Dynamic Analysis for Model-Driven Software Modernization

André van Hoorn\(^1\) and Holger Knoche\(^2\)

\(^1\)Software Engineering Group, University of Kiel
\(^2\)b+m Informatik AG, Melsdorf

June 08, 2011 @ KoSSE-Tag, Lübeck

The DynaMod project is funded by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IS10051
Overview of Project Topic

Project Overview

DynaMod
Dynamic Analysis for
Model-Driven Software Modernization

Motivation
- Long-lived software systems require continuous modernization
- System behavior & usage important for modernization decisions
- MDSD techniques promise high degree of automation

Methodology
- Combining static and dynamic analysis for model extraction
- Model enrichment supporting reverse and forward engineering
- Architectural transformation from outdated to modernized system
- Generating code & tests employing mature MDSD techniques

Expected Results
- Developing resuable methods, techniques, and tools for MDM
- Evaluation by 3 representative case studies
- Sustainable value of models for MDSD-based evolution & operation

A. van Hoorn, H. Knoche
DynaMod Project — http://kosse-sh.de/dynamod/
June 08, 2011
Project Consortium & Funding

Project Consortium:

1. **b+m Informatik AG**
   *(Development partner, consortium leader)*
   - Comprehensive MDSD know-how
   - Initiated openArchitectureWare (oAW)

2. **Software Engineering Group, Univ. Kiel**
   *(Research partner)*
   - Model-driven engineering, operation, and evolution of software systems
   - Emphasis on software quality (of service)

3. **Dataport**
   *(Associated partner)*
   - Provides ICT services for public/tax administrations

4. **HSH Nordbank AG**
   *(Associated partner)*
   - Leading bank for corporate and private clients in northern Germany
DynaMod Project — http://kosse-sh.de/dynamod/

Project Consortium & Funding

Project Consortium:

1. **b+m Informatik AG**  
   *(Development partner, consortium leader)*

2. **Software Engineering Group, Univ. Kiel**  
   *(Research partner)*

3. **Dataport**  
   *(Associated partner)*

4. **HSH Nordbank AG**  
   *(Associated partner)*

Funding:

- BMBF “KMU-innovativ”
- 2 years (01/11–12/12)

SPONSORED BY THE

**Kosse**

Kompetenzverbund Software Systems Engineering Schleswig-Holstein

Under grant no. 01IS10051
1. **AIDA-SH (Dataport)**

- Information management and retrieval system for inventory data of historical archives
- VB 6, MS SQL Server (7.0, 2000, 2003) and MSDE
Case Study Scenarios

Project Overview

1. **AIDA-SH (Dataport)**
   - Information management and retrieval system for inventory data of historical archives
   - VB 6, MS SQL Server (7.0, 2000, 2003) and MSDE

2. **Nordic Analytics (HSH Nordbank AG)**
   - Function library for assessment and risk control of finance products
   - C# implementation
Case Study Scenarios

Project Overview

1. **AIDA-SH (Dataport)**
   - Information management and retrieval system for inventory data of historical archives
   - VB 6, MS SQL Server (7.0, 2000, 2003) and MSDE

2. **Nordic Analytics (HSH Nordbank AG)**
   - Function library for assessment and risk control of finance products
   - C# implementation

3. **Permis-B (Dataport)**
   - System for managing health care allowance
   - z/OS (mainframe OS), Adabas-C, Natural & COBOL, EskerTun/HOBLink, ApplinX
The Reengineering Horseshoe
Based on [KWC98]

Project Overview

Original Source Code

Target Source Code

Code transformation
The Reengineering Horseshoe
Based on [KWC98]

Project Overview
DynaMod Working Packages

Project Overview

- Dynamic Analysis
- Static Analysis
- Definition of Transformations
- Code Generation
- Model-based Testing
- Evaluation
1 Project Overview

2 Model-Driven Instrumentation and Analysis
Model-Driven Instrumentation & Analysis
Overview of the Approach [vHKGH11]

Model-Driven Instrumentation and Analysis

A. van Hoorn, H. Knoche
DynaMod Project — http://kosse-sh.de/dynamod/
June 08, 2011

Domain
- Domain Model
- Annotated Domain Model
- Aggregated Monitoring Data
- Annotated DADL Model
- Generation (MDSD)
- Program Execution

Architecture
- DADL Model
- Annotated DADL Model
- Aggregated Monitoring Data
- Dynamic Analysis

Implementation
- Source Code
- Annotated Source Code
- Raw Monitoring Data
- Static Analysis

Manual Annotation

- Analysis

- Source Code

- Annotated Source Code

- Aggregated Monitoring Data

- Program Execution

- Dynamic Analysis

- Manual Annotation

- Source Code

- Annotated Source Code

- Aggregated Monitoring Data

- Program Execution

- Dynamic Analysis

- Manual Annotation
Overview—DynaMod Examples

Model-Driven Instrumentation and Analysis

Public Sub searchBook()
...
crm.getOffers
End Sub

DADL (DynaMod ADL)

DMeasurements
:AvgOperationRTMeasure
avgRTMillis=5676
Measurements
count=67643
:OperationInvocationCount

Catalog
Bookstore
searchBook()

DMeasurements
:OperationInvocationCountMeasure

:OperationExecutionProbe

DInstrumentation

:OperationExecutionRecord
tin=34
tout=38

class="CRM"
operation="getOffers"

:OperationExecution

tin=34
tout=38

:OperationInvocationCount
count=67643

DEvent

Monitoring Events

MonitoringRecord

tin=34
tout=38

@intercept#Call:OpExecIcptr["Bookstore", "searchBook"]
crm.getOffers

A. van Hoorn, H. Knoche

DynaMod Project—http://kosse-sh.de/dynamod/

June 08, 2011 9 / 11
Microsoft Visual Basic 6

- No formal grammar exists
- MSDN documentation of language features incomplete

Challenging language features (examples)

- Meta-data mixed with source code in module files
Microsoft Visual Basic 6

- No formal grammar exists
- MSDN documentation of language features incomplete

Challenging language features (examples)

- Meta-data mixed with source code in module files
- **Syntactic ambiguities due to inconsistent calling conventions**

```
sub_a x, y, z
func_a (x, y, z)
Call sub_a (x, y, z)
sub_b (3+5)
func_b (3+5)
```
Microsoft Visual Basic 6

- No formal grammar exists
- MSDN documentation of language features incomplete

Challenging language features (examples)

- Meta-data mixed with source code in module files
- Syntactic ambiguities due to inconsistent calling conventions
- Whitespaces relevant
Microsoft Visual Basic 6

- No formal grammar exists
- MSDN documentation of language features incomplete

Challenging language features (examples)

- Meta-data mixed with source code in module files
- Syntactic ambiguities due to inconsistent calling conventions
- Whitespaces relevant
- Colon (‘:’) used as end of statement as well as label delimiter
Conclusion

Model-Driven Instrumentation and Analysis

DynaMod

Dynamic Analysis for Model-Driven Modernization

Additional Information:

- [http://kosse-sh.de/dynamod](http://kosse-sh.de/dynamod) (in German)
Literature

Model-Driven Instrumentation and Analysis

Requirements for integrating software architecture and reengineering models: CORUM II.

Thomas Stahl and Markus Völter.

DynaMod project: Dynamic analysis for model-driven software modernization.

André van Hoorn, Holger Knoche, Wolfgang Goerigk, and Wilhelm Hasselbring.
Model-driven instrumentation for dynamic analysis of legacy software systems.