

# E6: Datenbankzugriff mit Java, Heute und in der Zukunft

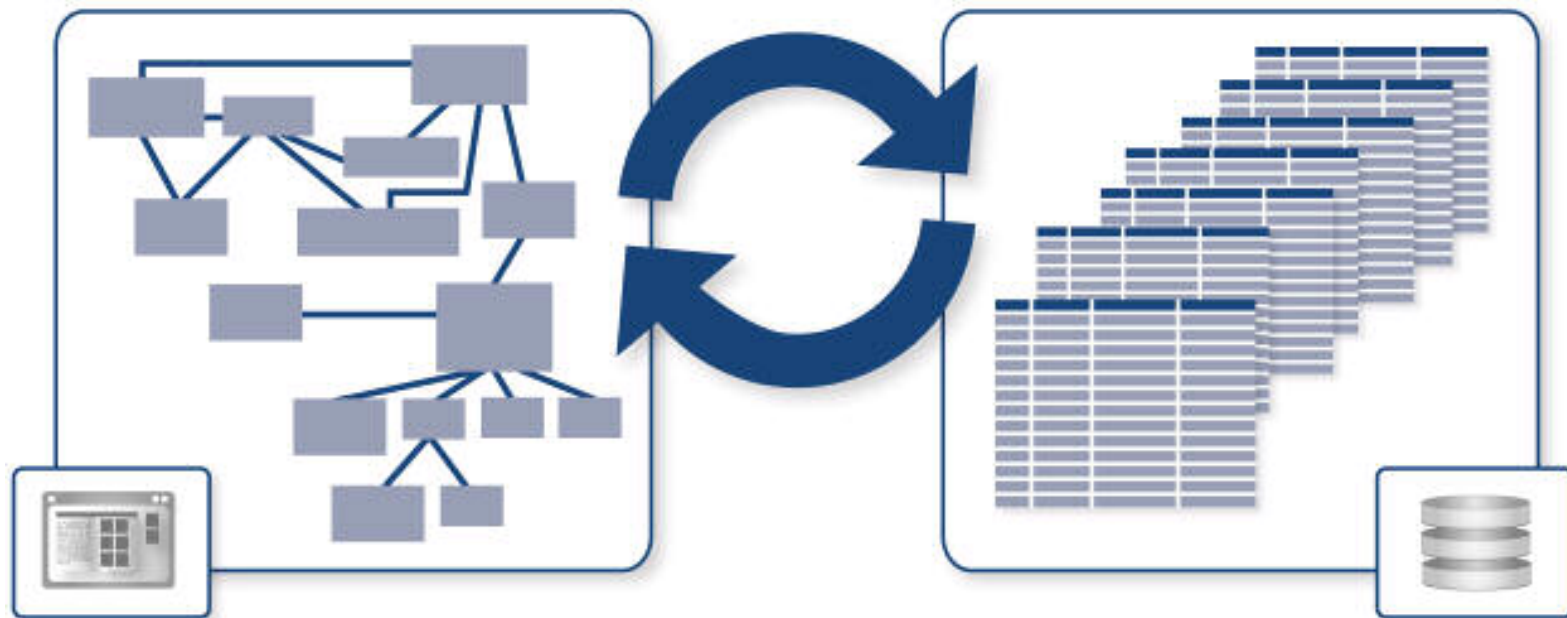
Martin Wessel  
Versant GmbH

**Copyright © Versant 2005.**

All products are trademarks or registered trademarks of the respective companies.  
The information contained in this document is property of Versant.

# Die Herausforderung

- OO- Entwicklung  $\leftrightarrow$  RDBMS



- Verschiedene Modelle
- Mapping wird benötigt

# Mögliche Lösungen, Heute

- JDBC
  - Erster Java Standard zum Zugriff auf RDBMS
    - Basistechnologie
  - “Impedance Mismatch” bleibt
  - Implementierung und Wartung sind aufwendig
  - Bindung an das Datenbanksystem
- Mappingtools
  - Bindung an den Tool-Hersteller

# Mögliche Lösungen, Heute

- EntityBeans - EJB 2.1
  - Keine objektorientierten Konzepte
  - Aufwendige Implementierung und Test
- Java Data Objects – JDO 1.0.1
  - Erster Standard für transparente Persistenz
  - RDBMS, ODBMS, XML
  - J2EE, J2SE, J2ME

# Die Java Persistence API

- EJB 3.0 (JSR 220)
  - EntityBeans sind jetzt POJOs
  - Vereinigt verschiedene Standards und Technologien
    - TopLink
    - JDO
    - Hibernate
  - Ist nicht auf J2EE beschränkt
    - Separates Dokument für das Java Persistenz API

# Vergleich: EJB 3.0 - JDO

- EJB 3.0
  - Java annotations (JDK 1.5)
  - XML metadata (JDK 1.3, 1.4)
  - EntityManager
    - persist
    - merge
    - close
    - createQuery
  - EJBQL, SQL
- JDO
  - XML metadata (JDK 1.3, 1.4, 1.5)
  - PersistenceManager
    - makepersistent
    - attach
    - detach
    - newQuery
  - JDOQL, SQL

# Das aktuelle Projekt?

- Immer Standards einsetzen
  - Heute mit JDO starten
    - Schon jetzt EJB 3.0 Migration planen
  - EntityBeans EJB 2.1 nicht mehr neu anfangen
    - SessionBeans mit JDO, Spring, ...
- Tools
  - Manche Tools unterstützen schon heute beide Standards
    - Eclipse JSR220-ORM
      - PersistenceManager (JDO), EntityManager (EJB 3.0)
      - JDOQL, EJBQL, SQL



# Eclipse JSR220-ORM

- Aktuell in „proposal“ Phase
  - <http://www.eclipse.org/proposals/eclipse-jsr220-orm/index.html>
  - [http://www.versant.com/opensource/index\\_html](http://www.versant.com/opensource/index_html)
- “Versant Open Access” ist jetzt Open-Source
  - Eclipse tools und runtime (EPL)
  - Entwickler:
    - Versant Open Access Team
- Unterstützt alle Java Persistenzstandards
  - JDO 1.0.1
  - JDO 2.0 (JSR243)
  - EJB 3.0 (JSR 220)



# Screenshot

The screenshot displays the Eclipse IDE's 'VOA JDO - Edit VOA Mapping' window. The left sidebar shows a project structure with 'JDOBio' containing various biological entities like 'Chromosome' and 'ChromosomeFrag'. The main area is titled 'Chromosome.chromosomeFragments java.util.LinkedList'. It features a 'Mapping' section with radio buttons for 'Link table', 'One-to-Many', 'Many-to-Many', 'Externalized', 'Not persistent', and 'Transactional'. Below this, the 'Element type' is set to 'ChromosomeFrag'. A diagram illustrates the relationships: 'Chromosome' (with 'chromosome\_id INTEGER') is linked to 'ChromosomeFragments' (with 'chromosome\_id INTEGER', 'seq INTEGER', and 'chromosome\_frag\_id INTEGER') via an 'OUTER' relationship. 'ChromosomeFragments' is further linked to 'ChromosomeFrag' (with 'chromosome\_frag\_id INTEGER') via another 'OUTER' relationship. The bottom of the window includes options for 'Delete objects in collection when owning Chromosome is deleted', 'Collection / Array ordering' (Unordered, Preserve index, JDOQL ordering), 'Fetch group to load when field is touched' (None, Default, Custom), 'SCO factory' (SCOLinkedListFactory com.versant.core.jdo.sco), and 'Collection' (JDO XML).

# Integration in andere Eclipse Projekte

- Data Tools Project (DTP)
  - Database Schema Model
  - Connection Management
  - SQL query editor
- Web Tools Project (WTP)
- Eclipse Modeling Framework (EMF, GMF)
- Integration in Applikationsserver

# JSR220-ORM besondere Features

- Optimale SQL Queries
  - Torpedo Benchmark: Zählt SQL statements für verschiedene Operationen
  - <http://www.torpedo-group.org>
- Codegenerierung zur Laufzeit
  - „Hyperdrive classes“

# Live Demo

- eclipse JSR220-ORM plug-in



# Was Sie mitnehmen sollen!

- Transparente Persistenz
  - Unabhängigkeit
    - Datenbank
      - RDBMS, ODBMS, XML
    - Software Architektur
      - J2EE, J2SE, J2ME
  - Geschwindigkeit
    - in der Entwicklung
    - im Test
    - in der Anwendung



# Die Frage nach der Datenbank ist wieder offen?

- Existiert die Datenbank bereits?
  - Sind Änderungen am Schema möglich?
- Gibt es Vorgaben bzgl. Reporting?
  - Ist das Reporting in die Applikation eingebunden?
- Wer kontrolliert die Datenbank?
  - DBA – Entwickler
- Ist die Datenbank für das Modell geeignet?
  - Performance

# Fragen?