ModelIrepository @ T-Mobile
Umsetzung und Einsatz

iX CeBIT Forum 2009
Carsten Sensler, T-Mobile Deutschland GmbH

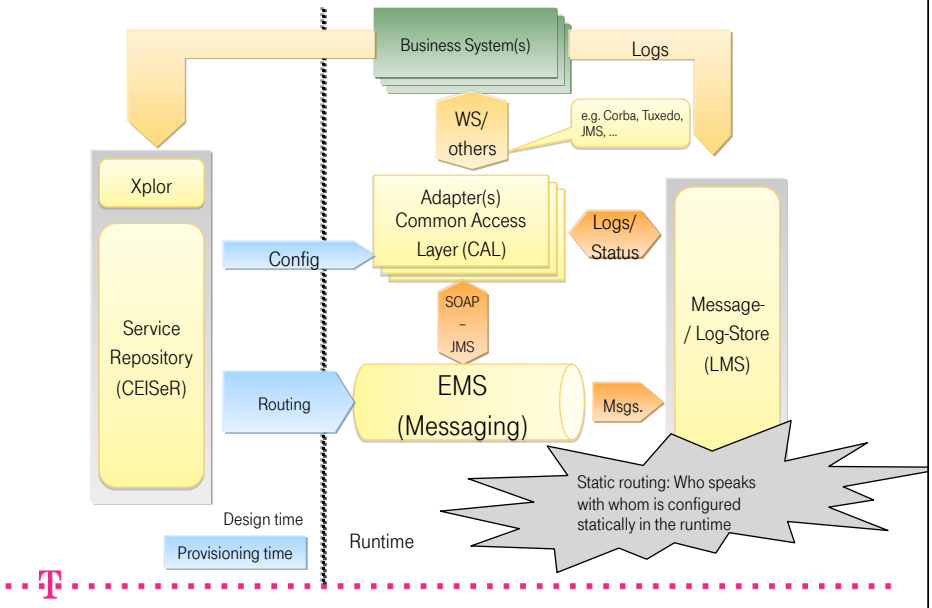
Table of Contents

- SOA Backplane overview
- Model repository @ T-Mobile
- Domain specific example: SOA
  - Common Enterprise Integration Service Repository
- Experiences
- Summary

Table of Contents

- SOA Backplane overview
- Model repository @ T-Mobile
- Domain specific example: SOA
  - Common Enterprise Integration Service Repository
- Experiences
- Summary

SOA Backplane – overview (simplified).



SOA Backplane – overview (international view).

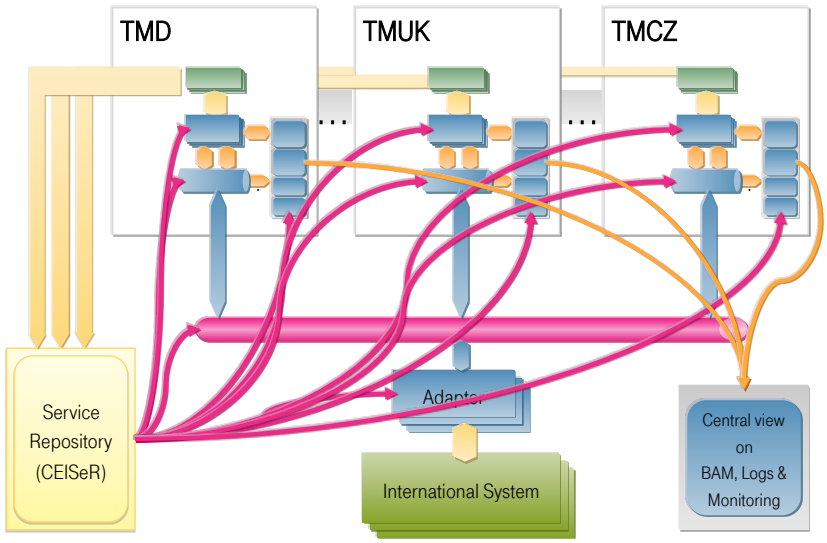
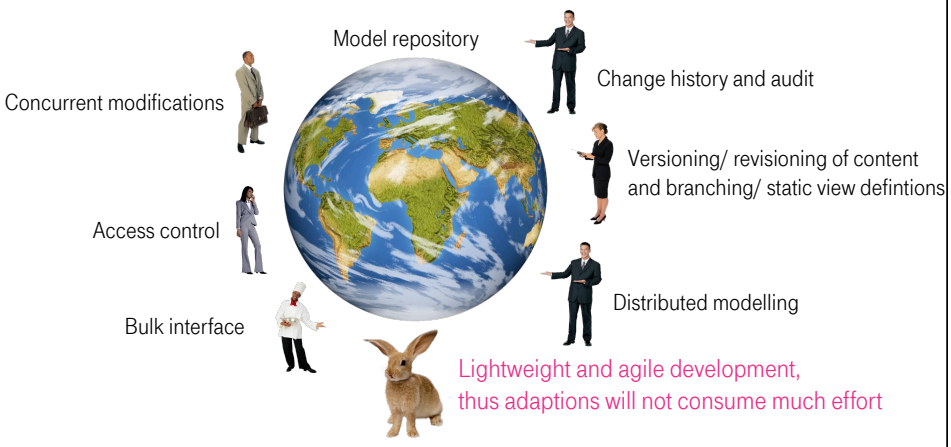


Table of Contents

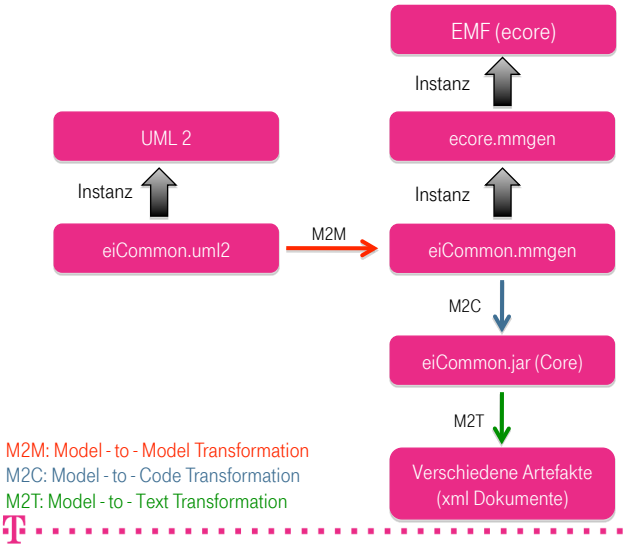
- SOA Backplane overview
- Model repository @ T-Mobile
- Domain specific example: SOA
  - Common Enterprise Integration Service Repository
- Experiences
- Summary

Model repository in general.



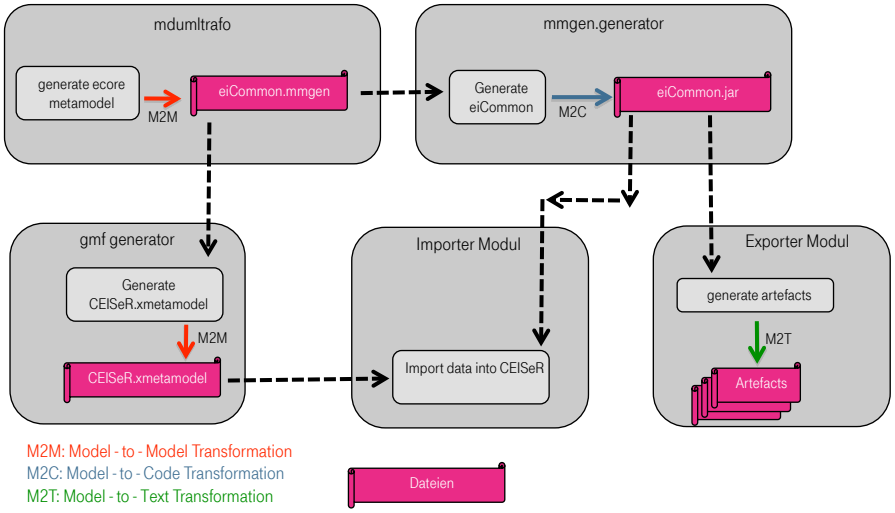
- Model repository stores Metadata where the context (domain) is defined in a formal model.
- The concepts of the model repository are defined in a meta meta model

Transformation view for generating the sources for the model repository and the export interface.

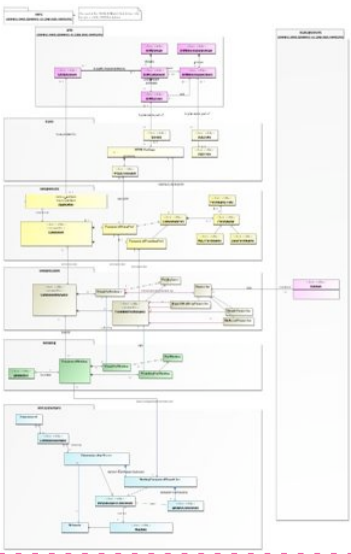


M2M: Model - to - Model Transformation
M2C: Model - to - Code Transformation
M2T: Model - to - Text Transformation

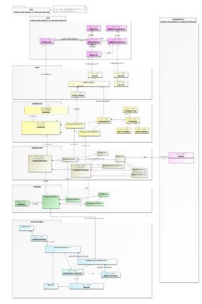
oAW-based MDSD generator chain for generating the sources/ artefacts for the model repository.



What is generated from the meta model ?



What is generated from the meta model ?



Simple constraint checks

Metaclasses (Java beans)

Hibernate persistence mapping (XML)

Generated by openArchitectureWare

Generated by Hibernate Schema Builder

Oracle DB schema

What is generated from the meta model (independent from the specific domain)?

Generated model management code

Java beans (instances of generated meta classes) represent the modelled entities

Persistence and instantiation of Java beans is done by Hibernate

Simple constraint checks

Metaclasses (Java beans)

Hibernate persistence mapping (XML)

Generated by openArchitectureWare

Generated by Hibernate Schema Builder

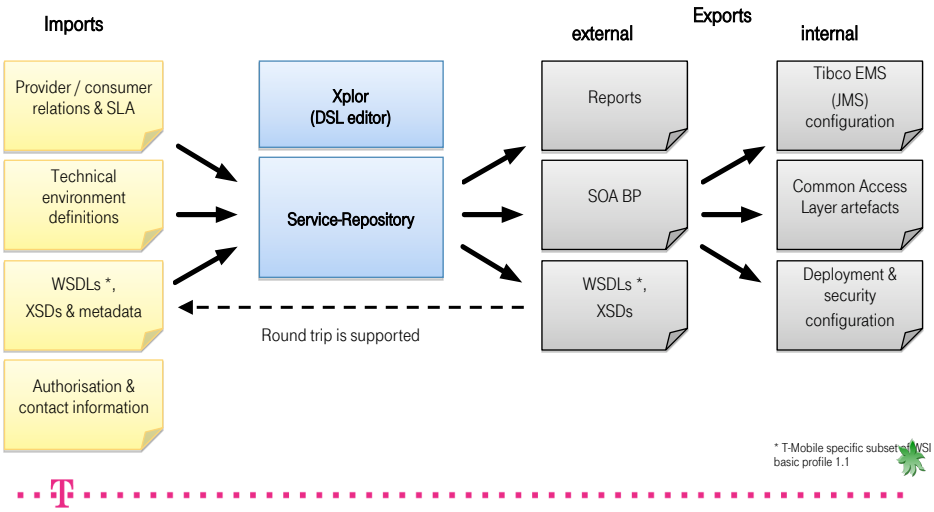
Oracle DB schema

Data storage in Oracle DB

Table of Contents

- SOA Backplane overview
- Model repository @ T-Mobile
- Domain specific example: SOA
  - Common Enterprise Integration Service Repository
- Experiences

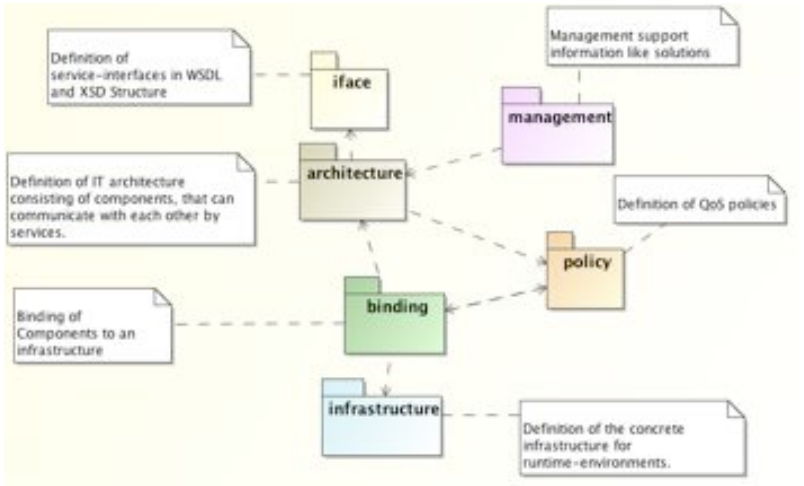
SOA Backplane – Service Repository Interfaces.



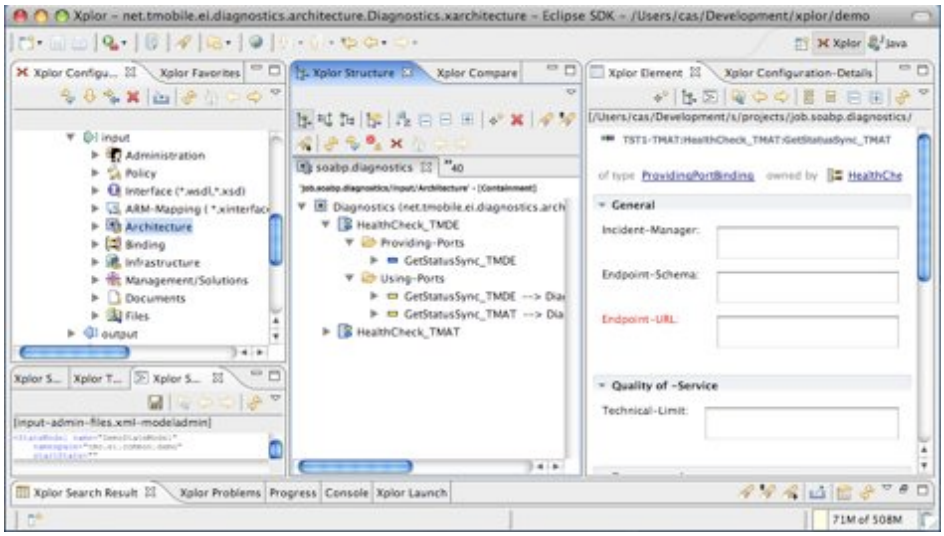
The Service Repository in a nutshell.

- Provide all information needed by service participants for consistent service implementation and utilisation (architecture)
- Store definition of different SOA backplane environments and binding of service participants to these environments (binding)
- Support fully automated configuration of SOA backplane environments (dev, test, prod, ...) (service provisioning)
- Support SOA governance (service discovery, reuse) and impact analysis (change / incident contact information) (management)
- Support change and configuration management by the revisioning concept and the auditing facility of CEISeR

The Underlying Meta-Model.



eXploring Modells – CEISeRs frontend.



Xplor Eclipse based client.

- Eclipse plug-in or Eclipse-based RCP
- DSL editor for CEISeR
  - Generic GUI based on CEISeR meta model, Import DSL meta model and local configuration
  - Partially generated from the CEISeR met model
- Supports
  - Editing of job definition files (manifest, architecture, binding, infrastructure; search, edit, delete, insert, change, ...)
  - Analysis of job results (model diff, constraint violations)
  - Analysis of model dumps (offline snapshot of partial or full model)
  - Analysis of a diff between two model revisions (diff of two snapshots)
  - **Check-in and Check-out of model fragments**
- Various wizards for making the work with CEISeR more efficiently

Table of Contents

- SOA Backplane overview
- Model repository @ T-Mobile
- Domain specific example: SOA
  - Common Enterprise Integration Service Repository
- Experiences

Benefits of MDSD based Approach.

- MDSD allows changes and enhancements of logical model and general enhancements of functionality to be executed without the need to rewrite mayor parts of code or recurring implementation efforts
- Model to model transformations allow efficient implementation of complex transformations needed for
  - Import of definitions form various sources and formats
  - Export of definitions to various formats
  - Data migration due to model changes / enhancements and harmonisation activities
- oAWs generator framework and powerful template language allow easy generation of all kinds of artefacts
- Application architecture supports clear separation of generated and hand coded parts
  - Easy adaptation of hand coded parts to when even necessary

Flexibility/ Extendibility without manual code changes.

- Many changes of the domain model will be propagated directly into the Xplor
  - Example: Adding of attributes/references to already existing entities
    - A full rebuild/regeneration ends in
      - New DB schema
      - New access methods
      - DB migration script
      - Xplor UI enhancements

  In a very agile domain (like SOA) I recommend to use MDSD to be very flexible without extra effort

- Note: Of course, the export has to be adapted manually by utilising the generated export- API

Future Prospects – CEISeR Evolution.

- With our strategic partner we will shift CEISeR to a standard product to decrease/ avoid internal development effort and therefore to safe budget (in the long run). But we will try to place our concepts into the product.

- We will realize human-oriented workflows with the use of bpm technologie regarding the processes working with CEISeR to formalize the processes
  - Defining a contract (service interface definition)
  - Defining an architecture, binding, ...
  - Service Lifecycle management

Additional Informations

- Dipl.-Ing. Carsten Sensler
  - Employee of T-Mobile Deutschland GmbH since April 2007 (but since December 2005 working in the SOA Backplane program)
  - Department of Enterprise Integration
  - System & Solution Designer
  - Functional leader of the international Service Provisioning Team
  - Contact: publication@sensler.de

- oAW – openArchitectureWare : http://www.openarchitectureware.org
- C. Sensler, A. Karalus, SOA@T-Mobile – Vollautomatische Service Provisionierung auf dem ESB – Teil 1-3 in: JavaMagazin, 10.2008 – 12.2008, http://www.sensler.de/public.html
- C. Sensler, A. Karalus, M. Märtens, Ein Blick hinter die Kulissen - ModelIrepository@T-Mobile, in JavaSPEKTRUM 1/2009