Java Persistence Landscape

Craig Russell

Sun Microsystems, Inc.

Agenda

- Java Persistence Alternatives
- Persistence Solution Strategies
- Religious Issues
- Recommendations
- Conclusion
- Q & A

ApacheCon Europe 06

Java Persistence Alternatives

- Hand-coded JDBC
  - JDBC 4.0 DataSet
  - Data Access Object Pattern
  - Active Record Pattern
- Container Managed Persistence
- POJO Persistence
  - Data Mapper Pattern

ApacheCon Europe 06

Hand Coded JDBC

- JDBC4 DataSet Usage

```
public class Employee {
  public String name;
  public int number;
}

interface EmployeeQueries extends BaseQuery {
  @Select(sql="SELECT name, number from EMPLOYEE" +
    "where NUMBER EQ {number}")
  DataSet<Employee> getByNumber(int number);
}
```


ApacheCon Europe 06

Hand Coded JDBC

- JDBC4 DataSet Usage

```
Connection conn = getConnection();
EmployeeQueries queries =
  conn.createQueryObject(EmployeeQueries);
DataSet<Employee> emps = queries.getByNumber(empNo);

for(Employee emp:emps) {
  System.out.println("Employee id: " + emp.number +
    "name: " + emp.name);
}
```


ApacheCon Europe 06

Hand-coded JDBC

- DataSet is Useful for:
  - Eliminating Trivial ResultSet Handling
  - SQL Oriented Applications
  - Trivial Domain Models
    - No Relationships
    - No Navigation
- DataSet Doesn't Help with:
  - Transaction / Connection Handling
  - Complex Domain Models

ApacheCon Europe 06

POJO Persistence

- Java Data Objects
- Hibernate
- TopLink
- EJB3

Java Data Objects

- JCP Specification (not a product)
- Apache Licensed Products Available
- Transparent Persistence
- Integration With Application Servers
- Standardized Object-Relational Mapping
- Java Language-like Query

ApacheCon Europe 06

Hibernate

- “Commercial Open Source” Product
- LGPL License
- “Relieve developer of 95% of common programming tasks compared with SQL and JDBC”
- “Portable to all SQL databases”

ApacheCon Europe 06

TopLink

- Commercial Product – Oracle License
- TopLink Essentials: CDDL License
- "Highly flexible and productive mechanism for storing Java objects and EJBs into relational databases"
- "Excellent performance and choice"

EJB3

- JCP Standard (not a product)
- Reference Implementation CDDL License
- Apache License Implementation Available*
- “Provide an object-relational facility for the Java programmer”
- “Use a Java domain model with a relational database”

* any day now

EJB3

Now that EJB3 is shipping, isn't everything else going away?

EJB3

- In a word, no...
  - EJB3 is “best of” (JDO, TopLink, Hibernate)
  - No published plans for any of these technologies to disappear
  - EJB3 is not the place for innovation, but consolidation of accepted ideas
  - Features will be added to EJB3 only after adoption by the community

Agenda

Java Persistence Alternatives
Persistence Solution Strategies
Religious Issues
Recommendations
Conclusion
Q & A

ApacheCon Europe 06

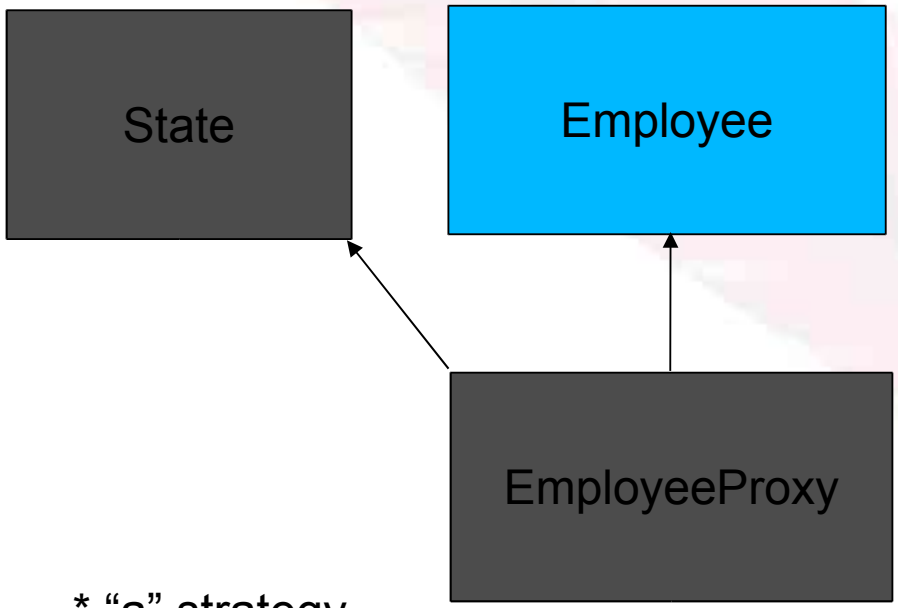
Common Strategies

- POJO Model
- Factory Pattern for Persistence Manager
- Map<Identity, PersistentInstance>
- XA Integration (Distributed Transactions)

ApacheCon Europe 06

Hibernate Strategy*

- Persistent Methods
- Subclass Proxy



Application-visible class instances have no state

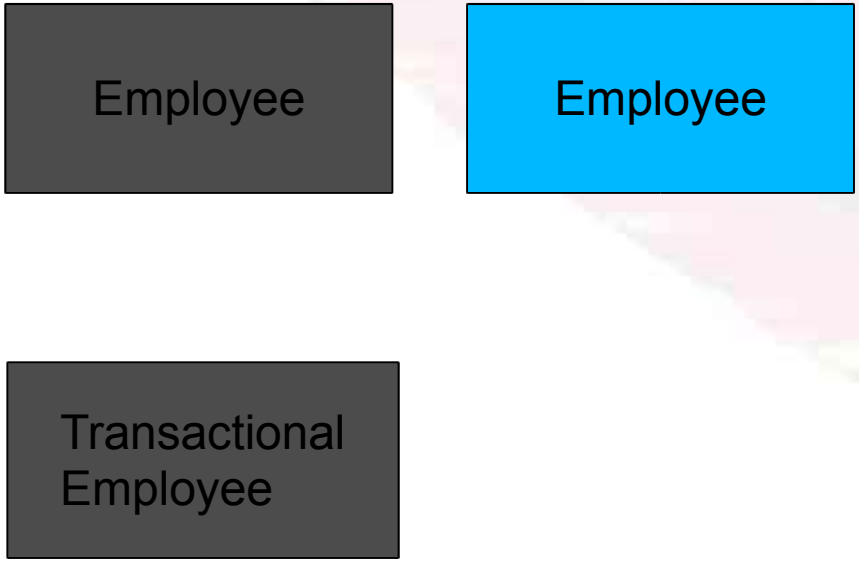
Proxy implements persistent methods, stores state in persistent state instance

* “a” strategy

ApacheCon Europe 06

TopLink Strategy

- Persistent Fields
- Clone Shared Cache Instances



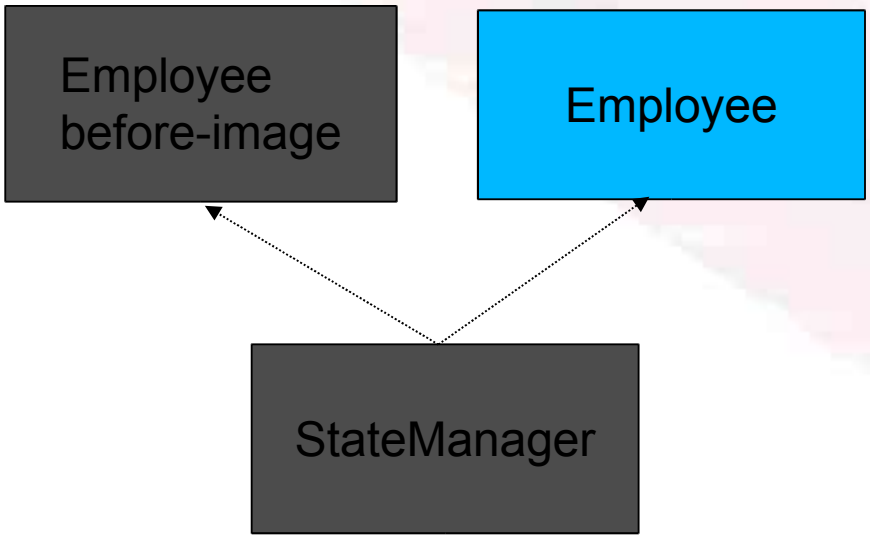
Application-visible instances have shared persistent state

Transactional (modified) persistent state

ApacheCon Europe 06

JDO Strategy

- Persistent Fields
- Copy on Change *



* “a” strategy

Application-visible instances have persistent state, managed by persistence manager

Modified instances might use before-image state instance

ApacheCon Europe 06

Agenda

Java Persistence Alternatives
Persistence Solution Strategies
Religious Issues
Recommendations
Conclusion
Q & A

Religious Issues

- Persistent Field vs. Method
- Visible Identity Field
- Focus
  - Store Objects in Databases
  - Represent Relational Rows as Objects
- Byte-code Enhancement
- Query Language
- Instantiating Object Graphs for Performance

ApacheCon Europe 06

Persistent Field vs. Method

- Persistent Method
  - BigDecimal getSalary() {return sal;}
  - setSalary(BigDecimal salary) {this.salary = sal;}
- Values set by persistence manager (using set method)
- Methods are shared by persistence manager and application
- Restricts validation logic
  - Single field, range or value only

ApacheCon Europe 06

Persistent Field vs. Method

- Persistent Field
  - BigDecimal salary;
  - Values set directly by persistence manager (via reflection or byte-code enhancement)
  - Allows arbitrary field validation logic
    - setters
    - getters
    - business methods

ApacheCon Europe 06

Visible Identity Field?

- Identity Issues
  - Java Identity (z == b)
  - Java Equality (z.equals(b))
  - Persistent Identity
    - Visible to Application (getId())
    - Managed by Persistence

ApacheCon Europe 06

Focus: Object or Relational Row?

- Plain Old Persistent Object
  - Persistent Class Restrictions
  - Persistent Field Restrictions
  - Persistent Method Restrictions
  - Mapping Restrictions

ApacheCon Europe 06

Byte-code Enhancement

- Rationale
  - Intercept Field Access Byte-codes
    - Just In Time Field Instantiation
    - Change Detection without Value Comparison
- Standardized by JDO 1
  - Before Aspect-Oriented Computing
- Vilified by Competition
- Now Accepted by All Persistence Vendors
  - “Byte-Code Weaving”

ApacheCon Europe 06

Query

- High-level Bit:
  - Java language-like or SQL-like?
- Static or Dynamic?
- String or API?
  - a.k.a. Criteria Query
- Multiple Query Language Support?

ApacheCon Europe 06

Performance Issues

- Change Detection
- Many Objects and Fields Instantiated in a Transaction: Flush only Changes to Database
  - Byte-code Enhancement is Lowest Overhead
  - Sub-classing with Persistent Method is Close
  - Graph Comparison with Reflection is Far Behind

ApacheCon Europe 06

Performance Issues

- Instantiate the Right Object Graph
  - Minimize Round Trips to Database
  - Get All Objects/Fields Needed and No More
- Alternatives:
  - Crude: Eager/Lazy Loading
  - Brute Force: Add Query Expressions
    - Every Use-Case Needs New Queries
  - Orthogonal: Fetch Groups and Fetch Plans

ApacheCon Europe 06

Agenda

Java Persistence Alternatives
Persistence Solution Strategies
Religious Issues
Recommendations
Conclusion
Q & A

Recommendations

- Understand Your Own Biases
- Document Key Requirements
  - Today's Projects
  - 2-3 Years (One Internet Generation) in Future
- Evaluate Architectures
  - Paper Comparison
  - Representative Application "Benchmark"
    - Productivity
    - Performance

Agenda

Java Persistence Alternatives
Persistence Solution Strategies
Religious Issues
Recommendations
Conclusion
Q & A

Conclusion

- POJO Persistence is Here to Stay
- Persistence Architectures Vary Widely
- Common Features Address >50% of Applications
- Consider Writing a Persistence Abstraction

ApacheCon Europe 06

Q & A

http://java.sun.com/javaee/overview/faq/ejb_persistence.jsp

http://www.hibernate.org

http://jpox.org

http://www.oracle.com/technology/products/ias/toplink/index.html