

Java & Persistence

More than just aspect ?

Me & Myself

- Werner Guttman
- Software architect @ ANECON (<http://www.anecon.com>)
- Committer at [http://castor.\(codehaus.\)org](http://castor.(codehaus.)org) (3+ yrs), both for ...
 - Castor XML (XML data binding)
 - Castor JDO (O/R mapping, persistence)
 - Release management, issue management
 - Professional services (investment bank)

Agenda

- Scope
- Questions
- (XML, Relational) Data binding – An aspect
- Separation of concerns
- Domain modelling
- Open source
- Summary
- Q&A

Agenda

- **Scope**
- Questions
- (XML, Relational) Data binding – An aspect
- Separation of concerns
- Domain modelling
- Open source
- Summary
- Q&A

Agenda (2) – What it is **not**

- Technical introduction
- Comparison of various O/R mapping frameworks
- Comparison of open source and commercial O/R mapping frameworks
- Detailed instruction on HOW-TO achieve

Agenda (3) - Goals

- (Historical) context definition
- Understanding of the evolution of persistence (frameworks)
- Anecdotal overview over market, vendors and specification(s)
- Insight into main drivers behind (more) recent developments, mainly Spring ORM and JPA
- Relevance of OPEN SOURCE
- Spirit of OPEN SOURCE (development)

Agenda

- Scope
- **Questions**
- (XML, Relational) Data binding – An aspect
- Separation of concerns
- Domain modelling
- Open source
- Summary
- Q&A

Questions – How many

- ... are using **XML** ?
- ... are using **RDBMS** ?
- ... are using **JDBC** ?
- ... are using an **O/R mapping** tool ?
- ... are using **Hibernate** ?
- ... are using **Spring ORM** ?
- ... actually know the term **JPA** ?
- ... plan to **migrate** to JPA ?

Agenda

- Scope
- Questions
- (XML, Relational) Data binding – An aspect
- Separation of concerns
- Domain modelling
- Open source
- Summary
- Q&A

Data binding – A Definition

- Castor product definition:
It's the shortest path between Java objects and other representations of the same data.
- Wikipedia.org (modified):
Data binding refers to the process of representing the information stored in an object in computer memory in other (well-defined) forms. This allows applications to access data stored in a variety of formats from the object (rather than using a presentation-specific API to retrieve data from the representation itself).

XML Data binding – A Definition

- Castor product definition:
It's the shortest path between Java objects, XML documents and relational tables.
- Wikipedia.org:
XML data binding refers to the process of representing the information in **an XML document** as an object in computer memory. This allows applications to access the data in the **XML** from the object rather than using an **XML-specific** API to retrieve the data from a direct representation of the XML itself.

XML Data Binding – A definition (2)

- When this process is applied to convert an XML document to an object, it is called **unmarshalling**.
- The reverse process, to serialize an object as XML, is called **marshalling**.

Prerequisite: A **mapping** between elements of the XML document and members of a class to be represented in memory.

O/R mapping – A Definition

- Wikipedia.org:
Object-Relational mapping is a programming technique that **links databases to object-oriented language concepts.**
- Relational data binding (o/r mapping) refers to the process of representing information stored **in a relational database** as objects in computer memory. This allows applications to access data in the database from objects rather than using a **database-specific API** to retrieve data from its representation in the database.

Agenda

- Scope
- Questions
- (XML, Relational) Data binding - An aspect
- **Separation of concerns**
- **Domain modelling**
- Open source
- Summary
- Q&A

Separation of Concerns

- Avoid intrusion/intrusiveness (JDBC)
- Separate application aspects into isolated areas, and fine-tune these unrelated areas individually
- Sample aspects:
 - Persistence, caching
 - XML data binding, web services
 - Domain objects
 - Application/business logic
- Use existing and tested tools/frameworks to implement these aspects

Modelling aspects

In today's tool landscape, there's an ever growing support for modelling aspects, e.g. as part of an UML model

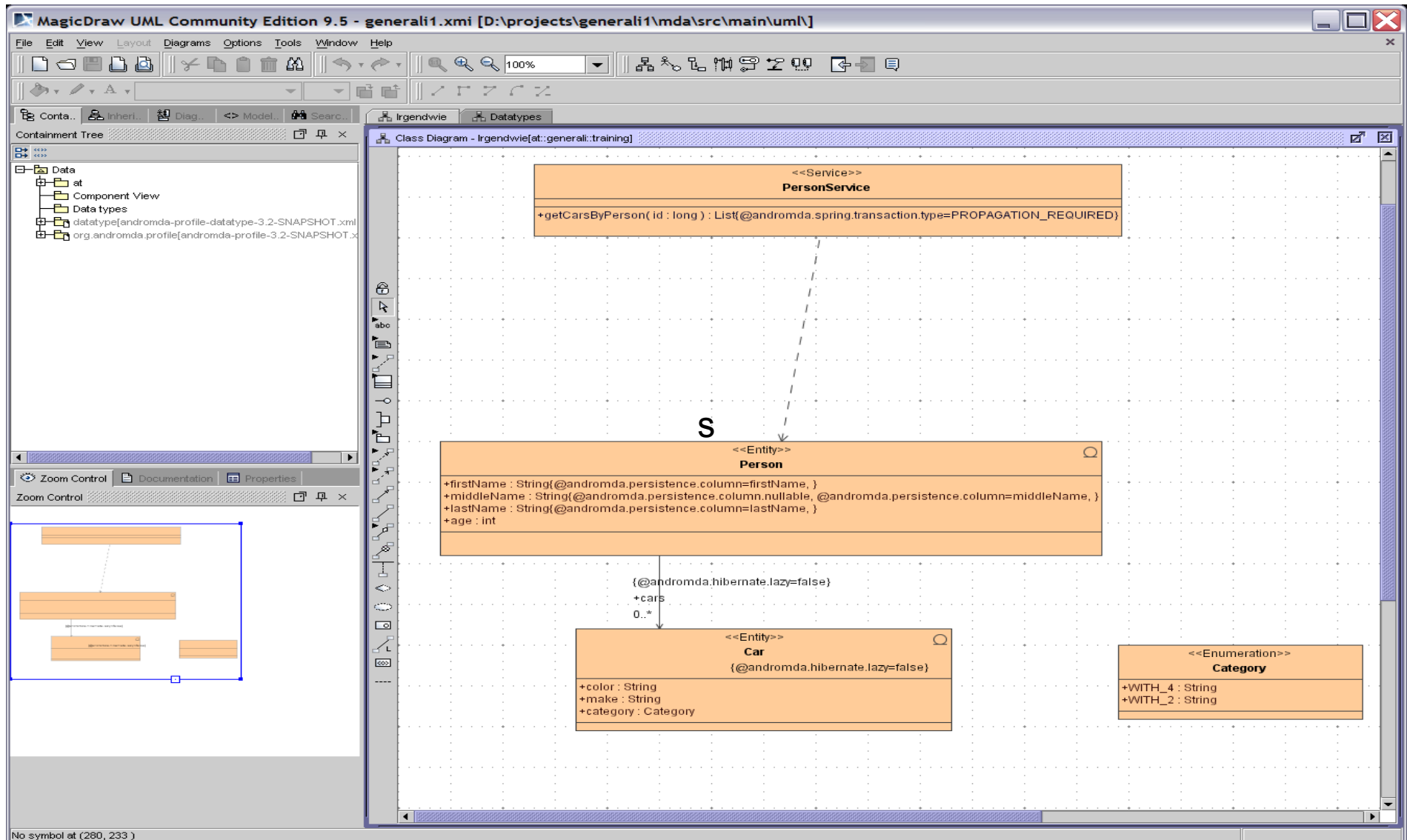
- UML profiles for persistence (e.g. Hibernate, Spring ORM using Hibernate)
- UML profiles for web services (e.g. AXIS), where AXIS has support for integrating with XML data binding tools (JAXB, Castor, ...)

Prerequisite: extract the aspects from the application code (e.g. business logic) and keep them separatedly.

Modelling aspects (2)

- Once we look at persistence, services, etc. as **aspects** of our application (model), we can move into (partial) **code generation**
- Possible generated artefacts:
 - Domain objects
 - Service interfaces (+ Spring configuration files, including transaction demarcation, Spring AOP)
 - Hibernate mapping file (+ Spring ORM setup)
 - Web service stubs/skeletons (AXIS)
 - (XML Schemas)

Modelling aspects (3)



Agenda

- Scope
- Questions
- (XML, Relational) Data binding - An aspect
- Separation of concerns
- Domain modelling
- **Open source**
- Summary
- Q&A

Castor

- Castor product definition:

It's the shortest path between Java objects, XML documents and relational tables.

Castor – Support for ... and

- XML Data Binding framework (7+ yrs)
Marshalling, unmarshalling; code generation from XML schema documents
 - widely used in various industry sectors (financial, ..)
 - Currently not JAXB 2.0-compliant (though work scheduled)
 - Integration with many WS frameworks (AXIS, XFire)
- Persistence framework (Castor JDO, 5+ yrs)
 - Spring ORM support (milestone release)
 - Implementation of JPA 3.0 support started

Spirit by Example - Castor

- 500.000 lines of code
- Extensive test suites (covering both persistence and XML data binding)
- Code coverage more than 65% (in some areas more than 80%)
- 6+ yrs on the market
- Extensive user base (financial sector, IBM, Tsunami ...)
- Professional services

Spirit by example – Castor (2)

- NYC/London-based **investment bank**

FpML (industry standard for financial transactions related to derivative products, set of XML schemas) --> code generation --> 1.450 domain classes
--> **Design by contract**

- **IBM**

Internal tool support for QWest

Agenda

- Scope
- Questions
- (XML, Relational) Data binding - An aspect
- Separation of concerns
- Domain modelling
- Open source
- **Summary**
- Q&A

Summary

- Avoid intrusion/intrusiveness
- Separate application aspects into isolated areas
- Aspects can be (more and more) modelled in tools in sufficient detail
- Use (tested) tools/frameworks, avoid home-growing solutions
- Make use of the 80/20 rule
- Use an abstraction layer to encapsulate tool decision --> Spring ORM

Agenda

- Scope
- Questions
- (XML, Relational) Data binding - An aspect
- Separation of concerns
- Domain modelling
- Open source
- Summary
- **Q&A**

Questions (2) – How many

- ... use persistence tools in **JEE clusters** ?
- ... use persistence frameworks with (commercial) **caches** enabled ?
- ... use **caches in a distributed environment** (e.g. JEE cluster)
- ...

The End

Contact:
wguttmn AT codehaus DOT org