

E6: Datenbankzugriff mit Java, Heute und in der Zukunft

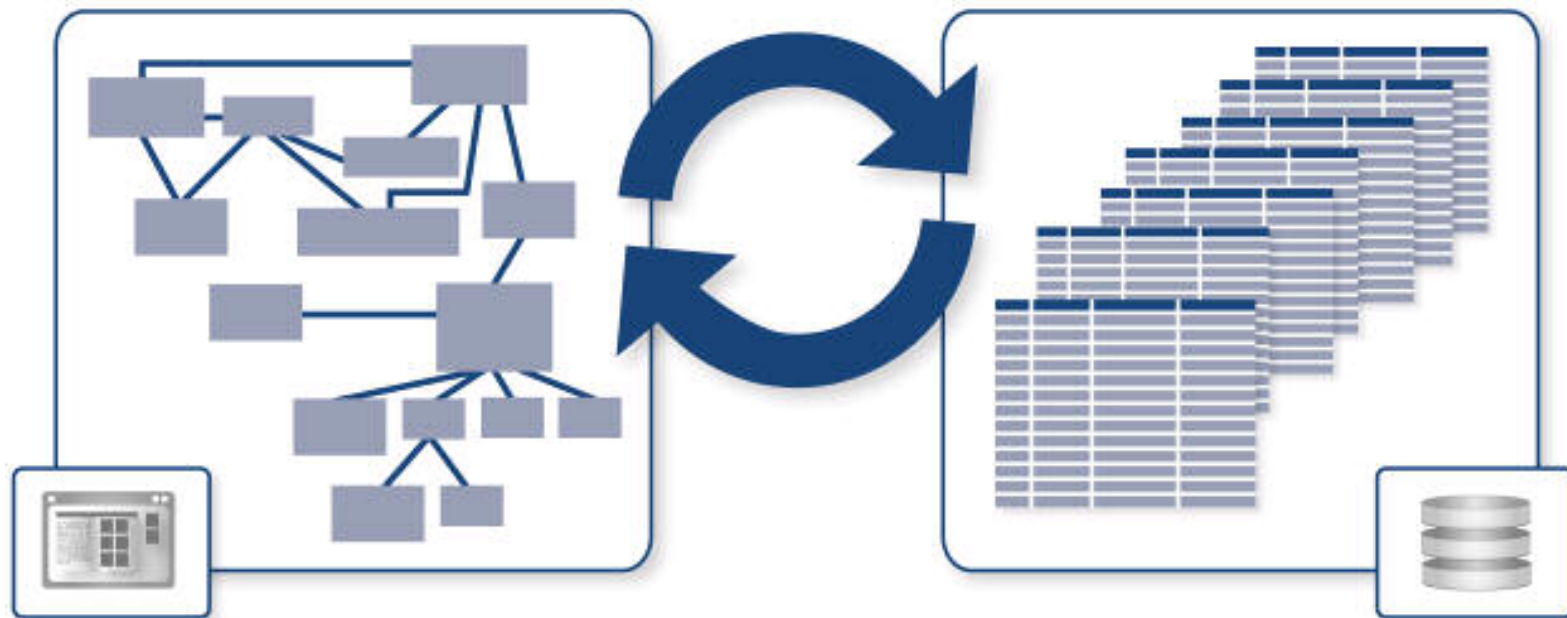
Martin Wessel
Versant GmbH

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
- “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 interface for editing VOA mappings. The main window is titled "VOA JDO - Edit VOA Mapping - Eclipse Platform". The left sidebar shows a project explorer with a tree view containing folders like "Eviction", "Finance", "ForwardEng", "Genomatica", "IMCLONE", and "JDOBio". Under "JDOBio", there is a "bioinformatics" folder containing various classes such as "Centromere", "ChromosomalElement", "Chromosome", "ChromosomeFrag", "ChromosomeFrag", "Codons", "Gene", "Genome", "Intron", "NonTransRegion", "OpenRFrame", "OriginsOfRepl", "Polypeptide", "PrimaryTranscript", "Promoter", "Protein", "RNAm", and "DMA".

The main configuration 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". The "Element type" is set to "ChromosomeFrag".

Below the mapping options, there is a class diagram showing three classes:

- chromosome** (Chromosome): Contains a field "chromosome_id INTEGER".
- chromosome_chromosome_frag** (chromosomeFragments): Contains fields "chromosome_id INTEGER", "seq INTEGER", and "chromosome_frag_id INTEGER".
- chromosome_frag** (ChromosomeFrag): Contains a field "chromosome_frag_id INTEGER".

 Arrows labeled "OUTER" indicate relationships between the classes.

Additional configuration options include:

- A checkbox for "Delete objects in collection when owning Chromosome is deleted".
- "Collection / Array ordering" with radio buttons for "Unordered", "Preserve index", and "JDOQL ordering".
- "Fetch group to load when field is touched" with radio buttons for "None", "Default", and "Custom".
- "SCO factory" set to "SCOLinkedListFactory com.versant.core.jdo.sco".
- "Collection" set to "JDO XML".

Integration in andere Eclipse Projekte

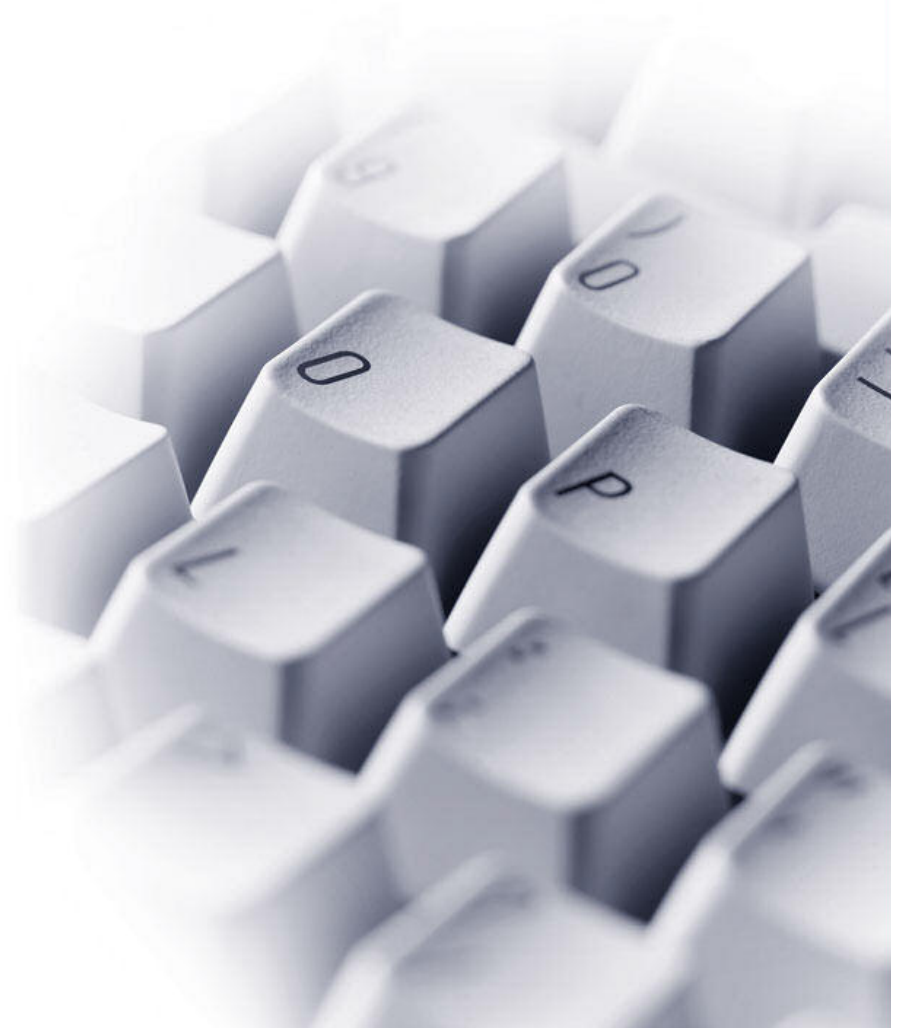
- Data Tools Project (DTP)
 - Database Schema Model
 - Connection Management
 - SQL query editor
- Web Tools Project (WTP)
- Eclipse Modeling Framework (EMF, GMF)
- Integration in Applikationsserver

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?