An Internet Accounting System

Kenji Ohmori
Faculty of Computer and Information Sciences, Hosei University
ohmori@k.hosei.ac.jp

Abstract

An Internet accounting system, where transactions are entered on the spot and business data created by transactions are distributed to appropriate staff of the company, becomes more important for companies operating worldwide. The Internet accounting system described in this paper consists of a three-tier structure: Web, application and database servers. The accounting system provides complete accounting functions including general ledger, accounts payable, accounts receivable, purchase order, sales order, inventory management, fixed assets, temporary payment, multi-users, multi-currencies, multi-companies and multi-languages allowing customer relation management, partner relation management, supply chain management and performance analysis. A prototype system has been completed and a full-scale system is now under development.

1. Introduction

Globalization forces every level of companies to operate worldwide. The traditional centralized accounting system[1], where officers of the accounting department mainly operate the system at the head office, does not appropriately function in the globalized environment. An Internet accounting system is strongly required in worldwide operating companies, while providing high-quality enterprise resource planning (ERP) and customer relation management (CRM). Transactions occurring at each branch office tightly related to customers are entered on-site into the system through a Web site. Transactions sent to the main office are used to provide ERP and CRM information for top executives and supervisors.

There are many accounting systems from the commercial side as well as open source software including SQL-Ledger[2], Linux-Kontor[3] and Compiere[4]. SQL-Ledger uses PostgreSQL implemented atop a Web server such as Apache. It provides accounts payable, accounts receivable, general ledger, billing/invoicing, purchase and sales orders, customers, vendors, inventory, assemblies, chart of accounts and administration modules, featuring customizable taxes, multi-users, multi-companies, foreign currency support, internationalization, access control, SQL server backend, financial statements, backup to file/email, import data and application interface. Linux-Kontor uses Adabas-D as the main underlying database system, Java and JDBC to build a platform-independent client and server software system. Compiere can optimize inventory, enter sales orders, receive orders from the Web, create invoices and record shipments, collect receipts and match with bank statements, generate or enter purchase orders, record supplier receipts and invoices, pay suppliers, enter manual journals and print report and statements. Compiere uses Oracle and Java.

Some accounting systems from open source software support a client-server-base environment using a Web server. However, none of these systems are fully Internet enterprise systems, where systems are built on Web, application and database servers.

2. System Requirements

Proceedings of the Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'03)
0-7695-1957-1/03 $17.00 © 2003 **IEEE**

The Internet accounting system is required to give Web, application and database services with a persistent and robust environment. The Internet accounting system is also required to provide complete accounting functions including general ledger, accounts payable, accounts receivable, purchase order, sales order, inventory management, fixed assets, temporary payment, multi-users, multi-currencies, multi-companies and multi-languages allowing customer relation management, partner relation management, supply chain management and performance analysis.

The Web service should provide interactive communication between end-users and the Internet accounting system. The Web service has to be equipped at each branch office, factory and subsidiary. A local database service should be equipped with the Web service to keep locally required sales, purchase and employer data and local accounting information. End users receiving the Web service are expected to be all the levels of employees of the company, who may not have accounting knowledge. Furthermore, it has to provide various levels of securities including user authentication, recognition and validation.

The application service has to be persistent so that transactions are secured between users and the main database system. Any failures during transactions caused by illegal operations, process interruptions, power failures and illegal data formats, have to be completely recovered. The database service is required to handle a large amount of data without causing troubles for end users.

3. System Design

The interactive accounting system depicted in Figure 1 has been designed using a three tier-structure, consisting of Web, application and database servers to fulfill the above requirements completely. This three-tier system allocates model, view and controller components to the Web and application servers. This division is a crucial issue from the viewpoint of designing a complicated interactive system. In this system, Tomcat[5] serves as the Web servers, so does JBOSS[6] as the application server and PostgreSQL[7] as the database server. Web application programs have been designed with Struts providing flexible design and implementation of complicated state transition of home pages.

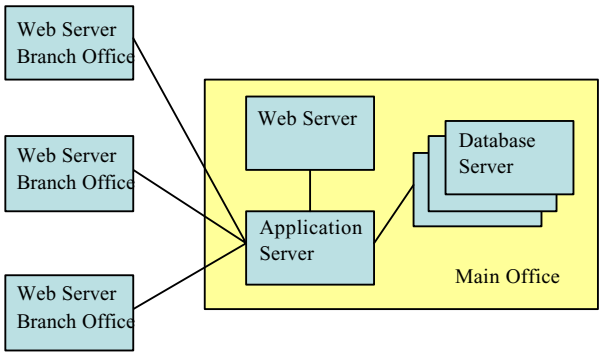

Figure 1. An Internet Accounting System

Enterprise Java beans (EJB[8]), serving as a model component and running on JBOSS, provide persistent functions for the system.

4. Implementation

4.1. User Interfaces

A Web server provides home pages serving as user interfaces between users and the accounting system. Home page programs are divided into several subgroups, each of which serves as an independent function of the accounting system. General ledger,

accounts receivable, accounts payable and temporary payments are examples. Information for accounts receivable is entered into the accounting system directly by staff of sales departments, so are accounts payable transactions handled by staff of purchase departments. Other transactions are also directly input not by staff of an accounting department but by members of on-site department.

As end users of this accounting system are non-experts, home pages provide user-friendly interfaces where the user can access to the accounting system not by technical accounting terms but by his/her ordinary words. This requirement leads to prepare many home pages. For example, the temporary accounting system where staff pays in advance and is reimbursed later needs 31 home pages. In such a system, the state transition of home pages is very complicated. After completing implementation, the order of home pages may be changed because of necessity to implement more user-friendly interfaces. Flexibility is the most important issue when designing home page programs. Typically, three functions consisting of view, controller and model have to be separated as described before. View and controller functions are provided by home page programs. The home page programs of this system are implemented by three different types of programs: Java Server Pages (JSP), action beans and action form beans. A JSP program provides a view function, so does an action bean a controller function. An action form bean is used to receive data from a previously executed JSP program and transfer modified data to a next executing JSP program.

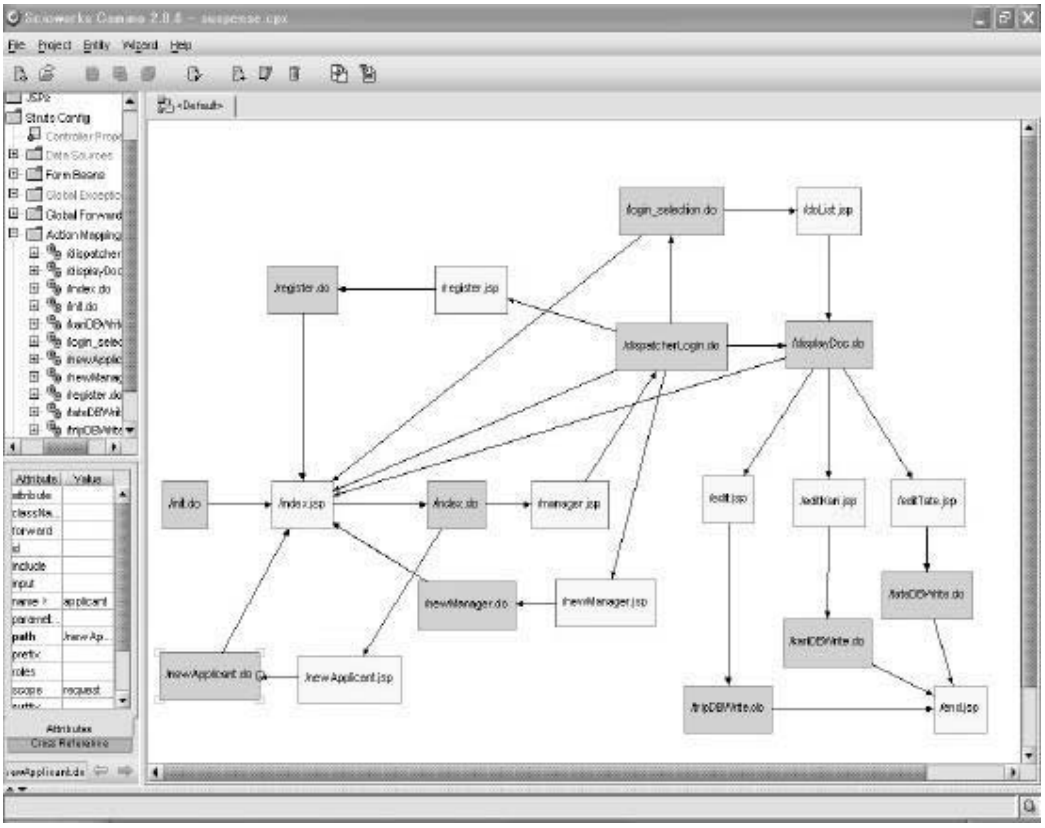

Figure 2. A State Transition Diagram of Temporary Payments.

The JSP programs have been implemented using Struts[10] giving a framework of flexible home page state transition. As some home page functions such as accounting code conversion and money presentation are very complicated when generating user-friendly interfaces, they cannot be implemented using JSP programs. These parts have been implemented by Java Scripts. Internationalization and localization are also important functions. Every JSP program is equipped in order that it supports multi-languages. A JSP program consists of class and tag library definition part, Java Script part and HTML body part.

The state transition for temporary payments is depicted in Figure 2. This state transition is for the prototype system. It consists of 10 JSP programs and 11 action beans. On the other hand, the full-scale accounting system consists of 31 JSP programs. The action form beans are not depicted in this figure. The prototype system has five action form beans. Some of JSP programs share the same action form beans. When the temporary payments subsystem is called, the initialization program is executed at first. The program initializes the subsystem and reads property files to prepare for internationalization and localization. Then, the first home page is displayed on the user screen. Figure 3 shows a home page of a data input stage in case of Japanese locale. The upper part is used to input user information such as applicant name and affiliation, accounting date and summary of this temporary payment. The lower part is information about an expected expense of each item which the applicant will buy.

Figure 3. A User Interface.

4.2. Session and Entity Beans

The main part of the accounting system is provided by enterprise Java beans (EJB) carrying out business logic. EJB are divided into two parts: session beans and entity beans. The session beans receive information from a Web server through an action bean. The traffic between Web and application servers depends on an accounting subsystem. Some subsystems have heavy traffic and others are not. For heavy traffic subsystems, session beans are tightly connected with the corresponding action beans. In this case, data between them are transported through data access objects defined by their entity beans.

On the other hand, light traffic subsystems transport data through XML. The current accounting system is expected to be often upgraded in order to cope with changes of accounting rules or technology advances. The current system has to be flexible to cope with these changes so that XML is used.

Receiving requests from an action bean, the corresponding session bean generates database data acceptable for its corresponding entity beans. An entity bean has access to

the database system using object-relational mapping. Relations between two tables of the database system are also covered by object-relational mapping. A one-to-one relation where each other's entity from two tables points each other is represented by an Object-to-Object relation, and so is a one-to-multiple relation where an entity of one table points one entity of the other table and the entity of the other table does several entities represented by an Object-to-Collection relation of entity beans.

There are two types of entity beans: BMP (Bean Management Persistent) and CMP (Container Management Persistent). In the accounting system, all the entity beans are implemented by CMP. Therefore, finder methods are used to retrieve the database system. The following program is an example of an entity bean described by Xdoclet[11]. Xdoclet can generate all the pieces of entity programs. The beginning of the program defines jndi-name used to call this program from other programs in a distributed environment. Then, the database table generated from this program is defined. The table name is Header. The class defined by this program is also depicted. This class inherits EntityBean Class. Header table is related to Journal table with a one-to-multiple relation defining getJournals and setJournals methods. Then, methods corresponding to the entries of this table follow. Lastly, EJB methods are defined.

```
/**
 * @author Kenji Ohmori (ohmori@computer.org)
 * @version $Revision: 1.0 $
 * @ejb:bean name="Header"
 *           display-name="Header Entity Bean" type="CMP" primkey-field="id" view-type="local"
 *           jndi-name="accounting/header" local-jndi-name="accounting/headerLocal"
 * @ejb:pk class="java.lang.Long" generate="false"
 * @ejb:transaction type="Required"
 * @ejb:data-object setdata="true" extends="org.gaia.transaction.interfaces.TransactionDTO"
 * @ejb:finder signature="java.util.Collection findAll()" query="SELECT OBJECT(o) from Header AS o"
 * @jboss:table-name table-name="header"
**/
public abstract class HeaderBean implements EntityBean {
  private EntityContext ctx;
   /**
    * @ejb:interface-method
    * @ejb:relation name="JournalToHeader"   role-name="Header-ReferredFrom-Journal"
   **/
  public abstract Collection getJournals();
  /** @ejb:interface-method **/
  public abstract void setJournals(Collection journals);
……..
   /**
    * @return Long
    * @ejb:persistent-field
    * @ejb:pk-field
    * @ejb:interface-method
    * @jboss:column-name name="id"
    */
  public abstract Long getId();
  /** @ejb:interface-method */
  public abstract void setId(Long id);
```

```
  /** @ejb:interface-method */
……..
  public Long ejbCreate(Long id) throws CreateException {
    if (id == null) throw new CreateException(" id is null"); setId(id);
      return null;
   }
  public void ejbPostCreate(Long id) {
   }
}
```

5. Conclusion

The new Internet accounting system of a three-tier structure consisting of Web, application and database servers is described. The Web servers with high-quality user-friendly interfaces serve as a presentation system to non-experts in the accounting system. The application server with robust persistent functions serves as a business logic system. The accounting system covers a complete accounting system. A prototype system has been completed and a full-scale system is now under implementation. It consists of eight subsystems. Each subsystem has 30 JSP programs for the Web service and 250 beans for the application service on average.

6. References

[1] Marshall B. Romney, Paul John Steinbart, "Accounting Information Systems", Prentice Hall; June 2002

[2] P.G. Sarang, Christopher Browne, Dietrich Ayala, Vivek Chopra "Professional Open Source Web Services" , Wrox Press Inc, July 2002

[3] http://www.linux-kontor

[4] http://www.compiere.org

[5] Chanoch Wiggers, et, "Professional Apache Tomcat", Wrox Press Inc, September 2002

[6] Meeraj Moidoo Kunnumpurath, "JBoss 3.0 Deployment and Administration Handbook", Wrox Press Inc, December 2002

[7] Korry Douglas, Susan Douglas, "PostgreSQL", New Riders Publishing, February 2003

[8] Floyd Marinescu, "EJB Design Patterns: Advanced Patterns, Processes, and Idioms", John Wiley & Sons, February 2002

[9] Simon Brown, et, "Professional JSP 2nd Edition", Wrox Press Inc, April 2001

[10] Ted Husted, Cedric Dumoulin, George Franciscus, David Winterfeldt, Craig R. McClanahan, "Struts in Action: Building Web Applications with the Leading Java Framework", Manning Publications Company, November 2002

[11] Damon Payne, Ed Lyons, "Professional Java Tools: Real world Ant, JUnit, CVS, Cactus, Buzilla, Maven, JMeter and XDoclet", Wrox Press Inc, April 2003