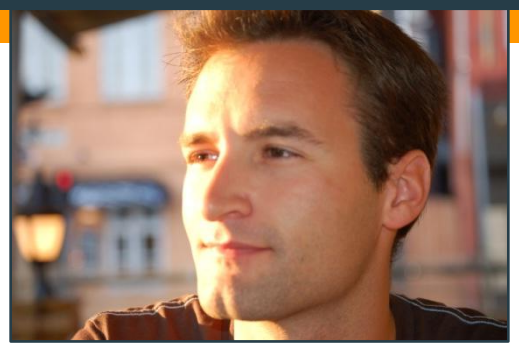
An OFBiz introduction

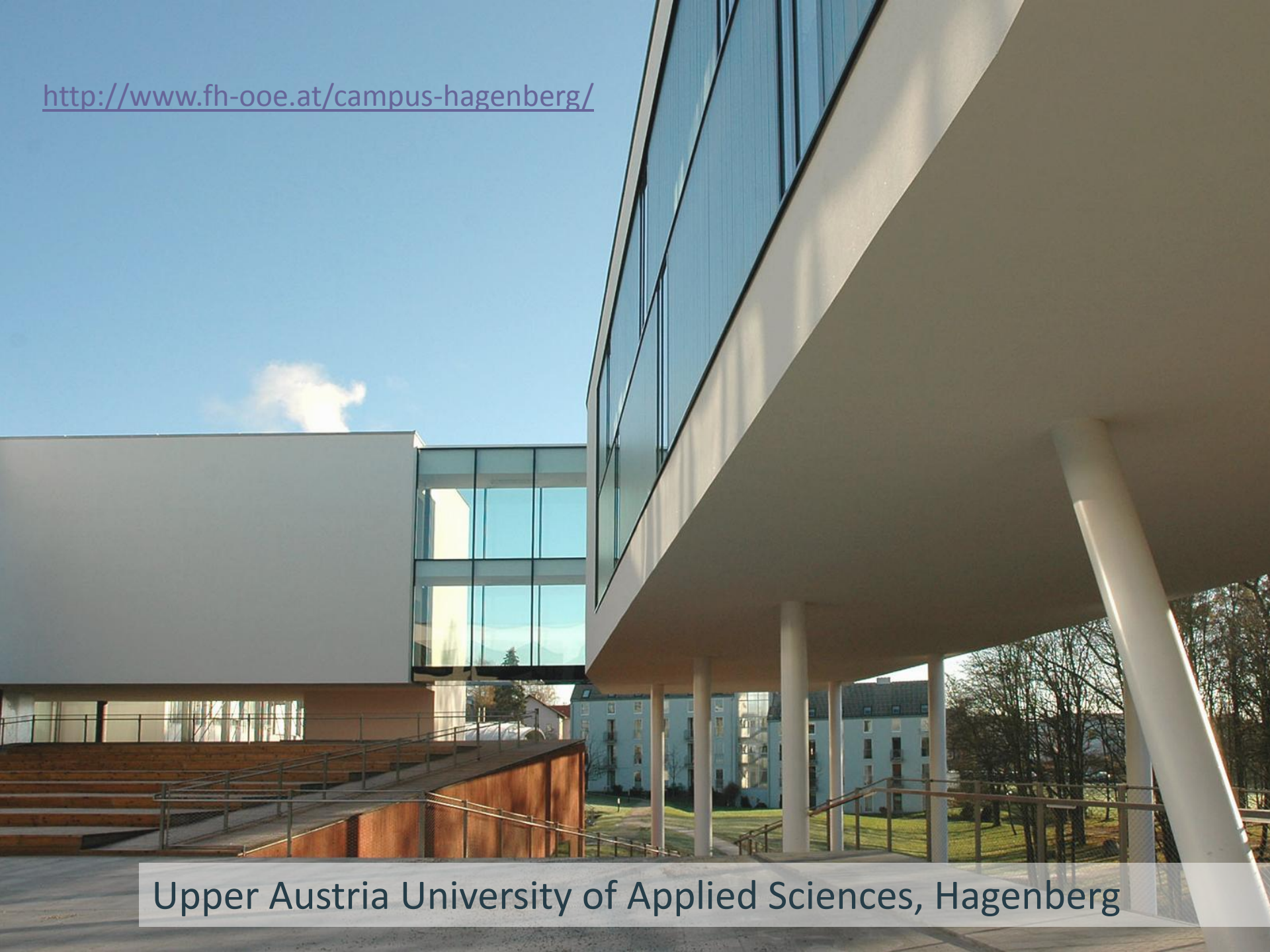
ERP-systems, Aalborg University, Information Systems

Klemens Schwarz (Upper Austria University of Applied Sciences, Hagenberg)



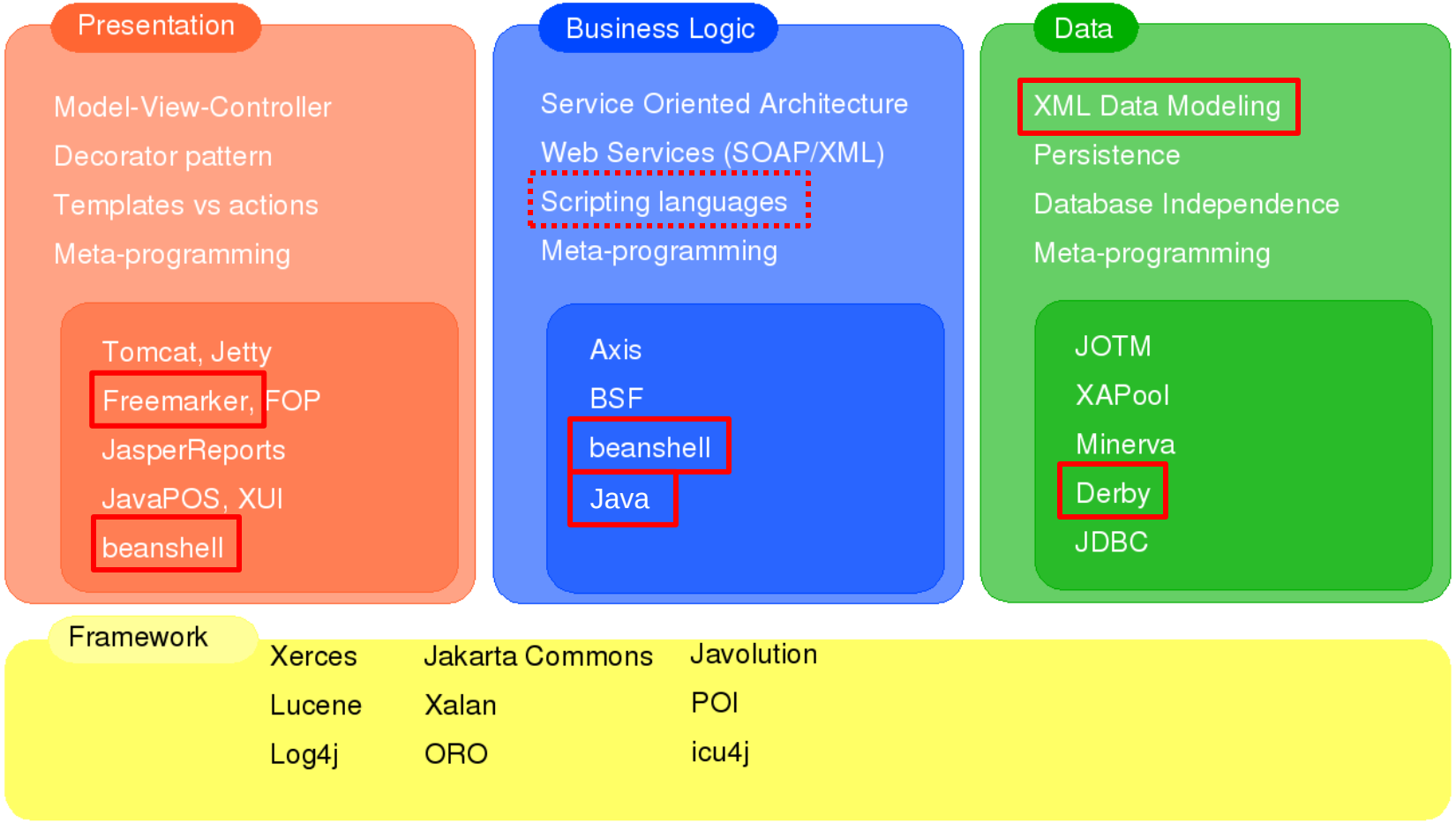
Introduction of myself

- Klemens Schwarz
- I studied *Software Engineering for Medical Purposes* at the Upper Austria University of Applied Sciences, Hagenberg
- Now I teach technical lectures (Basics of Programming, Information Systems, …) at the study course *Communication and Knowledge Media* (Hagenberg)



Upper Austria University of Applied Sciences, Hagenberg

The OFBiz Framework



Source: http://www.opensourcestrategies.com/ofbiz/index.php

Benefits of such an architecture

- Independent of platform and database
- Development style is agile and flexible
- Code re-use with Service Oriented Architecture
- New application can be plugged in easily (and often with little effort)

OFBiz Entity Engine (Data Model Layer)

- Entity = a small unit of data model
  - Declared in XML files, key/value pairs
  - Found by delegator
  - → Generic class with get, set, store methods

- Modelling the data within EntityModel.xml and EntityGroup.xml
- Interfacing the database with EntityEngine.xml and fieldtype.xml

OFBiz Service Engine (Business Logic Layer)

- Service = small unit of business logic
  → The Service Engine displays the business logic of the framework.
- Services are defined in services.xml files.
- Service Engine directly plugs into controller → no parsing needed
- Forms are automatically generated (and updated, if a service was changed)

OFBiz Presentation Layer

- Consists of different technologies:
  - JSP
  - Freemarker (*.ftl), Beanshell (*.bsh), Jpublish
  - Screen, Form, Menu, Widget

How to develop applications – Overview (1/2)

- OFBiz is developed as a three layered architecture. For building an application you build up these layers.

1. Data Layer:
   1. Design the data tables
   2. Build the data layer within the entitydef folder:
      - entitygroup.xml for defining the tables
      - entitymodel.xml for field types, primary keys and relationships
2. Business Logic Layer:
   1. Decide which services you need
   2. Build this layer:
      - In the servicedef folder: here you have to define the services including their types, invoked methods, in and out parameters, …
      - Implement the services eitzer via Beanshell (/script) or Java (/src)

How to develop applications – Overview (2/2)

3. Presentation Layer:
   1. Decide what pages the user needs.
   2. Build this layer:
      - The webapp folder contains the forms that might be included in some screens.
      - The widget folder contains the screens the application pages present.
4. Controller:
   - The Controller receives requests and forwards them to their location (URLs are also contained in the controller files).
   - The controller(s) is/are located in the WEB-INF folder of each web application (e.g. C:\ofbiz\applications\product\webapp\catalog \WEB-INF\controller.xml)
5. Don‘t forget to add your application to the component-load.xml file (located e.g. in C:\ofbiz\hot-deploy)

→ OFBiz Demo

- Directory structure
- Important XML files
- Application composition

Coding Guidelines

1. Do NOT use minilang → use Java, or a Java scripting language such as Beanshell, JRuby, etc.
2. Do NOT use the form widget → use the OFBiz form macro or just Freemarker and HTML
3. Do NOT use constructs like <entity-and> or <entity-condition> in the screen widget → use Beanshell instead
4. Do NOT use the screen widget to layout your files → use an FTL template (Freemarker)
5. Do NOT use static Java methods in an FTL page → prepare a seperate *.bsh file and pass the result as list or map
6. Do NOT create new forms completely new → use existing FTL files as much as possible
7. Do NOT oversuse the service engine → write a Java method for just looking up data (it‘s much faster)
8. Do NOT oversuse transactions → add a *use-transaction="false"* to your services XML defiintion
9. Put in log messages whenever there is an unexpected behavior, e. g. Debug.logWarning(“Failure in Moudle", module);
10. Always create demo data for testing purposes.

See for further information:

- OFBiz Wiki: http://ofbizwiki.go-integral.com/Wiki.jsp
- Opentabs Home: http://www.opentaps.org/index.php
- OFBiz Apache site: http://ofbiz.apache.org/
- OFBiz documentation on opensourcestrategies: http://www.opensourcestrategies.com/ofbiz/
- OFBiz Data Model: https://ofbiz.dev.java.net/servlets/ProjectDocumentList?folderID=236&expandFolder=236&folderID=0

(a little bit confusing, isn‘t it?!)