaluated based on the three below-described case study systems from the associated partners. Most likely, these systems will not be completely modernized during the DynaMod project. We consider them to be benchmark examples and representatives of modernization projects pending in practice.

- *AIDA-SH (Dataport)* is an information management and retrieval system for inventory data of historical archives. The system conforms to a client/server architectural style, largely based on Microsoft technology: database management systems (DBMSs) based on MS SQL Server (7.0, 2000, and 2003) as well as MS SQL Desktop Engine (MSDE), and a user interface implemented with Visual Basic 6 (VB 6). The major impulse for modernization is the outdated technology.
- *Nordic Analytics (HSH Nordbank)* is a function library for the assessment and risk control of finance products. The library is used in desktop installations (from an Excel front-end), and, deployed to a grid infrastructure, by online trading and batch processing systems. Nordic Analytics is implemented using C#. A modernization will focus on architectural restructuring.
- *Permis-B (Dataport)* is a mainframe system for managing health care allowance. The technical platform consists of z/OS (mainframe operating system), ADABAS-C (DBMS), COMPLETE (transaction processing monitor), as well as the programming environments NATURAL and COBOL. User interfaces are provided by an EskerTun/HobLink terminal emulation and a Web interface (supporting only MS IE 7) based on VB 6 and MS SQL Server 2003. Desire for modernization is pushed by the outdated technology and an eroded architecture, making it difficult to fulfill additional or changed functional requirements in the future.

References

[1] T. Stahl and M. Völter, *Model-Driven Software Development – Technology, Engineering, Management*. Wiley & Sons, 2006.

[2] A. van Hoorn, M. Rohr, W. Hasselbring, J. Waller, J. Ehlers, S. Frey, and D. Kieselhorst, "Continuous Monitoring of Software Services: Design and Application of the Kieker Framework," Dept. Comp. Sc., Univ. Kiel, Germany, Tech. Rep. TR-0921, 2009.