iObserve

Integrated Observation and Modeling Techniques to Support Adaptation and Evolution of Software Systems

Wilhelm Hasselbring,¹ Robert Heinrich,² Reiner Jung,¹ Andreas Metzger,³ Klaus Pohl,³ Ralf Reussner,² Eric Schmieders³

¹Kiel University, ²KIT, ³Uni Duisburg-Essen

3rd Workshop of the DFG Priority Programme 1593
Munich, Oct. 9 - 11, 2013
Project Goals

- Future long-living software systems will be engineered using **third-party** software services and infrastructures.
  - Key challenges for such systems will be caused by **dynamic changes** of deployment options on cloud platforms.
  - Third-party services and infrastructures are neither owned nor **controlled** by the users and developers of service-based systems.
  - System users and developers are thus only able to **observe** third-party services and infrastructures via their interface, but are not able to look into the software and infrastructure that provides those services.

- The iObserve project addresses those challenges by following a model-based approach.
  - Develop and validate new models and techniques for runtime observation and anomaly detection of future service-based software systems deployed on third-party platform and infrastructure services,
  - through extending and integrating previous work on **monitoring**, **meta modeling**, and **service-oriented systems**.

Changes in domain assumptions D_a, refer to Carlo's talk
Knowledge Carrying Software

Our Approach:
- Models+Verification@Runtime

Research Questions
- How to keep (design) models consistent with the (adaptive) system?
  - Observe it!
  - Monitoring & analysis of distributed cloud-based applications
- How to conduct continuous modeling and analysis?
  - For quality prediction and forecasting
    - Performance, Cost, Geo-Location, …
    - Evaluation of data-migration-policies, as example adaptation scenario

Wilhelm Hasselbring, iObserve, 3rd WS of SPP1593, Munich, Oct. 9 - 11, 2013
**Scenario: Data Management on the Cloud**

<table>
<thead>
<tr>
<th>Online Shop</th>
<th>DBaaS Provider</th>
<th>Infrastructure Provider 1</th>
<th>Infrastructure Provider 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Germany flag]</td>
<td>![Germany flag]</td>
<td>![Green flag] ![Red flag]</td>
<td>![USA flag]</td>
</tr>
</tbody>
</table>

- **Online Shop**
  - run discount campaign

- **DBaaS Provider**
  - observe increasing load
  - predict QoS degradation
  - initiate DB-replication process

- **Infrastructure Provider 1**
  - copy DB to DB'

- **Infrastructure Provider 2**
  - host DB'
Agenda

- Project goals
- **Addressed Case Study: CoCoME**
- Results
  - Reverse Engineering of CoCoME
  - Model-driven Instrumentation and Analysis
  - Enforcing Data Geo-Location Policies
- Summary & Outlook
Agenda

- Project goals
- Addressed Case Study: CoCoME

Results
- Reverse Engineering of CoCoME
- Model-driven Instrumentation and Analysis
- Enforcing Data Geo-Location Policies

Summary & Outlook
Reverse Engineering of CoCoME with Kieker

Goals

- Better understand the existing CoCoME implementation
- Compare reconstructed models in relation to the original design
- Provide a basis for SOA and cloud migration

Realization

- Static analysis of the deployment architecture
- Dynamic analysis of the application architecture
  [Hasselbring 2011, van Hoorn 2012]
- Comparison of reconstructed models with the design
  (software reflexion models [Murphy et al 2001])
Reverse Engineered Models for CoCoME (via Kieker)
Software reflexion models
Agenda

- Project goals
- Addressed Case Study: CoCoME

Results
- Reverse Engineering of CoCoME
- Model-driven Instrumentation and Analysis
- Enforcing Data Geo-Location Policies

Summary & Outlook
Model-driven Instrumentation and Analysis

Automatic Transformation

In instrumentation record languages (IRL), the transformation $T_{IRL}$ generates monitoring records.

In instrumentation aspect languages (IAL), the transformation $T_{IAL}$ generates probe placements.

Target language specific:
- C struct, Java class, Perl record

Technology specific:
- AspectJ, CXF, WF-Filter, AspectC, ...

Target language specific:
- C, Java, Perl API
Model-driven Instrumentation and Analysis

Instrumentation Aspect Language

- Queries to determine application model nodes
- Target language support
- Support for different probe technologies
- Support for probe configuration

```java
package org.spp.kieker.instrumentation

probe /TradingSystem/**/*(*) : * {
    before collect BeforeOperationEvent(
        time, id, index,
        ./name,../name)
    after collect AfterOperationEvent(
        time, id, index,
        ./name, ../name)
}

collect AverageMethodResponseTime ( String methodName )
    average AfterOperationEvent - BeforeOperationEvent
    scope ( pcm.repository.Operationsignature.entityName == methodName )
    measure BeforeOperationEvent
    measure AfterOperationEvent

Evaluation metrics based on MAMBA [Frey 2012].
```

Wilhelm Hasselbring, iObserve, 3rd WS of SPP1593, Munich, Oct. 9 - 11, 2013
Agenda

- Project goals
- Addressed Case Study: CoCoME

Results
- Reverse Engineering of CoCoME
- Model-driven Instrumentation and Analysis
- Enforcing Data Geo-Location Policies

- Summary & Outlook
(1) Palladio-based Performance Prediction

(2) Component owner sends planned action

(3) Check access & location vs. DL-policy

(4) Responds with DECLINED
Enforcing Data Geo-Location Policies

Policy decision process

Planned action

Create Model for verification

Check model against Property

[unsatisfied]

[satisfied]

Create Accept

Create Decline

Response

Model

\[ \models \neg (g \in E) \]

Idea: Runtime verification, e.g., via model checking

Wilhelm Hasselbring, iObserve, 3rd WS of SPP1593, Munich, Oct. 9 - 11, 2013
Agenda

- Project goals
- Addressed Case Study: CoCoME
- Results
  - Reverse Engineering of CoCoME
  - Model-driven Instrumentation and Analysis
  - Enforcing Data Geo-Location Policies
- Summary & Outlook
Integrate expertise of three groups

**CAU:**
- Kieker [van Hoorn 2012]
- SLAStic [van Hoorn 2009]
- MAMBA [Frey 2012]

**KIT:**
- Palladio [Becker 2009]

**UDE:**
- Runtime verification (S-Cube) [Metzger 2010, Schmieders 2001]
Summary & Outlook

Summary

- Integration of Palladio and Kieker
- Elaboration of the CoCoME case study
- Scenario and approach for data privacy policies

Outlook

- Multi-objective optimization [Frey 2013]
  - Performance / Cost / Data policies
- Anomaly detection, Analysis [Ehlers 2011]
- Model-driven CoCoME (co-operation with other SPP projects)
  - iObserve DSL to augment Palladio
  - Generator for complete CoCoME system

(Technical report on this presentation is available at http://eprints.uni-kiel.de/22077/)

Invitation:

Joint Kieker/Palladio Days 2013
Symposium on Software Performance
27 - 29 November 2013, Karlsruhe
Deadlines: Oct. 19, 2013 (abstracts)
http://www.kieker-palladio-days.org/
Publications


