Database Applications For Enterprise Use

Arkady Ganov

IGK Credit Management

Riga 2013

Basic Rules For Development

1. *Designed for Internet:* central database, web-located client applications, OLTP and batch, secure communications

2. *Rich client UI:* multi-window, multi-threaded, no potential restrictions

3. *3-tier business logic:* DB stored procedures, MT enterprise operations, CT processing (local I/O, XSLT)

4. *Data management framework:* IDE with automation for DBRE and proposed appl. architecture, models for UI controls, CT txn state detection, customizable datasource operations

5. *Role-based data access:* controlled by database for every user individually, includes RLS and CLS

6. *Code generators* by module types: common template is customizable by tagged *<inserts>*, DBRE for column iterators

Working Environment

- *Oracle Database SE 10.2:* central database, views and stored procedures, incl. application security with RLS and CLS

- *Oracle GlassFish OSE 3.1:* application server for running Web Services and EJB-s, central web-enabled repository for applications deployment

- *NetBeans 7.3:* IDE used to build MT components running on Oracle GlassFish OSE server

- *Oracle JDeveloper 10g (9.0.5.2):* IDE for building data-centric applications, includes Oracle Application Development Framework (ADF)

- *Oracle Internet Application Server SE 10g:* J2EE containers (OC4J), acting as a copy of MT services for remote clients - *in process of disabling as of November, 2013*

1. Designed For Internet

Used standard technologies:

- *SOAP Web Services:*

    centralized enterprise operations (incl. *Web Data Service*) are called by remote clients and results are returned in program communications through Internet, where messages are sent via HTTP in the form of textual XML documents

- *Java Web Start deployment:*

    single web-enabled repository keeps current version of every application for automatic deployment to client devices

2. Rich Client UI

- *Java SE* provides comprehensive environment for building multi-window, multi-threaded, fast UI, with no potential restrictions as having wide variety of available free tools

- *Java Swing components* have all the necessary requirements along with *model support* from Oracle Application Development Framework (ADF, see *Data Management Framework* below)

- Java client applications are considered *Rich Enterprise Applications* (REAs) since they are:
  - deployed by JRE/JWS from single web-enabled repository
  - work with remote database making calls to Web Data Service
  - access other enterprise operations in remote web services
  - have advanced UI running in standard platform independent engine

3. Three-Tier Business Logic

- *Database stored procedures:*
  - data consistency and integrity control, doing business logic, providing application security

- *Middle-tier enterprise operations:*
  - remote clients get Internet access to central database through single uniform Web Data Service, which listens for client requests and makes JDBC calls to database for processing
  - generating documents, sending different types of e-mail messages by requests from remote clients

- *Client-side processing*:
  - validation of input data where database is not required
  - I/O with local file system: read/write files in different formats
  - improve reaction by decreasing network traffic and making finishing processing locally (get XML, transform into PDF)

4. Data Management Framework

Oracle Application Development Framework:

- ADF is recommended by Oracle for data-centric applications
- JDeveloper IDE builds application code for all 3 tiers, including database, application server and client; ADF related components are generated from database objects (DBRE)
- ADF supports different types of UI components, including Swing
- ADF Swing proposes application architecture and wizard
- ADF provides data models for Swing components, which are embedded into common application binding architecture
- ADF detects model changes and manages client-side txn state
- ADF has API to its datasource operations, which could be completely re-defined in order to work with arbitrary source
- Created *Web Data Service* solution allows for accessing remote database in the low model level, keeping UI without changes

5. Role-based Data Access

- Separate application manages data access declarations stored in database – in addition to Oracle security features

- Stored declarations define data access for business groups, application roles and table privileges – down to the read/write access to certain column values (CLS)

- Users get their access rights in a business group by assigning a certain application role

- Row Level Security and write column access are based on Oracle security means and application access declarations

- Read CLS is based on application access declarations and provided by stored procedures with *pipelined* functions

- In case of Read CLS, client doesn't send clear SQL, but rather a *QueryID* and *predicates*, while database returns selected rows without column values that should be hidden from the User

6. Code Generators

- Code generators exist for the main module types and contain common business logic and behavior – for unifying across all the application structure for easy management

- Same approach: type-specific template is customized by *parameters*, tagged *<inserts>* and *column iterators* (generate from database object definitions, DBRE)

- Simple technique of replacing *parameters* and *inserts* allows for flexible and easy management, because replacing unit can contain any piece of code and is not restricted by syntax rules

Notes On Implementation

- *JDeveloper 9.0.5.2* was the last version where I could disable default JDBC calls from BC4J (and call Web Data Service instead) – as it was described in *ADF Developer's Guide*

- Originally created for Java 1.4, used ADF version (Jdev 9052) should be slightly modified for running under Java 6+, *see* *JUTableBinding.txt*

- *Logging* is provided by the MT web services, since all user actions go through them; e.g. database actions are logged by the Web Data Service