Building Enterprise Database Applications

using Rational Rose, Java and Mr Architecture

A short course by Kade Hansson

Course Contents

- **UML, object-orientation and design patterns**
- **Java language and essential APIs**
- **Java GUI components and event model**
- **Java I/O and TCP/IP sockets**
- **JDBC, Servlets and JSPs**
- **Mr Architecture**

Building a Mr Architecture Application

- Model the database using a Class Diagram in Rational Rose
  - One *EntityBean* class per bean type
  - All classes should be abstract
  - Remember to indicate primary keys
  - Be sure to apply all relationships, including multiplicities but omitting navigability
- Generate getters and setters using *CreateGetSet* and manually tweak results
- Generate further EJB components using *GenerateEJB* and tweak results
- Build a relational schema from the model using Rose
- If necessary (e.g. when the class diagram was created using Java notation,) tweak the schema to include data types and widths[1]

[1] Additionally, the current meta data CSV file generation script *MetaGen* requires you to reverse engineer this schema to obtain a relational model in order to pick up appropriate meta data

Building a Mr Architecture Application (continued)

- If meta data is required (e.g. there is a table with clobs,) build CSV files using *MetaGen* and tweak manually
- Create a *MrContainerFactory* for the application which has a *populateDeploymentContainer()* implementation containing *Container.addType()* calls for all bean types and compile– **Factory** pattern
- Execute *DeploymentTool* passing in the name of this class
  e.g.
```
java -cp ../WEB-INF/classes;../WEB-INF/lib/j2ee.jar
        au.gov.tas.dpiwe.mr.tool.DeploymentTool
        au.gov.tas.dpiwe.lid.LIDContainerFactory
```
- Create a *populateContainer()* implementation containing *Container.addType()* calls for core bean types and *Container.addTypeLater()* calls for modular bean types
- Compile application with deployed beans

- *Details of some steps of this process are available as a recipe, and some other steps are automated by using application-specific scripts*

Building a Mr Architecture Application (continued)

- Choose a form for the front-end application
  - Client-side Mr Architecture applet or application with server-side Mr Architecture connected to a database using JDBC
  - Server-side Mr Architecture connected to a database
    - choose either Servlets or JSPs, with or without client-side scripting or active content (e.g. applets)
    - possibly with Ms Architecture extension for data presentation
  - Client-side trusted Mr Architecture applet or application connected to a database using JDBC
- Produce front-end in chosen form
  (model, generate skeleton from Rose and fill-in methods)