validus

Integrated validation & consultancy

Enterprise Application Development

In Java with AJAX and ORM

ACCU London March 2010
ACCU Conference April 2010

Paul Grenyer
Head of Software Engineering
p.grenyer@validus-ivc.co.uk
http://paulgrenyer.blogspot.com

Validus
Paston House
Princes Street
Norwich
NR3 1AZ
www.validus-ivc.co.uk

Agenda

- About Me
- What is an Enterprise Application?
- Developing a Data Access Layer
- Integration Testing a Data Access Layer
- Developing an AJAX Presentation Layer with GWT
- Integrating Spring

About Me

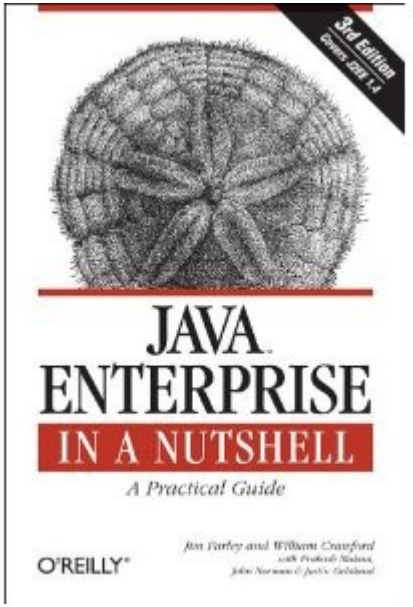
- ACCU Committee Member
- Programming for over 20 years (C++, C# & Java)
- Head of Software Engineering at Validus IVC Ltd.

What is an Enterprise Application?

Enterprise Computing

“At its heart, enterprise computing is all about combining separate applications, services and processes into a unified system that is greater than the sum of its parts.”

- Jim Farley et al. in Java Enterprise in a Nutshell

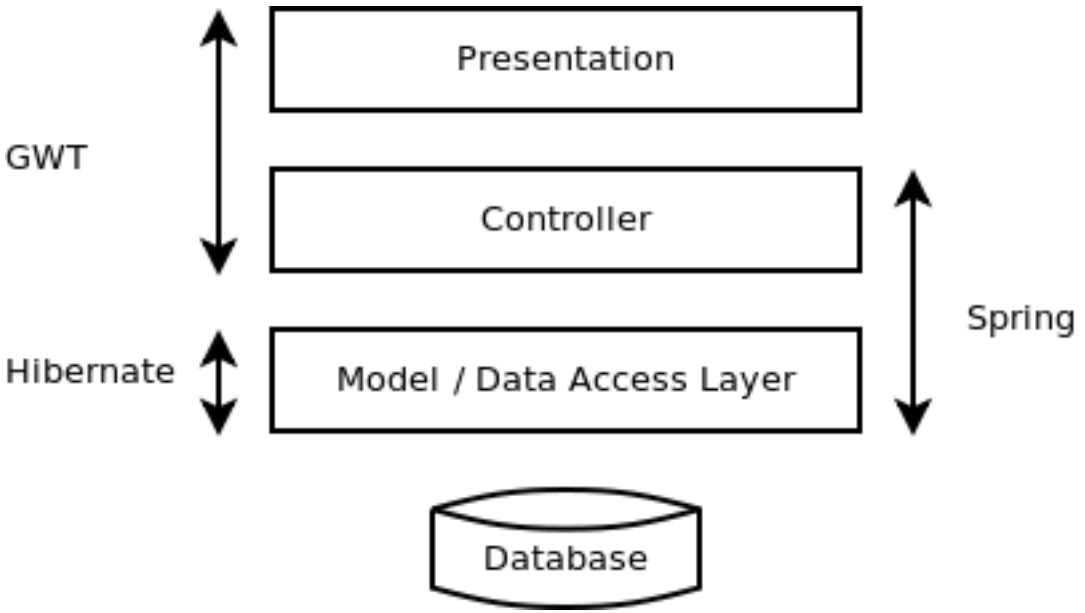


Enterprise Application

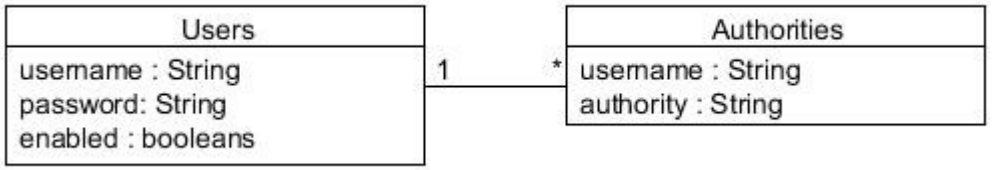
“An Enterprise Application is a combination of other applications, services and processes that, when unified, form a application that is greater than the sum of its parts.”

- Paul Grenyer

Enterprise Application Architecture



Developing A Data Access Layer



Creating the Database

- SQL Scripts
- Build script
- Database drivers (Ivy)

Persisting Users

- JDBC
- Hibernate
- HibernateTemplate

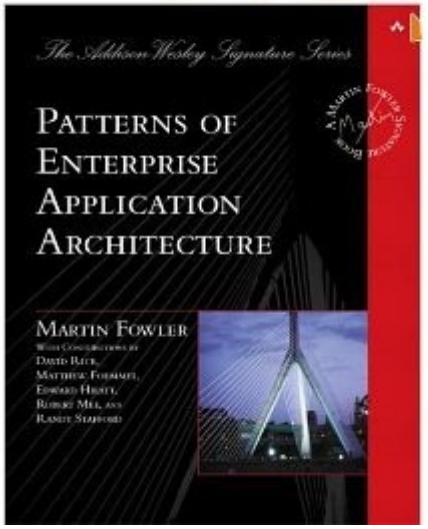
What is Hibernate?

Hibernate is a powerful **Object Relational Mapping** (ORM) persistence and query tool.

“Hibernate lets you develop persistent classes following object-oriented idiom - including association, inheritance, polymorphism, composition, and collections. Hibernate allows you to express queries in its own portable SQL extension (HQL), as well as in native SQL, or with an object-oriented Criteria and Example API.”

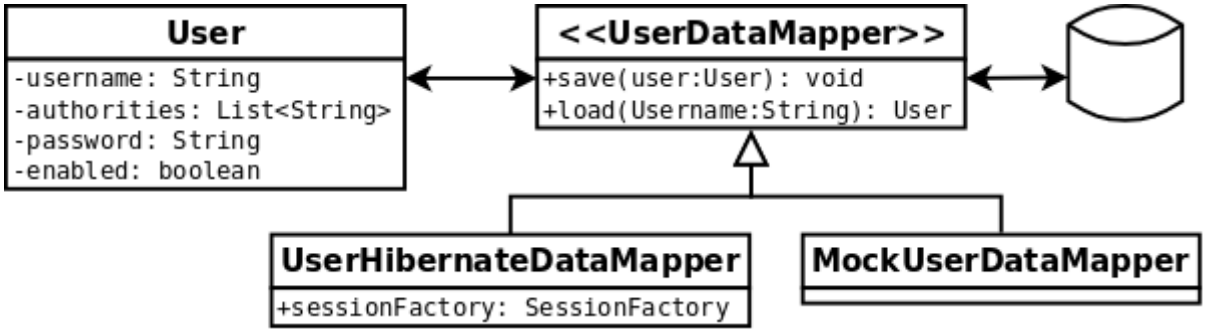
Is HibernateTemplate Enough?

- No:
  - It does not hide the client from the fact they are persisting the `User` object to a database.
  - Unit testing is difficult.



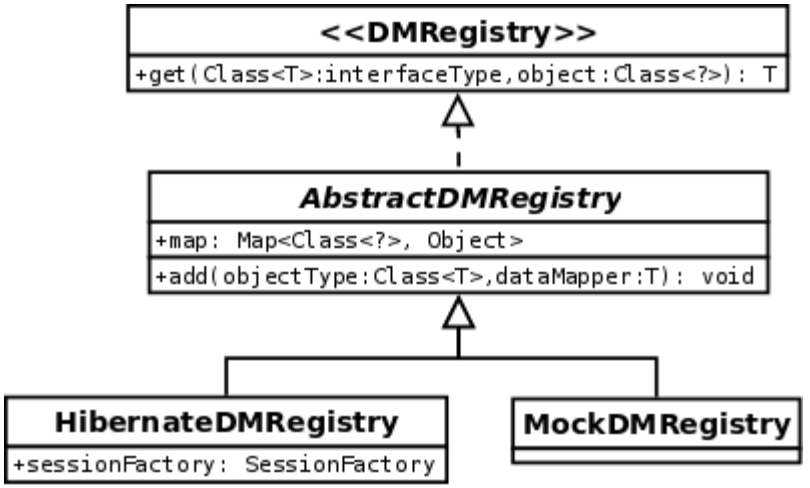
Data Mappers

A layer of mappers that moves data between objects and a database while keeping them independent of each other and the mapper itself.



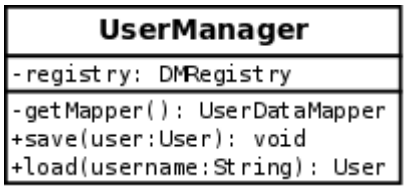
Registry

A well-known object that other objects can use to find common objects and services.

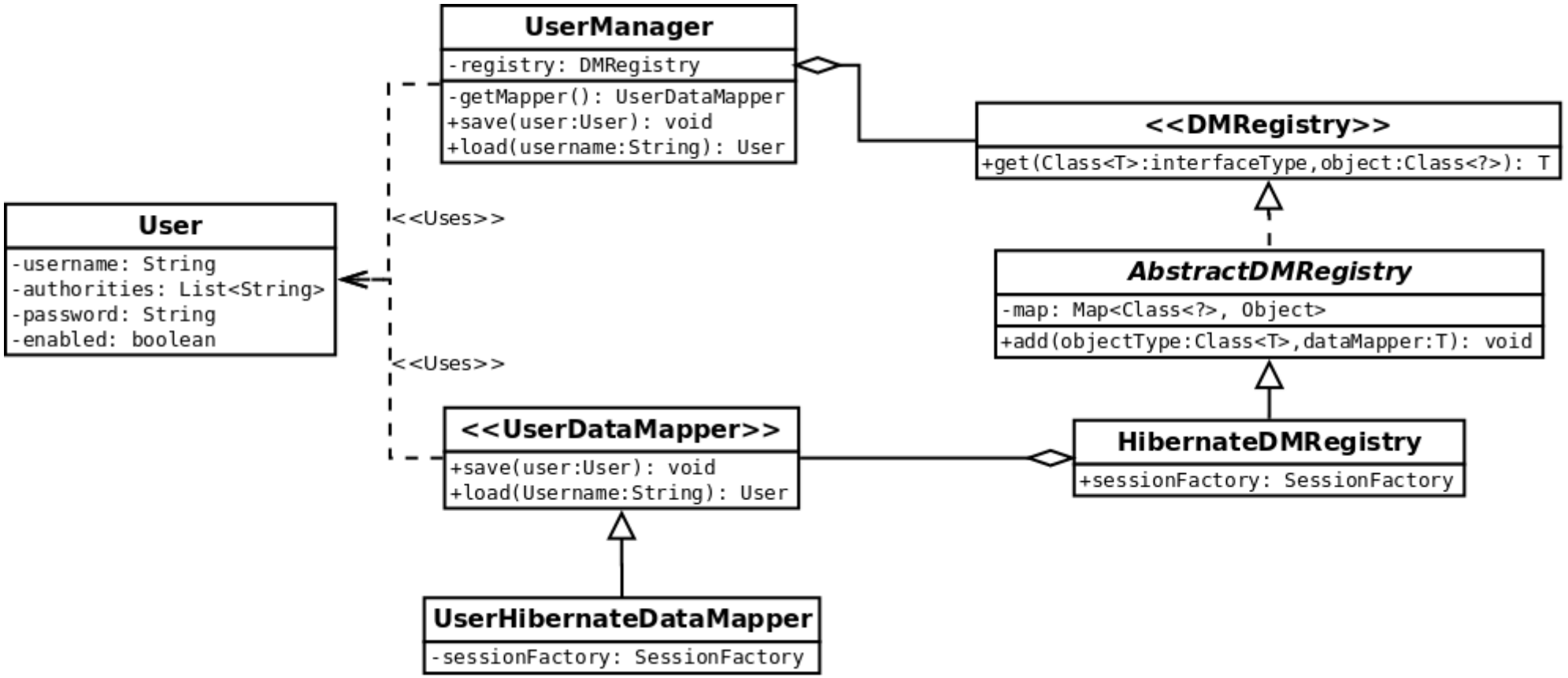


Managers

An object that simplifies persisting of objects.



Data Access Layer



Integration Testing a Data Access Layer

- What is Integration Testing?
- Transactions
- Writing Integration Tests

Google Web Toolkit

- Develop in Java, compile to Java Script
  - Use exactly the same Java classes on client and server sides
  - Unit test
- RPC
  - Single web page
  - Smaller round trips to the server
- Development Mode
  - Embedded Jetty
  - Run in browser with plugin

Google Web Toolkit Project Files

- Module XML Files
  - Inherited modules
  - An entry point application class name
  - Source path
- HTML Host Page
- Entry Point Class
- WAR Directory
  - Static content you provide, such as the host HTML page
  - GWT compiled output
  - Java class files and jar files for server-side code
  - A web.xml file that configures your web app and any servlet

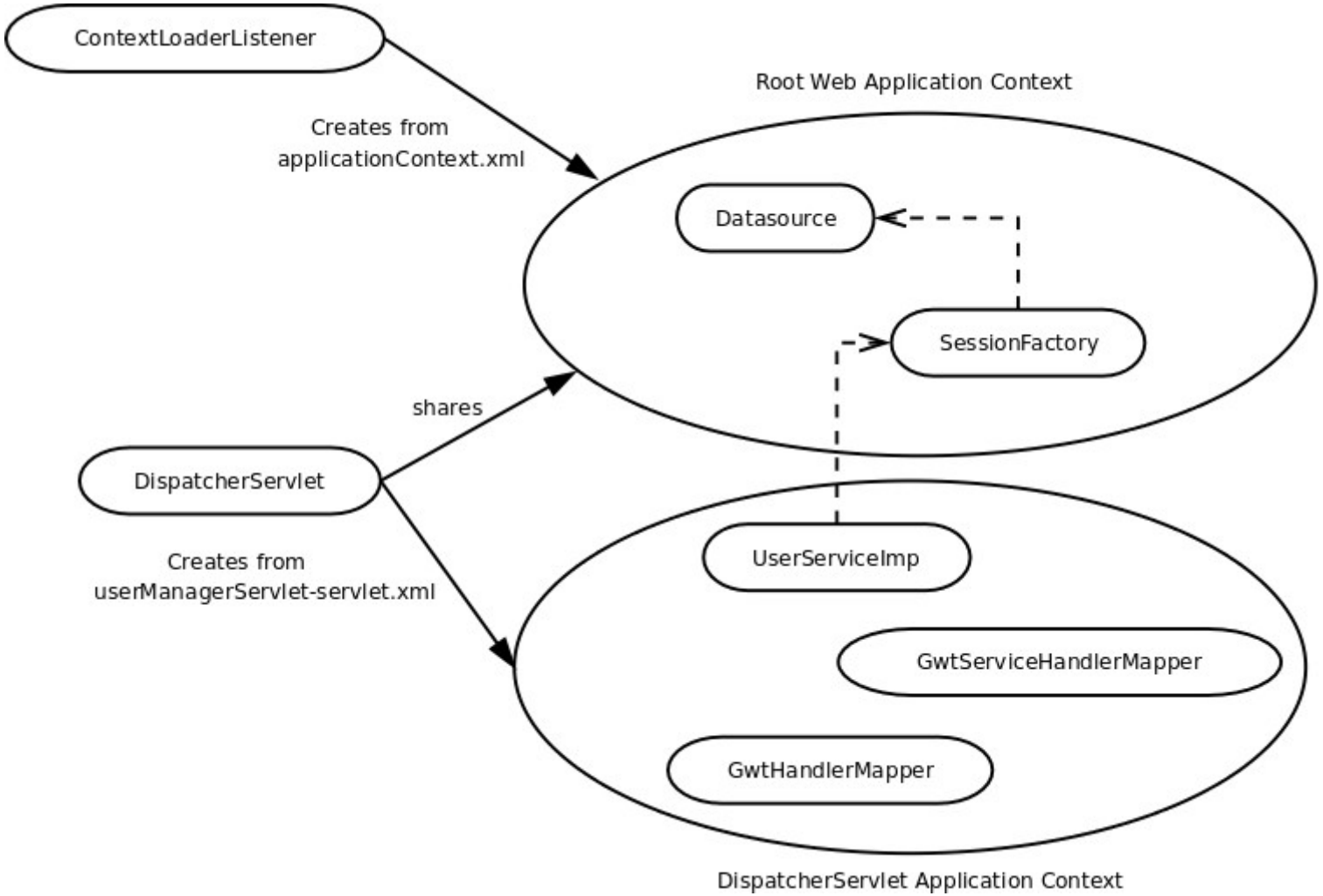
GWT User Manager Application

- Widgets:
  - User name Text Box
  - Find Button
  - Save Button
  - Password Text Box
  - Enabled Check Box
  - Authorities List Box
  - Authorities Text Box
  - Add Authority Button
  - Remove Authority Button
- Grid Layout

Introducing Spring

- Enterprise Application Framework
- Developed by Rod Johnson as a lightweight alternative to EJB
- Loosely coupled
  - Annotations
  - XML
- Spring Container / Application Context
- Dependency Injection (Inversion of Control)
- Aspect Orientated Programming
- Database support
- Transactions
- Security
- Testing support

Spring Application Context



What is a Spring Bean?

- A bean that is managed by the Spring Container:
  - Plain Old Java Object (POJO)
  - **Take advantage of Dependency Injection**
  - Take advantage of Spring Aspect Orientated Programming (AOP)
  - Take advantage of Spring Transaction Management
  - ...

Springifying an Application

- Instantiate a `ContextLoaderListener`
  - Loads `applicationContext.xml` and root creates the web application context.
  - Web application context manages database connection pools and Hibernate session factory.

- Instantiate a DispatcherServlet
  - Loads `userManagerServlet-servlet-xml` and creates a second application context which has access to the root application context.
  - Maps RPC calls onto handlers.

GwtHandlerMapping

- Implements `AbstractDetectingUrlHandlerMapping`
- At start-up, Spring passes the name of every Spring bean to the `determineUrlsForHandler` method
- If the Spring bean implements `RemoteService`:
  - It is considered to be a GWT RPC endpoint
  - Its name is used to determine its URL
- Reflection only used at application start-up – so no performance overhead

GwtServiceHandlerMapper

- Spring's `DispatcherServlet` passes RPC calls to Spring beans that implement `HandlerAdapter` to process.
- The `handle` method caches the incoming RPC request *handler* and calls `doPost`.
- `processCall` call gets called as a result of calling `doPost` and:
  - Takes the incoming RPC call
  - Decomposes the incoming RPC call
  - Invokes the relevant method on the *handler*

Finally

- What an Enterprise Application is.
- Developed a Data Access Layer
- Integration Tested a Data Access Layer
- Developed an AJAX Presentation Layer with GWT
- Integrating Spring
- Any questions?

See you in the bar!