THE IT-ARCHITECTURE PROFESSIONALS

Case study at Austrian Railways – illustrating concepts, application and ROI of Convergent Architecture with MDA

Jens-Hagen Syrbe
Outline

◆ Introduction
  - Austrian Railways (ÖBB) and the LDZ project
◆ MDA @ work
  - Convergent Architecture
  - Modeling style
  - Generation of source code and infrastructure code
  - Performance and Optimizations
◆ Conclusion
  - LDZ project statistics
  - MDA Benefits
Introduction: Austrian Railways (ÖBB)

- Industry: National Railroad, Central IT
- Project: Portal application for determining charges on a "per usage" basis
- URL: www.woebb.at

ÖBB facts:

- Staff: 49,000
- Track: 3,545 miles
- Major building block in the trans-European railroad network
Introduction: Project Goal

- Calculation of charges by mileage, track category, weight, time slot etc. to reflect actual usage of railway infrastructure
- Creation of the portal application, based on a uniform J2EE infrastructure
- Implementation of future-oriented IT-architectural style
Introduction: Project Scope

* 15,000 Track-Slots
* 750,000 Sensor points
* 300 Ad-hoc changes/day
* Across 50 regional stations and teams.

EAI with existing:
ARTIS / RZU systems
with data from
* 8,000 Trains/day
* 400,000 Sensor hits/day

* Monthly bills with precise, itemized usage data.
Approach

◆ Convergent Architecture
  ▶ One single model for business experts and IT experts
  ▶ This model/repository allows different views, revealing information on different levels of abstraction
    ● Traceability
    ● Proper separation of concerns between business and IT

◆ MDA
  ▶ Core element of Convergent Architecture
  ▶ Model-driven (vs. code-centric), generative approach
  ▶ Generation of EJB, web artefacts (html, jsp), test environment and build infrastructure
Advantages of MDA

- **Faster**
  - Increased productivity through generation of clue code (e.g. home and remote interfaces for EJB)

- **Better**
  - Consistently high quality of generated code, less manual coding
  - Propagation/Enforcement of project guidelines in the code

- **Specialization of skill set**
  - Not all developers need to understand all details of the technologies involved

- **Future-proof**
  - Easy migration to new implementation technologies with model+generation
Convergent Architecture defines three abstractions of business objects (Organizations, Resources and Processes)
Organization

- Management of Resources and Processes
- Contain mostly finder and create methods
- Encapsulates home objects
- No own data, pure mediator functionality

Stateless Session Bean
Resource

- Core business objects (Track/Trasse, Train/Zug, Client/Kunde)
- Contains business logic and data
- Has persistent attributes and associations
  - Associations implemented using portable assoziation as generated by ArcStyler

Entity Bean with Container Managed Persistence
EJB Modeling Style (4)

◆ Process

▶ Heavy-weight process (e.g. "Use Case Train Run"/GFZF)
  ● Long-lived
  ● Compares to a file documenting a procedure or business flow Entity Bean

▶ Light-weight process (e.g. activation of a train)
  ● Short-lived – e.g. single user interaction Statefull Session Bean
EJB Modeling Style (5) – Some Special Cases

- Search across EJB boundaries
  - Complex search queries involving many EJBs realized directly at data base level, but encapsulated in Data Access Objects
  - Separately modeled classes

- Komponententest
  - Explicitly modeled, thus testing either individual beans or whole use cases
  - Hierarchy of tests
For EJB’s

- Complete home and remote interface
- Bean class with
  - Attributes and associations
  - Skeletons for business methods
  - Helper methods (e.g. access to home interface for dependencies)
For Java classes

- Generation of attributes and operations according to modeled properties (e.g. visibility)
- Access methods for attributes
Generates Artifacts (3)

- Web Application
  - Accessors
    - Generations of state machine from model
  - Representer
    - HTML pages and JSPs with predefined elements
Test and Build Environment

- Test classes with mit skeletons
- Build scripts for Ant

- *.java
- *.xml
- *.bat
- *.jpr
Application Structure

ArcStyler: UML model

Organization
Transformation no data

Process
Light weight
Heavy weight

Resource
Business information

Value Objects
Query on several tables

Data base

Client
Representer
Representer
Representer
Representer

Web Server
Accessor

Stateless Session Beans
Stateful Session Beans
Entity Beans
Value Objects

JDBC
Deployment Environment

Client: IE 5.5

Ca. 5 Server

Tomcat; BAS 4.5

Ca. 2 DB-Server

Oracle 8.1.7
Performance ???

- **Scale of application**
  - Number of Entity Beans processed per day: 200,000-300,000
  - Typical number of concurrent users: 250

- **Mapping of technology-agnostic business model to EJBs and database access**
  - Implemented by technology projection that optimizes standard scenarios
  - Room for optimization required to handle special cases performantly

- **Compromise between performance and other aspects such as easy of maintenance, simplicity of design**
  - Follow standards and defaults of modeling style and mapping if possible
  - Optimize if necessary
  - Extensions to modeling style and generation ensure controlled technology-specific optimizations
Optimizations

- Optimized Configuration of Container
  - Check data base tables prior to creation
  - Cache sizes (per transaction, overall)
  - Transaction modes
  - Definition of colocation

- Optimized database structures
  - Storage management or indices

- Data Access Objects (DAO)
  - To handle complex queries across several types
  - Use connection and transaction management provided by container
  - Encapsulate data base access in classes at model level
Austrian Railways – LDZ Project Statistics

- MDA init/bootstrap phase: 2 months, 3 persons
- Realization phases: 15 persons in 4 teams
  - Number of use cases - 15
  - Number of components – ca. 250
  - Number of modeled classes – ca. 700
- Total development time <1 Year:
  - First deployable system increment: 3 Months after start.
  - Beta test after year: May 2002.
  - In production use: July 2002
  - First production results, billing etc. in August 2002.
MDA Benefits

- Uniform, easy to maintain component infrastructure
- Enforcement of IT architectural guidelines through generation (as opposed to error-prone, paper-based organization efforts)
- Encapsulation of technological, non-business details: frees up important time for solving business problems and development of mapping guidelines and optimizations
- Quality assurance at a high level through model verification
- Increased productivity and quality through code generation
- Automatic generation of the professional build environment
- Model-driven test infrastructure enables convenient and structured component testing
- Migration problem solved (here: Oracle 8.1.6->8.1.7 in mid-devel.)
THE IT-ARCHITECTURE PROFESSIONALS

Model Driven Architecture for the Enterprise
http://www.ArcStyler.com/
October, 2002
jhs_UMLworkshop_oebb.ppt

Interactive Objects Software GmbH
2105 Bridgeway
Sausalito, CA 94965
ph. 415-339 0412
fax 415-339 0411
Jens-Hagen.Syrbe@io-software.com