Model-driven development in the context of technical SOA


Michael Rauch, Software Architect
Christoph Gutmann, Software Architect
Swiss Mobiliar: key facts at a glance

Swiss Mobiliar

- is structured as a mutual company and is first and foremost obliged to its customers
- is the oldest private insurance company in Switzerland, founded in 1826
- focuses on the Swiss and Liechtenstein markets
- offers a broad selection of modern insurance products, including life policies
- lets insured persons participate in the company’s success in the form of bonus payments
- has the highest solvency ratio of all insurance companies operating in Switzerland
- has approximately 1.5 million policyholders
- employs a workforce of around 4,000 employees and 300 trainees
- employs a workforce of around 400 employees in IT departments
Motivation

The SOA Situation

✓ define core concepts and standards
✓ reference architecture
→ enforce interface standardization!
→ know all dependencies!

Initiative „MAIA“: Mobiliar Application and Infrastructure Automation

Code Generation (Forward Engineering)

Dependency Management of Services
How we solved that problem

with forward engineering by using models
→ edit the model
→ use the model
Editing the Model

- Xtext
- EMF

- POJO

- EMF

Domain Model
one per company
Using the Model

Domain Model
one per company

Generator
one per company

Artifacts
readonly

Forward Engineering

- EMF
- Xtend
- Xpand
- EMF-Compare
- <none>
Using the Model

- Domain Model
  - one per company

- Generator
  - one per company

- Artifacts
  - readonly

- Forward Engineering

- Datatypes
- Services
- Components
- Wiring + Planning

- Service Catalog
- Java Gluecode
- Reporting
provide a service…

```java
@Stateless
public class MyBean implements MyService
```

- JEE – JBoss 5
- JEE – JBoss 7

consume a service…

```java
@Inject private MyService service
```

- JEE – JBoss 5
- JEE – JBoss 7
- SE
- Spring
- GUI - Liferay on Tomcat
- GUI - Standalone JBoss 7

pull generated code from Maven repository
component dependencies (wiring)

reuse of datatypes

- visualize dependencies
- identify hotspots
- analyze impacts
- plan refactorings
Main Focus is on the Domain Model

Domain Model

edit

use
Benefits

- **velocity:** simplified development and integration of services
- **portability:** business-logic code is decoupled from webservice technologies
- **governance:** parts of reference architecture are enforced
- **consistency:** specification, code and documentation are consistent
- **SPOT:** Single Point of Truth implemented by modeling and forward engineering
- **manageability:** well-known services, components and dependencies, unified lifecycle management, planning as a part of the domain model
- **sustainability:** safely canned knowledge by defining the domain model, implementing generators and automated publishing processes
- **expandability:** domain model and generators can grow as needed, new partitions can be introduced independently
Giveaways

- start small
  - do small iterations, deliver new features with each iteration
  - chose a small project as pilot
  - apply changes in your sphere of influence

- strive for acceptance
  - deliver mature product with high quality
  - integrate with existing tools and workflows
  - offer support and coaching

- reduce variability
  - only introduce concepts in your domain model that you really need

- when growing
  - make friends
  - explain concepts and opportunities
Contact

- Team Members of MAIA:
  - michael.rauch@mobi.ch
  - christoph.gutmann@mobi.ch
Thank you for your attention!