Application Servers
G22.3033-011

Session 6 - Sub-Topic 1
Using JavaBeans

Dr. Jean-Claude Franchitti

New York University
Computer Science Department
Courant Institute of Mathematical Sciences

Session Outline

• Enterprise JavaBeans tips
Enterprise JavaBeans Tips

• Three Enterprise JavaBeans revisions
  – Moscone
    • Released second quarter 1999
    • Included fixes to clarify the 1.0 specification
    • Described the standard for specifying deployment descriptors using XML
  – Javits
    • Released 6 months after Moscone
    • Emphasized connectivity with legacy systems

• Enterprise JavaBeans revisions (continued)
  – Milano
    • Released 9 months after Javits
    • Further specified Entity beans
  – Recommended Java servlets (or another server-side mechanism) to communicate with EJBs and create fewer compatibility issues
  – Related press releases & information
    • http://java.sun.com/pr/1998/12/pr981208-05.html
    • http://java.sun.com/products/ejb/faq.html
Enterprise JavaBeans Tips

• Need to thoroughly test code that must run in multiple EJB environments
  – Reference implementation & conformance tests for cross-container portability not yet available
• Prototyping is recommended
  – Design validation & implementation test
• Abide to specification
  – e.g., EJB container does not allow starting or manipulating new threads

Enterprise JavaBeans Tips

• Internet applications
  – Use Java servlets or some other server-side mechanism to communicate with your EJBs
• Use both EJB and CORBA
  – Many vendors use the Internet InterOrb Protocol
  – RMI-over-IIOP has been released
  – EJB acts has a toolkit that sits in the middle tier and provide a variety of services
Enterprise JavaBeans Tips

- Integrate EJB and legacy systems now
  - EJB can make a call to the legacy system
  - Legacy system can make a call to an EJB
- Pass Java Messaging Service (JMS) messages to an EJB
  - EJBs can send JMS messages using the JMS client API
  - There is no standard way in containers to process incoming JMS messages (need to create a non-EJB proxy to call a bean remote I/F)

Enterprise JavaBeans Tips

- Creating arrays of structures out of arbitrary-length query result sets?
  - JDBC drivers don’t let you find out the number of rows returned by a query
  - Until JDBC 2.0, drivers supported forward-only cursors
  - To insert new data, you must know the array size in order to allocate it
  - Solution
    - Build a wrapper class to a hashtable
Enterprise JavaBeans Tips

– Solution (continued)

• Create a new add() method that takes the necessary arguments, constructs a new object, and adds it to the hashtable
  public void add(int employeeID, Data date, double hours, int contract)
  {
      DataObject tsl = new DataObject(employeeID, date, hours, contract);
      put (tsl, tsl);
  }

• Create a new method that finds the number of objects in the hashtable, create an array of that size for each object in the table, adds it to the array, and return the array

```java
import java.sql.Date;
import java.util.Enumeration;

public class MyHash extends java.util.Hashtable
{
    DataObject[ ] getDataObjects( )
    {
        int size = size();
        DataObject[ ] contents = new DataObject[size];
        // get list of keys in hashtable
        Enumeration mykeys = keys( );
        // for each key, retrieve the contents
        for (in I = 0; I < size ; I++)
        {
            contents[I] = (DataObject) get(mykeys.nextElement( ));
        }
        return contents;
    }
}
```
Enterprise JavaBeans Tips

• Know when to use Session and Entity beans
  – Entity beans should represent a particular domain entity (e.g., row in a database table)
  – Entity beans can be shared among multiple clients, and can last beyond the lifespan of a single client
  – Stateful session beans are associated with a particular client
  – Session beans are created/removed by the client
  – Legacy system can make a call to an EJB

(continued)

Enterprise JavaBeans Tips

• Know when to use Session and Entity beans
  – Use an Entity bean to represent a sharable entity that persists across client connections
  – Use a stateful Session bean to cache state information
  – Use a stateless Session bean when you don’t need to keep state information in the middle tier
Enterprise JavaBeans Tips

• Don’t use Entity EJB objects to represent fine-grained data
  – Use higher-level object and aggregates instead
  – e.g., time tracking system
    • likely to have Employee Entity bean but not a TimeSheetLineEntry Entity bean
  – Methods should be relatively large as well
• Isolate vendor dependencies