

Toplink

- **Bob Brown**, Transentia Pty. Ltd.
- **Paul King**, ASERT Pty. Ltd.

ASERT

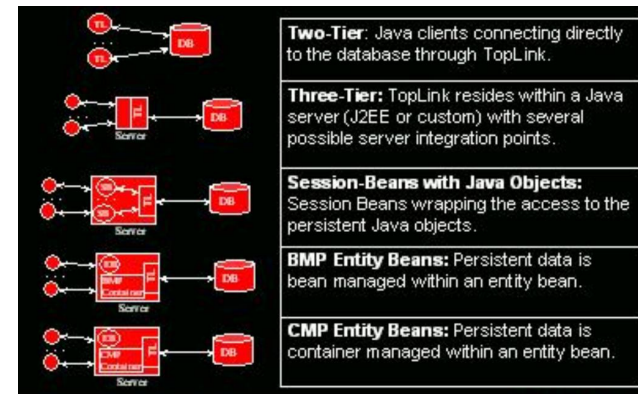
transentia

Positioning

- Persistence Manager & Toolkit
 - Sits on top of JDBC
 - Permits flexible Object/Relational mapping
 - Integrated application server support
 - Oracle9iAS, Weblogic Server, Websphere...
 - To be part of standard installation in 9.0.4
- (history...)
 - Smalltalk & The Object People, Webgain
 - Now Oracle...
- Extensive feature set
 - (as we'll see...)

Problems

- Two/three/n-tier architecture
- Dealing with
 - Complex relationships
 - Disparate data sources
 - SQL/not
 - Legacy code/data
 - Various technologies
 - Change
- Need to partition systems
 - Disparate skill sets; custodians of data/systems; etc.



'The object world and the relational world do not match up. One world consists of tables, rows, columns, and foreign keys; the other world contains object references, business rules, complex relationships, and inheritance. This is often referred to as the object/relational "impedance mismatch".'

Key Indicators

- If your database/data model changes frequently
- If you are writing an off-the-shelf application which needs to map with customers' database/data models which aren't known until deployment time
- If you need loose coupling between developers (who can create nice OO models) and DBAs (who can optimise store behaviour)

Benefits

- Little impact on ideal application design
 - Design to business needs rather than be forced by a particular approach to satisfy persistence needs
- Code-Free Integration
 - XML-metadata driven approach
 - Supports bean-to-bean and bean-to-object and object-to-object relationships directly
 - Provides for data-driven transforms
- Cut Development Time
 - A large proportion of a project's development time & cost is associated with persistence mechanisms
 - Graphical Mapping Workbench tool
 - Can auto-generate mapping code
- Permits tuning and optimisation of persistence mechanism
 - Claim is "up to 500%"...

Benefits...

- Leverage Current Developer Skills
- Provides several querying options so can use whatever is 'natural': SQL and stored procedures; Toplink's TOP Expression framework; EJB-QL, etc.
- Avoid Writing Infrastructure Code
- Provides a variety of locking strategies
- Sequence number management
- Logging facilities
- Distribution
- Application Server Integration
- Support for non-CMP/EJB architectures

Advanced Mapping Support

- Object relationships differ from database relationships
 - difficult to map by hand
- Toplink supports 15 mapping types, including:
 - Mappings to handle object/relational types as defined by JDBC
 - Arrays, etc.
 - Direct to Field, One to One, One to Many, Many to Many
 - Any kind of key relationships in database supported including intermediate tables
 - Object Type, Transformations
 - Enumeration ('Male' maps to 'M') or conversions (e.g. String to Number or Boolean)
 - Others
 - Serialized mappings, direct collections, stored procedures

Mappings...

- Inheritance
 - Not supported in 'straight' J2EE environments
 - Allows a subclass to reside in the same table as its superclass or be in a different table
 - Queries and relationships based on abstract classes
 - Class/subclass information maintained when reading/writing
- Interface
 - Queries and mappings can be defined on Interfaces rather than concrete classes
 - Allows heterogeneous types to be mapped via interfaces
- Aggregations
 - Multiple objects/beans per row
 - Map an object/bean to multiple tables
 - Mapping across databases

'Indirect' Mappings

- Allows “on-demand” retrieval
 - Improves performance
 - Not invisible
 - Although catered for by the mapping workbench
 - Direct/method access to ValueHolders

```
protected Address address;
public Employee() {
    address = null;
}
public Address getAddress() {
    return address;
}
public void setAddress(Address address) {
    this.address = address;
}
```

```
protected ValueHolderInterface address;
public Employee() {
    address = new ValueHolder();
}
public Address getAddress() {
    return (Address)address.getValue();
}
public void setAddress(Address address) {
    this.address.setValue(address);
}
```

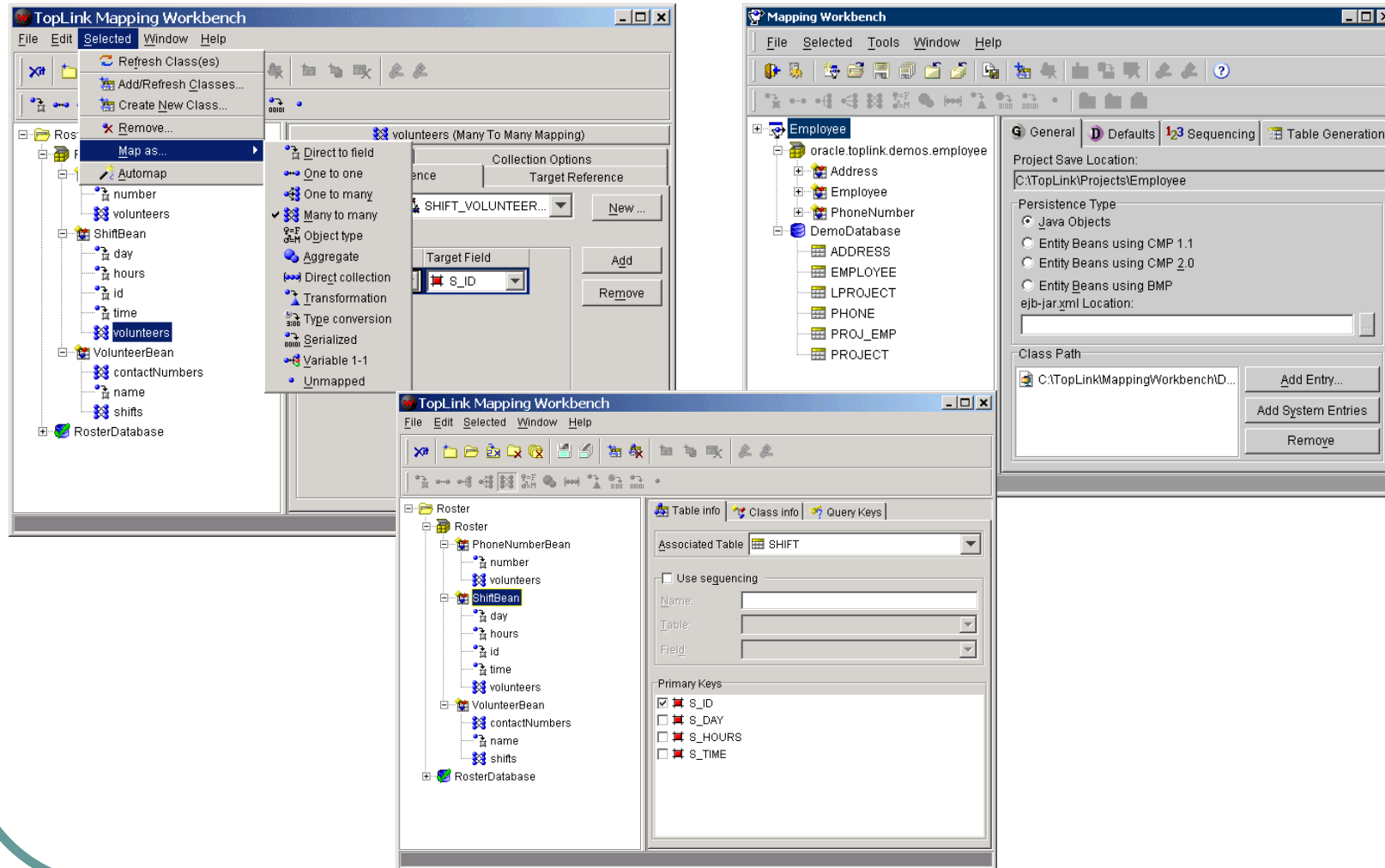
Mapping Workbench

- **Toplink IDE**
 - Assists developers in defining how objects map to the database
 - Currently separate from JDeveloper; will be incorporated in later releases

Workbench...

- Allows for rapid development and prototyping
 - Classes can be imported into the Mapping Workbench
 - Can generate a schema from the object model
 - Database meta-data can be read in through a JDBC connection
 - Can generate a plain Java object model or one based on Enterprise Java Beans
- Can identify inconsistent mappings or missing information

Workbench...



Object-Level Querying

- SQL doesn't cut the mustard for querying across collections of Java objects
- ExpressionManager
 - Can build sophisticated queries at the object-level, rather than using SQL or JDBC

```
ExpressionBuilder emp = new ExpressionBuilder();

Expression exp = emp.get("address").get("street").equal("Meadowlands Drive");
Vector employees = session.readAllObjects
    (Employee.class, exp.and(emp.get("salary").greaterThan(10000)));

SELECT t0.VERSION, t0.ADDR_ID, t0.F_NAME, t0.EMP_ID, t0.L_NAME, t0.MANAGER_
ID, t0.END_DATE, t0.START_DATE, t0.GENDER, t0.START_TIME, t0.END_TIME,
t0.SALARY FROM EMPLOYEE t0, ADDRESS t1 WHERE ((t1.STREET = 'Meadowlands')
AND (t0.SALARY > 10000)) AND (t1.ADDRESS_ID = t0.ADDR_ID)
```

Querying...

- “Query By Example”
 - Allows for queries to be specified by providing sample instances of the persistent objects to be queried
 - Can constrain/modify the operation via additional policies

```
ReadAllQuery query = new ReadAllQuery();
Employee employee = new Employee();
employee.setFirstName("B%");
employee.setLastName("S%");
employee.setSalary(0);
query.setExampleObject(employee);
QueryByExamplePolicy policy = new QueryByExamplePolicy();
policy.addSpecialOperation(String.class, "like");
policy.alwaysIncludeAttribute(Employee.class, "salary");
query.setQueryByExamplePolicy(policy);
Vector results = (Vector) session.executeQuery(query);
```

Uses LIKE operator for String comparisons and includes the salary even if it is zero

Querying...

- Can interact with stored procedures
 - Very important for opening up legacy stores/applications

```
StoredProcedureCall call = new StoredProcedureCall();
call.setProcedureName("CHECK_VALID_POSTAL_CODE");
call.addNamedArgument("POSTAL_CODE");
call.addNamedOutputArgument("IS_VALID");
ValueReadQuery query = new ValueReadQuery();
query.setCall(call);
query.addArgument("POSTAL_CODE");
Vector parameters = new Vector();
parameters.addElement("L5J1H5");
Number isValid = (Number) session.executeQuery(query,parameters);
```

Transactions

- Provides a transaction framework
 - Associates a “Unit of Work” with a database session
- Enhances database commit performance by tracking and updating only the changed portions of an object

Performance/Tuning

- “Just-in-Time” reading
 - Delay reading the related objects until used
 - If an object is never referenced, the resources required to read in the object are never used
- Object Caching
 - Eliminates the need for a database call if the object is requested within a timeout period
 - Several different types of caches and they can be set on a class-by-class basis
 - (A)Synchronous mechanisms ensure cache consistency in a clustered environment
- Batch operations
- Efficient ‘blocked’ cursor operations

Performance...

- Features are available to set options such as:
 - Ordering
 - Query optimization (batch reading, joining)
 - Cursors
 - Refreshing, caching options
 - Pessimistic locking
 - Outer joins
 - Nested sub-selects
 - Parameter binding, statement caching
 - Can specify custom SQL
- Query can be run “in memory” against cache

Toplink vs. EJB 2.x CMP

- EJB 2.0 does not provide a persistence solution for common Java architectures:
 - Regular Java objects using Servlets and JSPs
 - Session Beans accessing "regular" Java business objects
 - Entity Beans using Bean Managed Persistence
 - Mapping data to non-database stores
- Facilitates and manages persistence in both EJB and non-EJB architectures
- Considerable advantages over the basic EJB 2.0 specification:
 - Complete freedom to mix entity beans with lightweight Java objects
 - Better performance
 - Enhanced developer productivity with visual Mapping Workbench
 - Support for database features such as stored procedures and outer-joins
 - Optimized transactions and database interaction
 - In-memory querying support
 - Ability to use same concepts on non-CMP architectures
 - Advanced, sophisticated mapping support
 - Flexible optimistic and pessimistic locking strategies
 - Reporting framework

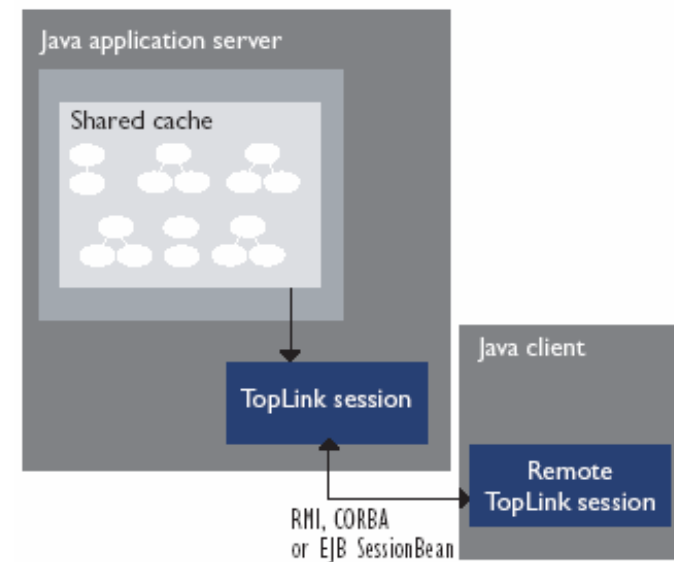
Toplink and CMP

- Think of Toplink as a very configurable persistence manager for the container

'TopLink Container-Managed Persistence provides a number of entry points for advanced customization of mappings, logins, and other aspects of persistence. These can be used to take advantage of advanced TopLink features, JDBC driver features, or to gain "low-level" access to TopLink for Java APIs that are normally masked in the container-managed persistence layer.'

Remote Toplink Sessions

- Brings 'server-side' functionality to a client application
 - Client-side caching and object identity
 - Client-side units of work with nested and parallel support
 - Remote proxies for on-demand loading of object relationships from the server
 - Complex querying support on the client



Future

- Continue to support JDBC-compliant databases
 - Oracle, IBM DB2, MS SQL Server, Sybase, Informix and Microsoft Access, etc.
- Continue to support standalone Java applications
- Continue to support J2EE-compliant application servers including Oracle9iAS, IBM WebSphere, and BEA WebLogic
 - CMP integration with 9iAS, WebSphere & WebLogic
- Expand support of hardware platforms
 - Solaris, NT/Windows 2000, HP-UX, Tru64, AIX, Linux
- Better JTA integration
- Support for Oracle9iAS connection pooling
- Oracle9iAS/OC4J CMP & BMP entity bean support
- Enhanced support for the Oracle 9i database and JDBC drivers
- Mapping Workbench to be integrated with JDeveloper