

JAVA DAYS '98



Java™ Blend™ - Integrating Java Objects with Enterprise Data

Dr. Gerhard Müller-Pröfrock
Geschäftsführer
Tech@Spree Software Technology GmbH





The Situation

- Enterprise data are the real asset of any business
- Enterprise data are kept in RDBMS at best
- RDBMS market place is still expanding
- RDBMS will dominate the market place for the future
- RDBMS is *proven technology*

BUT



The Situation

- Use of *object technology* is equally growing
- Most new IT projects are now started with object technology
- Especially Java will make its way in enterprise applications!

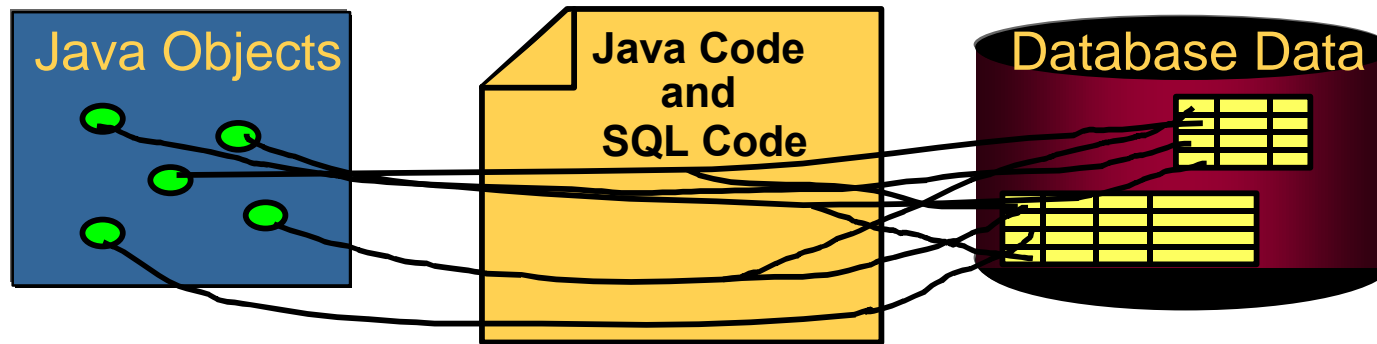


Usage Scenarios

- **Use Java as a front-end to present parts of the enterprise data via Internet (e-catalogs, e-commerce)**
- **Use Java as a computational model for new Intranet applications based on enterprise data**
- **Use Java for building completely new applications creating new enterprise data**



Database Applications Today



Impedance Mismatch

- Data *manually converted* from Java object-oriented representation to database representation and back
- Programmer must *split code* into Java portion and SQL portion, must understand both



Programmer s Burden

- **Dismantle complex object structures into flat relations**
- **Map object handles to foreign keys**
- **Map inheritance to joins or internal tables**
- **Use SQL and write code to do the mapping transparently, write code to allow navigation**
- **Introduce transactions**



Manager's Burden

- **Pay for the initial development efforts (add 1/3 of the overall development effort as estimate)**
- **Never change the object model afterwards since customers will not believe costs for simple changes!**
- **Handle the programmer with care, if s/he leaves ..**



Why not ...

- ***JDBC* ?**
Makes ODBC functionality available to Java but does not help to reduce the burden!
- ***JSQL/SQLJ?***
Extends Java by embedded SQL with static compilation but not much more!

Still two different Data Models
Java objects + relational tables



Alternative ...

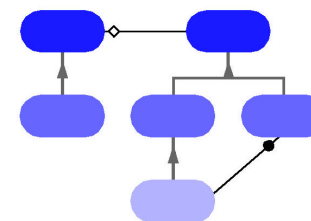
Java Blend



- A new product of Sun Microsystems,
- Developed in close co-operation between Baan Company, Sun Microsystems, and Tech@Spree



Tech@Spree

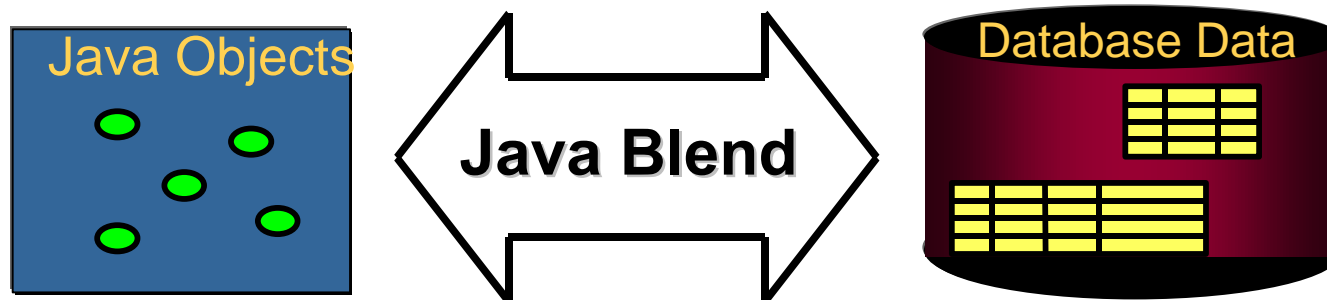




Java Blend is

A Persistence Framework

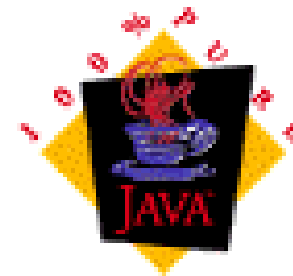
- which allows to develop enterprise applications entirely in Java
- which transparently maps Java objects to database records, and vice versa (reverse engineering)
- which first overcomes the impedance mismatch





Functional Aspects

- Database adapter for Java, based on RDBMS
- Abstract access to RDBMS via *JDBC*
- Full *transaction* capabilities
- Full *object-oriented query* management
- *100% Pure Java* environment (development and runtime)
- ODMG 2.0 compliance





ODMG 2.0 Compliance (1)



Object Data Management Group defines:

- An abstract *object model* for persistent objects,
- An *Object Query Language* (OQL) and
- For commonly used object-oriented languages (Java, C++, Smalltalk) a specific *language binding*



ODMG 2.0 Compliance (2)

Reasons to follow ODMG

- **Definitions are quite mature**
- **Supported by all OODBMS vendors**
- **Supported by OMG**

(and helps to hinder engineering fights!)

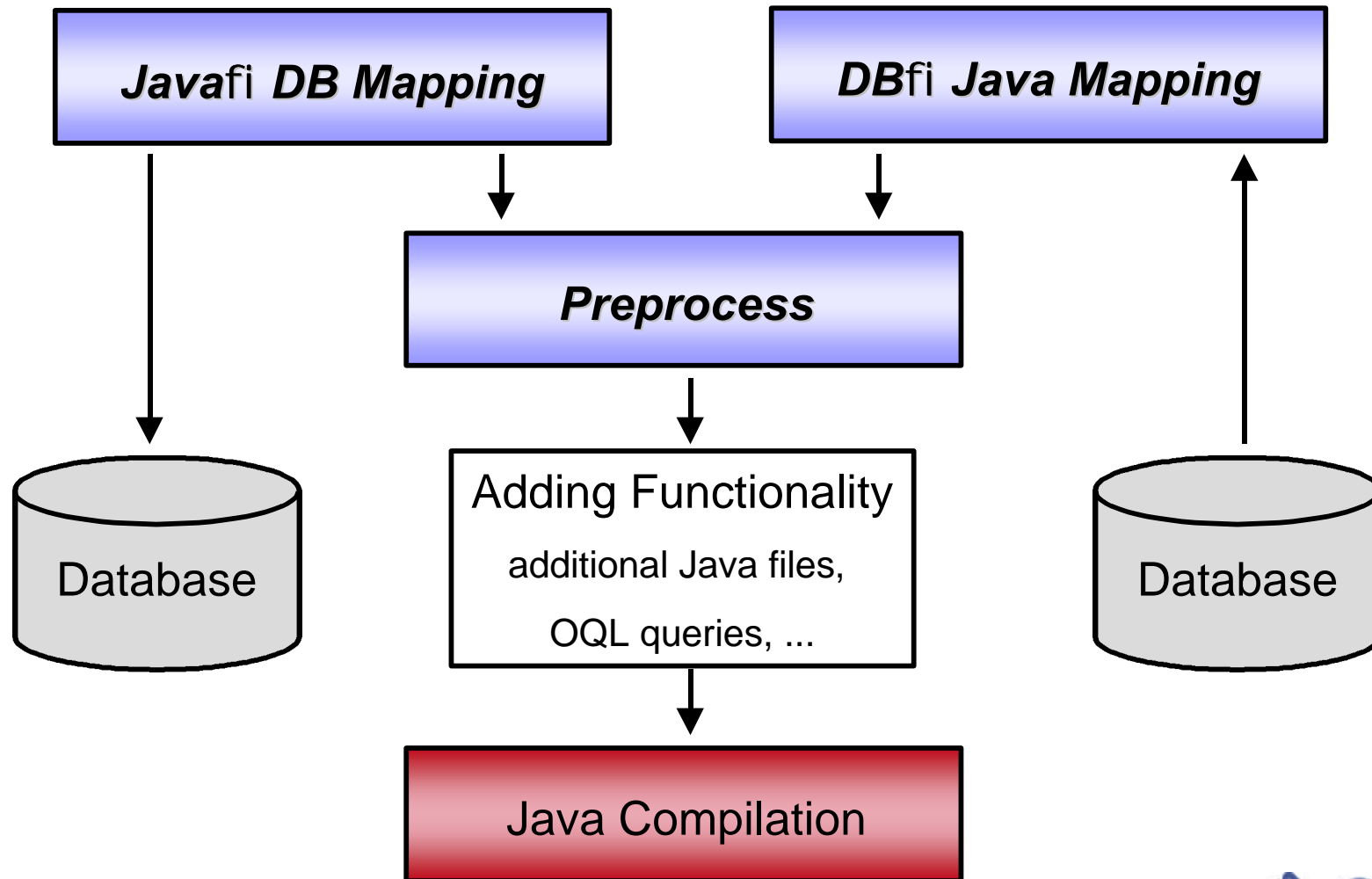


Application Model

- Java language binding by *pre-processing*
- Persistence by *persistence-capable* class and object
- All modifications (insert, update and delete) done by Java operations
- Java object access by
 - Navigation and
 - Querying (OQL)
- Persistent relationships between classes
 - uni-, bidirectional
 - 1:1-, 1:n-, n:m-cardinality



Two Approaches





Java→DB Mapping

- **Default mapping provided**
- **All persistence-capable classes of a Java package are mapped to tables, primary keys, foreign keys, ...**
- **Strict separation of object identification and object data: additional OID columns generated**
- **Mapping can be changed by programmer**



Database Info

The screenshot shows a 'Database Info' window with a tree view on the left and a 'Table Info' panel on the right. The tree view shows a database structure with tables: TableCourse, TableLectureRoom, TablePerson, TableProfessor, and TableStudent. The 'TablePerson' table is selected, and its 'Column' tab is active in the 'Table Info' panel.

Table Info - Column Tab

Column Name	SQL Type	Nullable
firstname	VARCHAR(100)	true
birthday	DATE	true
lastname	VARCHAR(100)	true

Below the table, there are input fields for 'Name', 'Type', 'Length', 'Precision', 'Scale', and an 'Update' button.

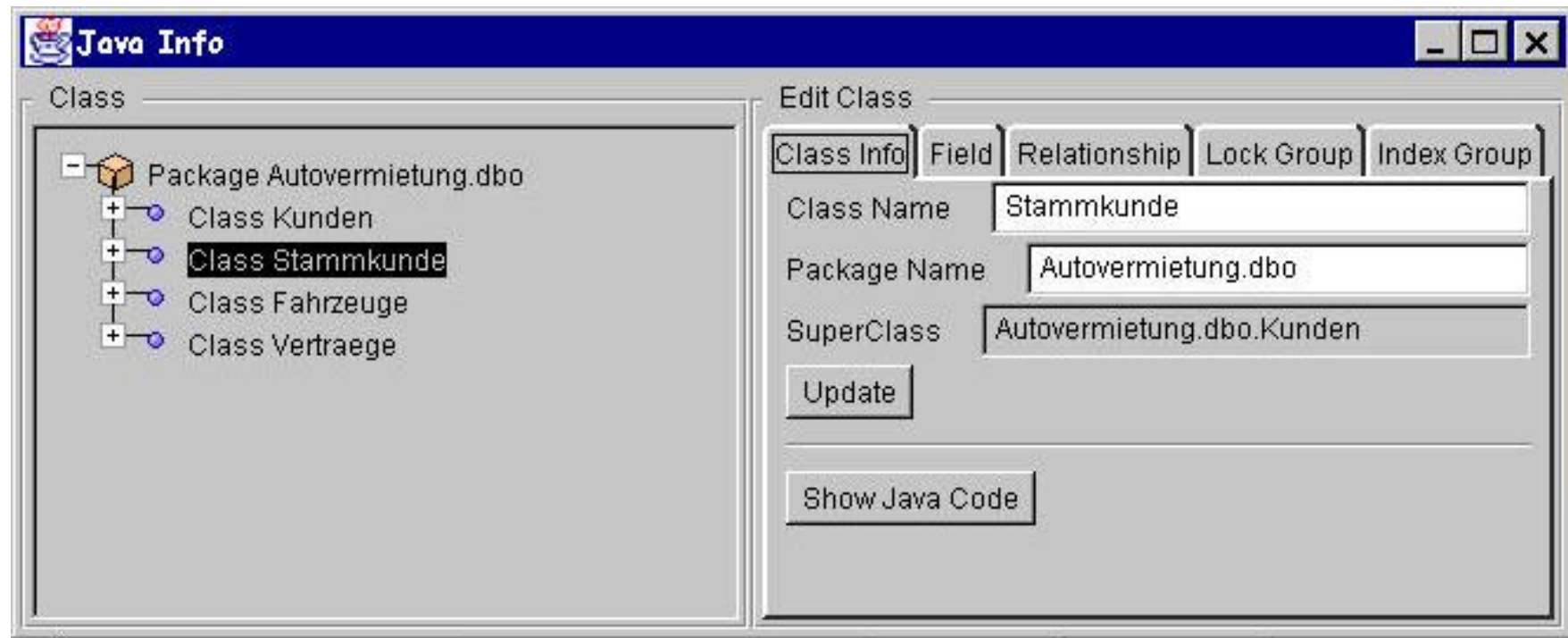


DB→Java Mapping

- Mapping process uses schema information to determine a default mapping
- Primary keys are used for object identification (each user-defined PK can act as **OID**)
- Foreign keys are used for relationships
- Mapping can be enhanced by the user



Java Info





Transactions (1)

- All modifications to persistent objects occur within the scope of a Java Blend transaction
- Same transaction model regardless of underlying database
- Transparently stores objects back to database on commit transaction
- Transaction abort rolls back persistent objects to their value at the beginning of the transaction



Transactions (2)

- **Optimistic and pessimistic locking**
- **Support of lock groups: parallel optimistic transactions can modify the same object**
- **Multiple threads with multiple transactions**
 - multiple threads can join a running transaction
 - multiple threads can run parallel transactions in one Java-VM



Querying Persistent Objects

- **Queries are written in terms of Java classes, not in terms of underlying tables**
- **Queries can include**
 - **Complex object-oriented expressions**
 - **Relationship navigation**
 - **Field access and method calls**
- **Query evaluation takes care of updates**
- **Query optimization**
 - **translates OQL queries to SQL queries or**
 - **makes use of pre-generated SQL statements for index groups**



Code Example (1)

```
import com.sun.javablend.*;

public class Person implements PersistenceCapable {
    protected String firstname;
    protected String lastname;
    protected Address homeAddress; // 1:1 persistent relationship

    public Person (Database db, String firstname, String lastname) {
        db.makePersistent(this);
        this.firstname = firstname;
        this.lastname = lastname;
        this.homeAddress = new Address (db, 10783, "Berlin", "Buelowstr.", 66);
    }

    public updateTownInAddress (int newZipCode, String newTown) {
        Transaction t = new Transaction();
        t.begin();
        homeAddress.setZipCode(newZipCode);
        homeAddress.setTown(newTown)
        t.commit();
    }
}
```



Code Example (2)

```
import com.sun.javablend.*;

public class PersonManagement {

public Bag selectPersonsByLastName (String lastname) {
    OQLQuery q = new OQLQuery();
    q.create(Person.class.getResourceAsStream("SelectPerson.ser"));
        // SELECT p
        // FROM p IN de.spree.Person
        // WHERE p.lastname = %(string)lastname
    DCollection result;
    q.bind("lastname", lastname);
    result = (DCollection)q.execute();
    for (Iterator e = result.iterator(); e.hasNext(); )
    {
        ((Person)e.next()).doSomething();
    }
    return result;
}
}
```




Scalability and Performance

- **Optimistic locking schema increases concurrency**
- **Lock groups increase concurrency**
- **Index groups increase query performance**
- **Query optimization for pessimistic locking option (OQL→SQL translation)**
- **Stored procedures for efficient transfer of data**
- **Multiple threads**
- **Different mapping strategies**



Benefits

- **Ease of use**
 - Java developers become database programmers
 - Single data model
 - Fully integrated query language
- **Sophisticated, customizable mapping**
- **Database independence**
- **Legacy compatibility**
- **Supports multiple architectures**
 - as client in 2-tier architecture
 - as middle tier in 3-tier architecture



References

■ Base Literature

- Cattell, R.G.G., et.al.:
The Object Database Standard. ODMG 2.0.
Morgan Kaufmann, 1997, ISBN 1-55860-463-4
- Hamilton, G.; Cattell, R.; Fisher, M.:
JDBC Database Access with Java . A Tutorial and Annotated Reference.
Addison Wesley, 1997, ISBN 0-201-30995-5

■ Product Homepage

- <http://java.sun.com/products/java-blend>

■ Tech@Spree Contact

- <http://www.spree.de>
- <mailto:tech@spree.de>
- Tel.: +49 (0)30 235 520 0

