AN INTEGRAL CLIENT/SERVER DESIGN OF ACCOUNTING INFORMATION SYSTEMS

Jia-Lang Seng

諶家蘭*

摘要

分散式運算環境是未來企業資訊技術應用之主軸，跨平台之透通性，延展性，及可攜性，證明資訊應用系統必須走向開放及合作的整合型設計，在分散式環境中，資料庫扮演舉足輕重的角色，其中財務及會計資料是其核心，如何整合性設計一套會計資訊系統是企業分散式運算成功的基石，本文將提出改良後整合型設計之主從架構會計資訊系統。

Abstract

Distributed computing is the future trend where data and applications are stored and processed at multiple sites on networks. Data and applications must collaborate and communicate to accomplish cross-platform and mission-critical transactions.

Client/Server architecture is the current technology to enable the platform-independent collaboration and communication. Large-scale enterprises are righsizing corporate-wide applications on client/server platform. Distributed databases must support a variety of intra-organizational and inter-organizational information systems. The core of the databases is the financial and accounting data resources.Current Practice of accounting information system development is a partial design, or single-user design, or non-integral which cannot incorporate data resources to meet the requirements of distributed enterprise computing. In this paper, we present an integral design of the data resources and processes for the sales/purchases/inventory/personnel/payroll/general-ledger systems. The integration is based on the common data resources with their functions. The design is implemented in a three-term classroom project to show the performance and benefits of this approach.

Keywords: Client/Server Architecture, Middleware, Distributed Database Management Systems, Object-Oriented Technology, Competitive Advantage, Relational Database Management System, Accounting Information Systems

1 Introduction

A distributed database management system (DDBMS) consists of a collection of

* 作者爲本校資訊管理學系副教授

sites, connected together via some kind of communication network, in which (1) each site is a database system site in its own right, but (2) the sites have agreed to work together to that a user at any site can access data anywhere in the network exactly as if the data were all stored at the user's own site [5]. Distributed computing system contain software programs and data resources dispersed across independent computers connected through a communication network [2]. Client/Server (C/S) architecture is the latest trend in application of database technology and local area network (LAN) technology to give enterprises the price/performance competitive advantages. It allows you to optimize the use of hardware and software resources at both the front-end application and the back-end database server. C/S computing is regarded as a simple and special case of distributed computing system in which (1) some sites are client sites and others are server sites, (2) all data resides at the server sites, (3) all applications execute at the client sites, and the (4) the seams show, i.e. full location independence is not provided [5]. Front-end clients are the various applications that run on top of the DDBMSs and back-end servers are the DDBMSs to provide basic database functions such as data definition, data manipulation, data security, data integrity, data communication, concurrency control and recovery. The division of workloads improves the performance of access of shared data in networking environment and enables more users to access the same data using the existing software system [9]. Figure 1 shows an example of the C/S architecture based on the above description.

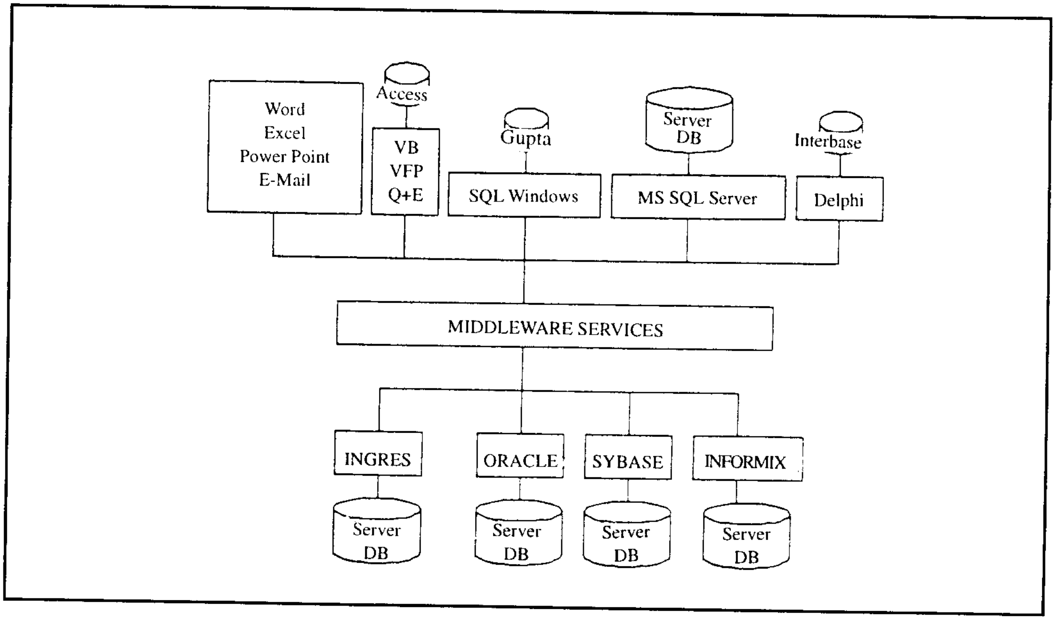

Figure 1: An Example of Client/Server Architecture

C/S system mirror the distributed structure of an enterprise. It gives the efficient accessibility, effective processing, and better availability. [4] points out that the computing facilities of enterprises are evolving into a utility which is a corporate-wide network of information services that allows knowledge workers to access application, database, and printing services as if they access a desktop appliance easily. The platform-independence architecture gives the transparency, connectivity, and interoperability in the resolution of the heterogeneity of distributed computing. It gives large-scale enterprises the competitive advantage with the coherent and uniform information services utility. Combined internet-and intranet-platform can emerge from the utility.

In Taiwan, [1] reports over 80% of the corporations have or are installing C/S architecture and more than 50% of the rest of the surveyed companies are thinking of setting up a C/S system. The key issues faced by most chief information officers (CIOs) interviewed are the incompatibility, poor design, lack of training, and lack of the ability to troubleshoot disconnections. Failures and redundancies prevail due to the non-integral and isolated designs. New systems cannot work with the existing application programs and databases. No prior experiences exist in the company to design effective C/S applications. It is hard and expensive to transfer an train everyone onto a new platform. New skills are needed but lack of to detect and solve connection problems, operation interrupts, and network performance.

[6] points out that the business area which creates the core of the distributed databases is the financial and accounting information systems (AIS). AIS captures, processes, and records transactions related to sales, purchases, inventory, and payroll which from the backbone of most businesses. Distributed computing would depend on AIS applications and data resources to support daily and ad-hoc management needs such as the middle-level management reports, control, and decision-making, and the executives' strategic planning and prediction, and the electronic data interchange (EDI). A well-designed AIS is important to a successful C/S implementation.

In this paper, we present an integral design of a client/server accounting information sustem. We intend to give an in-depth description of the data and process designs for an AIS project. We give an extensive overview of the C/S technology and choose one widely used C/S system, Microsoft Visual Basic 4.0, and Microsoft SQL Server 6.0, to implement the design. The prototype has been presented in a public exchibition. The paper is organized as follows. Section one is the introduction. Section two reviews the C/S technology. Section three gives the design of an integrated AIS with data and processes, and the implementation. Section four concludes the paper with a summary.

2. Client/Server Technology

2.1 Clients, Middleware, and Servers

Current three-tier C/S architecture divides the system tasks into (1) presentation management, (2) computation management, (3) information management, (4) control management, (5) communication management, and (6) data storage and retrieval [7]. Figure 2 shows that partition of the tasks. Clients, middleware, and servers manage these tasks in three layers. Clients and servers are defined as above. Middleware is a general-purpose gateway service that sits between the back-end server platforms and the front-end client applications. It is defined by the interfaces and protocols it supports. These standard interfaces and protocols allow the C/S systems to connect to different server platforms, to communicate different data requests and responses, and to inter-operate application programs and data resources between computing sites. It allows the computing environment to be transparent, portable, and scaleable across platforms.

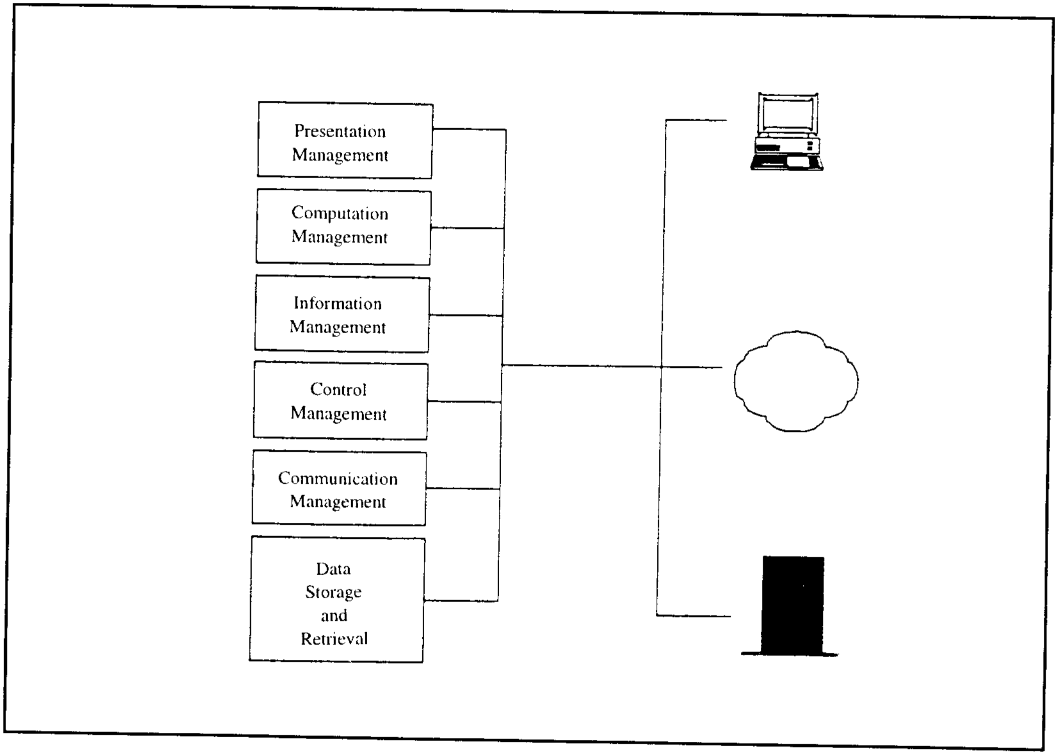

Figure 2: Tasks Partition in Three-Tier Client/Server Architecture

As quoted from [5], distributed C/S architecture implies that a single application should be able to operate transparently on data that is spread across a variety of different databases, running on a variety of different machines, supported by a variety of operating systems, and connected together by a variety of different communication networks so that from a logical viewpoint all data were managed by a single DBMS running on a single machine as a non-distributed system. The well-known Date's twelve rules consist of (1) local autonomy, (2) no reliance on a central site, (3) continuous operation, (4) location transparency, (5) fragmentation transparency, (6) replication transparency, (7) distributed query processing, (8) distributed transaction management, (9) hardware independence, (10) operating system independence, (11) network independence, (12) DBMS independence.

C/S technology depends on the middleware services to achieve the desired outcomes such as transparency, connectivity, and interoperability. Middleware mediates the client requests, transmits requests, converts requests, and secures results from different servers. It allows the servers to manage only the tasks that they can handle such as query optimization, concurrency control, and recovery. There are two bases used to develop C/S frameworks. One is based on remote procedure calls (RPC). The other is based on object-oriented programming (OOP). Two well-known RPC-based C/S framework are the OSF's (Open Software Foundation) DCE (Distributed Computing Environment) and the SunSoft's ONC (Open Network Computing). There are three widely used OO-based C/S frameworks: Microsoft OLE (Object Linking and Embedding)/DCOM (Distributed Component Object Model), CIL's (Component Integration Laboratories) OpenDoc/DSOM (Distributed System Object Model), and OMG's (Object Management Group) CORBA (Common Object Request Broker Architecture). These competing standards are predicted to be merged to be COBRA-compliant [10].

2.2 End User Programming, Compound Document, and Component Software

[10] predicts the global marketsize of the distributed industry software for the next decade will reach 40 billions US dollars. These types of software will represent 50% of the industry's information technology procurement. Three emerging technologies lead the development of distributed software. They are the end-user programming, compound documents, and component software. End-user programming uses object-oriented application generators to quickly create user-centric applications. Compound document technology transforms the way software applications and switch from language-centric to document-centric. Comround document allows heterogeneous data to be linked

and used with distributed programs. Component software addresses the general problem of designing systems from application elements that were constructed independently by different developers using different languages, tools, and computing platforms. The goal is to have end users and developers enjoy the same high levels of plug-and-play application interoperability [2] [3]. AIS is an example of component software to be manufactured with other business systems if AIS is created with concepts of distribution and interoperability.

2.3 Relational Systems

Relational systems dominate the database marketplace. Oracle, Informix, and Sybase are the key players. Most business applications are built on top of the relational database management systems (RDBMS). The reason has been that business forms, reports, and queries are in tabular format which is natural in RDBMS. Relational systems are defined in [5] as (1) the data is perceived by the user as tables, and (2) the operator at the users disposal are operators that generate new tables from old and those operators include at least SELECT, PROJECT, and JOIN. And, online transaction processing (OLTP) applications run well on RDBMS. However, RDBMS is extended to meet the new requirement of online analytical processing (OLAP) application systems. Another reason of the use of RDBMS is the entity integrity and referential integrity provided by relational systems. The concept of primary key and foreign key link related tables together comes natural to business applications. AIS will be designed based on relational model.

3. Accounting Information Systems

3.1 Subsystems Description

We base our design of AIS on [6] and [8]. In general, AIS is divided into five subsystems. They are the sales subsystem, subsystem, pruchase subsystem, inventory subsystem, human resources subsystem, and general ledger subsystem. These represent the main functions in an enterprise. According to [6], the sales subsystem is to create information flows that support (1) repetitive work routines of the sales order department, the credit department, and the shipping department, and (2) decision needs of those who manage various sales and marketing functions. The purchase subsystem is to (1) handle the repetitive work routines of the purchasing department, the receiving department, the accounts payable department, and the cashier, (2) support the desision needs of those who manage the departments, and (3) assist in the preparation of internal and external reports. The inventory subsystem is to create information flows that (1) support the repetitive work routines of the inventory control department, (2) support the decision needs of those who manage the warehouse and the inventory control department, and (3) assist in the preparation of internal and external reports. The human resources subsystem performs the following functions that (1) captures, records, and stores data concerning personnel and payroll activities, (2) generates a variety of personnel and payroll forms and documents, (3) prepares a number of management reports, (4) prepares various governmental reports, and (5) assists in the preparation of external financial statements. And finally, the general ledger subsystem gives the following functions to (1) accumulate data, classify data by general ledger accounts, and recording data in those accounts, (2) fuel the financial reporting, managerial reporting, and other reporting subsystems for internal and external uses, (3) prepare general purpose, external financial statements, and (4) ensure that the external financial statements conform to GAAP (Generally Accepted Accounting Principles).

3.2 Data and Process Designs

[6] and [8] gives single-user and non-integral designs of AIS. We collect data from field practice, more trade literature, and integrate functions and information flows of each subsystem. We use the entity relationship diagram (ERD) to show the tables, relationships, and cardinality. We show the main functions in a top-down menu chart.

The key to the integral design is the data integration. We need to integrate tables which are used across subsystems. First, the files needed in each subsystem are determined and described. Second, we point out the common area where more than one subsystem will use and process. This common area will be the integration point. As shown in Figure 3, in AIS, the area is the tables of customers, suppliers, inventory, employees, and journals, Customer table is used by the sales subsystem to check validity and update balance, and by the general ledger subsystem to check and update accounts receivables. Supplier table is accessed by the purchases aubsystem to check status and update balances, and by the general ledger subsystem to check and update accounts payables. Further, inventory table is used by multiple applications. Inventory table supports the sales order application, the shipping application, purchase order application,

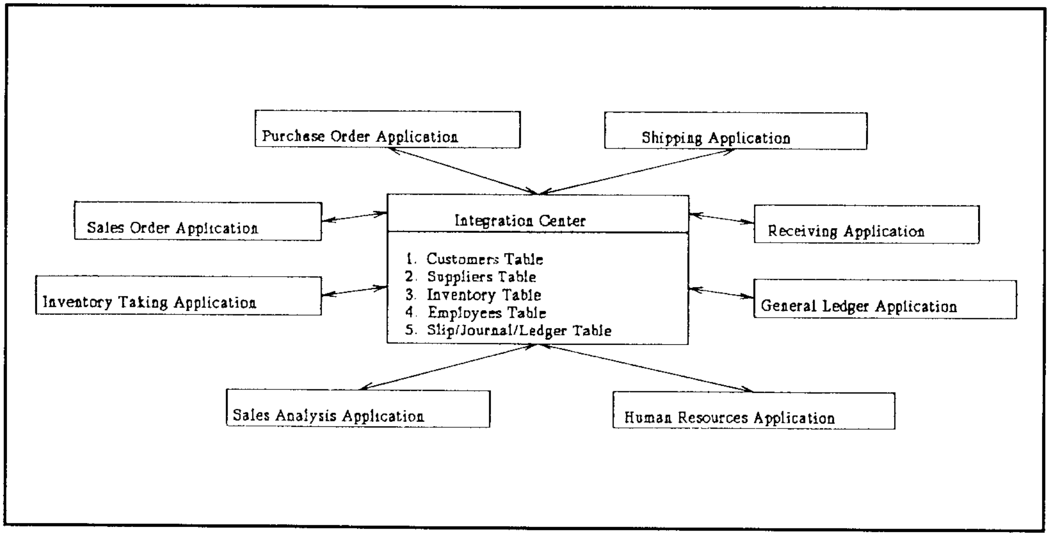
Purchase Order Application
Shipping Application
Sales Order Application
Integration Center
1. Customers Table
2. Suppliers Table
3. Inventory Table
4. Employees Table
5. Slip/Journal/Ledger Table
Receiving Application
Inventory Taking Application
General Ledger Application
Sales Analysis Application
Human Resources Application

Figure 3: Integration Center of AIS

receiving application, inventory taking application, and general ledger application. Employees table is in the similar situation where salesmen performance analysis, buyers/ agents performance analysis, and labor costs analysis share the data.

A. Sales subsystem

To fulfill the functional requirements of a sales subsystem, we design four main processes. They are (1) the sales order management which validates sales orders, picks inventory, and ships out orders, (2) the customers management which bills customers and processes customer payments, (3) the salesmen management to analyze salesmen performance in coperation with the personnel/payroll subsystem, and (4) the sales report management which produces sales analysis reports, salesmen performance report, and customer accounts aging reports. Figure 4 shows the sales management functions.

The data requirements are illustrated in Figure 5 where all necessary files are depicted. In sales subsystem, each sales order comes from one customer, and each sales order may contain one to many inventory items which are to be picked by warehouse people. Further, each customer will receive an invoice once the goods are shipped out, and one customer can receive multiple shipments and each shipment is billed with one invoice. More, when customers pay, each payment can pay for multiple invoices

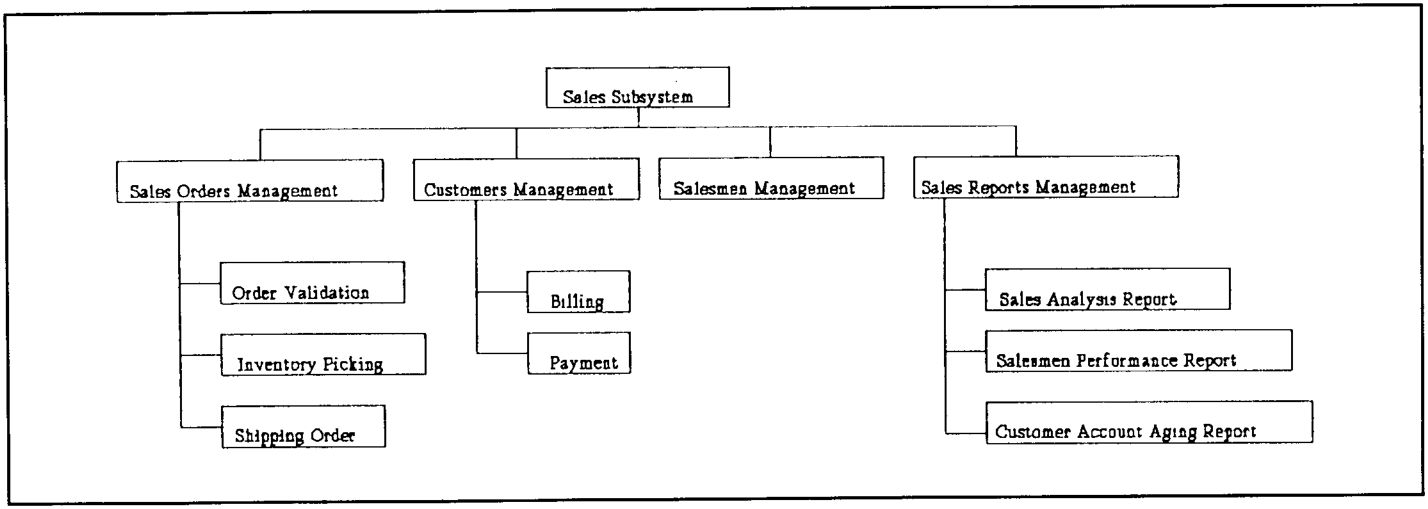

Figure 4: Main Process of Sales Subsystem

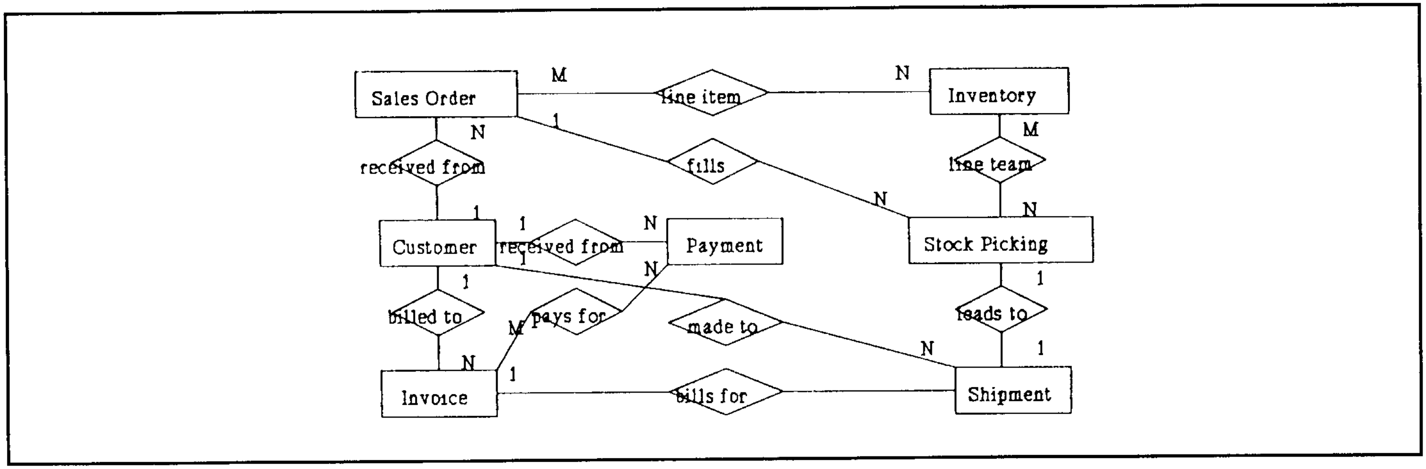

Figure 5: Entity Relationship Diagram of Sales Subsystem [6]

and each invoice can be paid in several payments. Based on the ERD, the ERD, the tables in sales subsystem include Sales-Order, Sales-Lines, Inventory, Stock-Picking, Customer, Sales-Invoice, Payment, Cash-Receipt, and Shipment. Table 1 gives the complete listing of all tables in AIS.

B. Purchases Subsystem

Main functions in the purchase subsystem include the purchase orders management, supplier management, buyers/agents management, and purchase report management. The purchase orders function is to prepare and issue purchase order, to receive and inspect coming goods, and to transfer goods to warehouse. We process vendor invoices and cash disbursements under the suppliers management function. The main reports from the subsystem include the purchase orders listing, the supplier status analysis report, and the cash discounts taken listing. Figure 6 shows the purchase functions.

The tables needed in the purchasessubsystem include the Purchase-Order, Inventory, Purchase-Lines, Inventory-Receipt, Suppliers, Cash-Disbursement, Vendor-Invoice, Disbursement-Lines. Purchase-Lines records the line item to be recorded, and stocked. The Disbursement-Lines records the line item that which purchased and received item is paid by which check on which invoice. Figure 7 shows the ERD of the purchase subsystem.

C. Inventory Subsystem

The inventory subsystem contains two main functions. They are (1) the inventory transaction management which handles the increments and decrements from sales and purchases, and the inventory taking, and (2) the inventory report management which gives the inventory detail listing and the inventory. Inventory is the key table of the subsystem. Figure 8 is a diagram of its functions.

D. Personnel/Payroll Subsystem

Human resources subsystem combines the personnel functions and the payroll functions. The personnel functions are to keep track of the employees basic information and activities. The payroll functions are to compute and record the attendance, salary,

Table 1: AIS Tables and Attributes

1. salesman (salesman_no salesman_name dept_no)
2. customer (customer_no customer_name contact title address telephone fax credit_limit balance)
3. sales (sales_order_no sales_date invoice_no customer_no tax_code terms total_amount salesman_no slip_no)
4. sales_lines (item_no sales_order_no sales_date item_description quantity price amount pick_no)
5. slip (slip_no slip_sequence slip_date account_no db_amount cr_amount remark)
6. purchases (purchase_order_no purchase_date requisition_no invoice_no supplier_no total_amount slip_no)
7. purchases_lines (item_no purchase_order_no purchase_date item_description quantity_ordered quantity_shipped cost amount)
8. inventory (item_no item_description unit location quantity_on_hand quantity_reserved quantity_back_ordered economic_order_quantity safety_stock average_cost retail_price)
9. supplier (supplier_no supplier_name contact title address telephone fax rating)
10. take_inventory (take_no item_no item_description take_date employee_no quantity loss gain slip_no)
11. salary (employee_no salary_code salary_description base_salary hour_rate commission_rate insurance_rate tax_rate vacation_rate sick_leave_rate overtime_rate)
12. employee (employee_no employee_name dept_no title rank gender hire_date termination_date date_of_birth education training experise language addredd telephone fax)
13. payroll (payroll_no employee_no payroll_day payroll_month payroll_year base_salary base_wage commission_pay overtime_pay tax insurance)
14. time_card (employee_no time_card_month time_card_year total_work_hours total_vacation_hours total_sick_leave_hours total_overtime_hours)
15. time_card_lines (time_card_no employee_no time_card_day time_card_month time_card_year time_in time_out work_hours vacation sick_leave overtime)
16. account (account_no account_name)
17. sub_account (sub_account_no account_no sub_account_name)
18. check (check_no check_date slip_no slip_sequence account_no supplier_no customer_no terms db_amount cr_amount automatic_slip remark)
19. closing (closing_no closing_date slip_no account_no sub_account_no db_amount cr_amount remark)
20. stock_pick (pick_no, pick_date, pick_by, sales_no)
21. inventory_lines (item_no, pick_no, quantity_picked)
22. shipment (ship_no, ship_date , shipped_by, pick_no, customer_no, invoice_no)
23. sales_invoice (invoice_no, ship_no, invoice_date, amount, customer_no)
24. cash_receipts (remittance_no, customer_no, amount, remittance_date, invoice_date)
25. inventory_receipt (receipt_no, receipt_date, received_by, purchase_order_no, vendor_no, invoice_no)
26. vendor_invoice (invoice_no, invoice,date, amount, vendor_no)
27. cash_disbursement (disbursement_no, vendor_no, amount disbursement_date)
28. disb_inv (disburdement_no, invoice_no, amount)

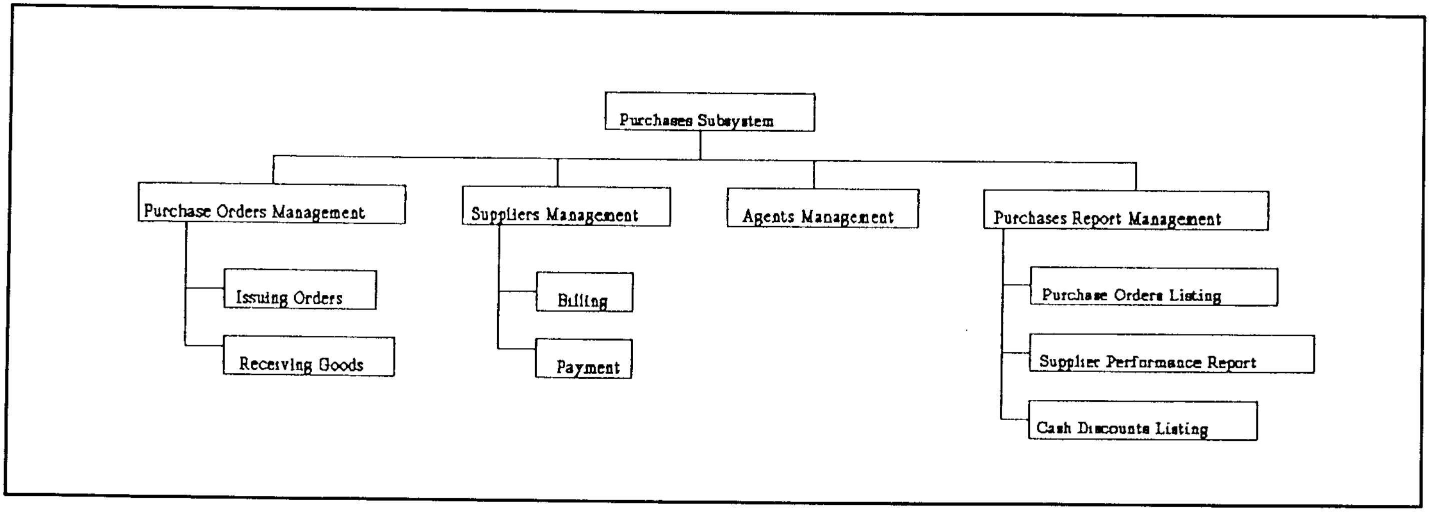

Figure 6: Main Process of Purchase Subsystem

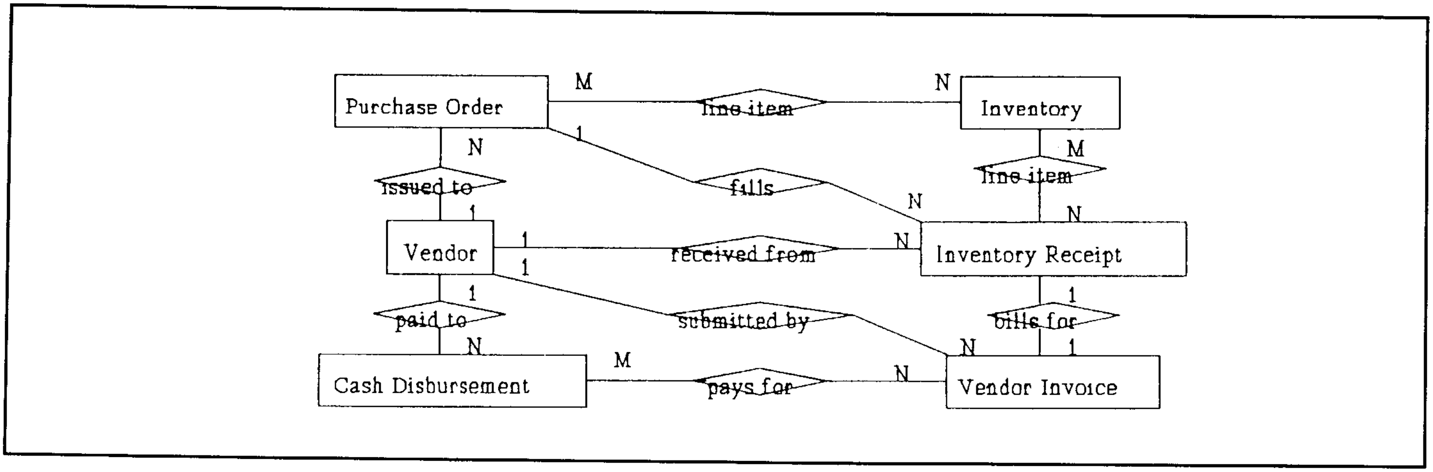

Figure 7: Entity Relationship Diagram of Purchases Subsystem [6]

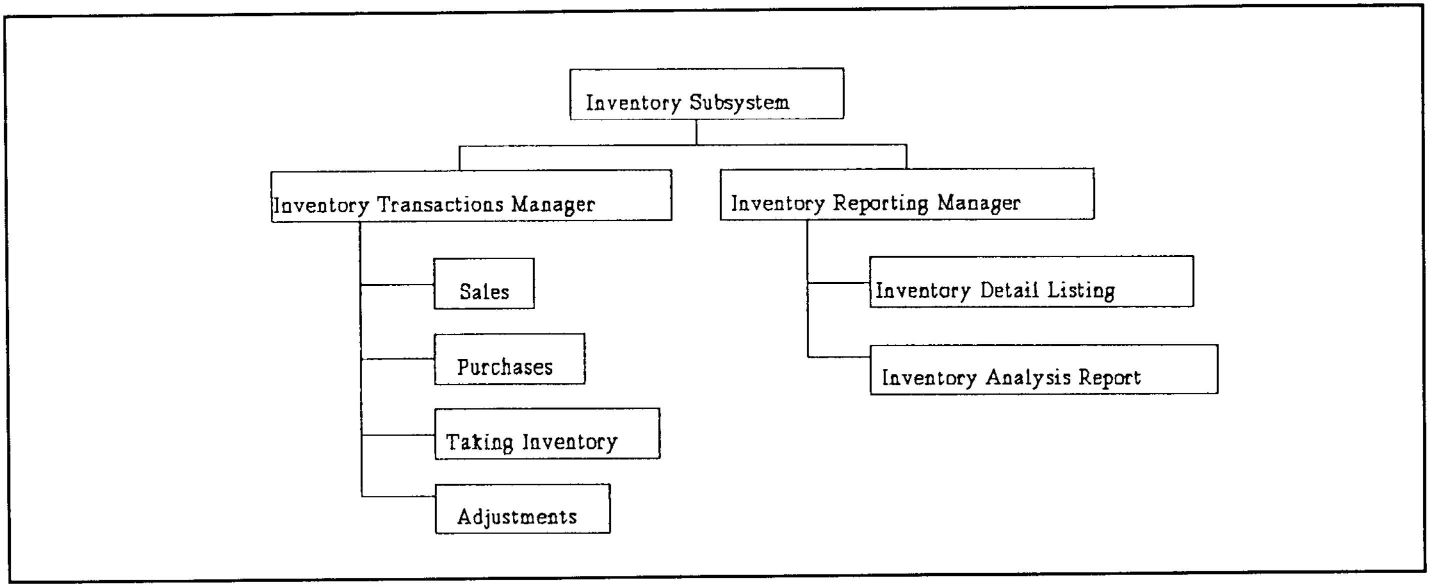

Figure 8: Main Processes of Inventory Subsystem

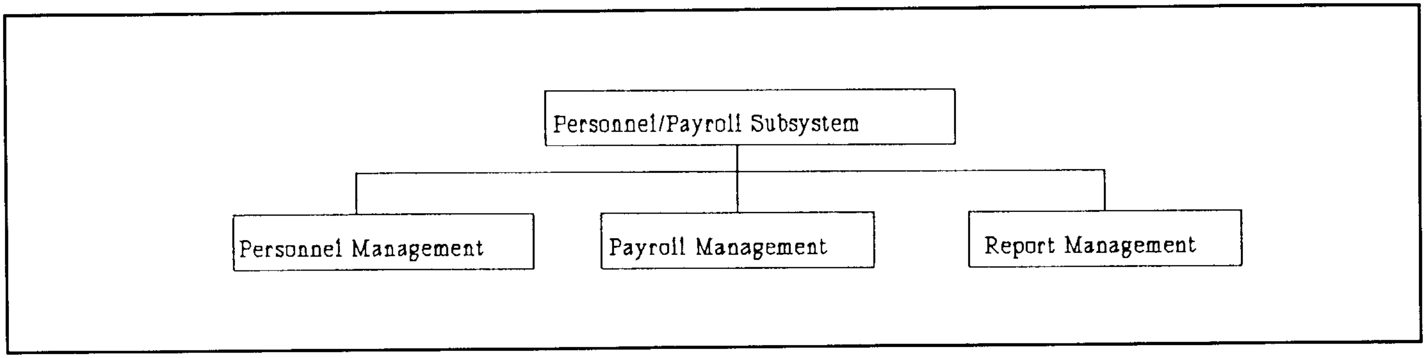

Figure 9: Main Processes of Human Resources Subsystem

and taxes for each active employee. As mentioned above, the Employee table is shared with the sales subsystem and the purchases subsystem. Other tables such as the Payroll file is to keep the basic salary/wage/overtime/addition/deduction/tax rates, the Salary file gives the computed salary for each employee, the Time-Card file Keeps track of monthly attendance with the Time-Card-Lines for daily attendance records. Figure 9 gives the functions chart.

E. General Ledger Subsystem

General ledger subsystem receives inputs from the above four systems and creates new journal entry into the Slip table. The subsystem does not have complex functions. However, it must support the bookkeeping cycle and is able to perform posting, trial balance, adjusting, closing, and reporting. That is why the subsystem keeps a set of four tables to meet the data requirements of the accounting cycle. These tables tables are the Slip table, the Check table to work with the Suppliers and the Customers tables, the Account and Subaccount tables to keep the account numbers and account names, and the Closing table to record the accounts information and balances after closing. Financial reports include the balance sheet, income statement, cash flow analysis report, and annual revenues/expenditures analysis report. General ledger subsystem is shown in Figure 10.

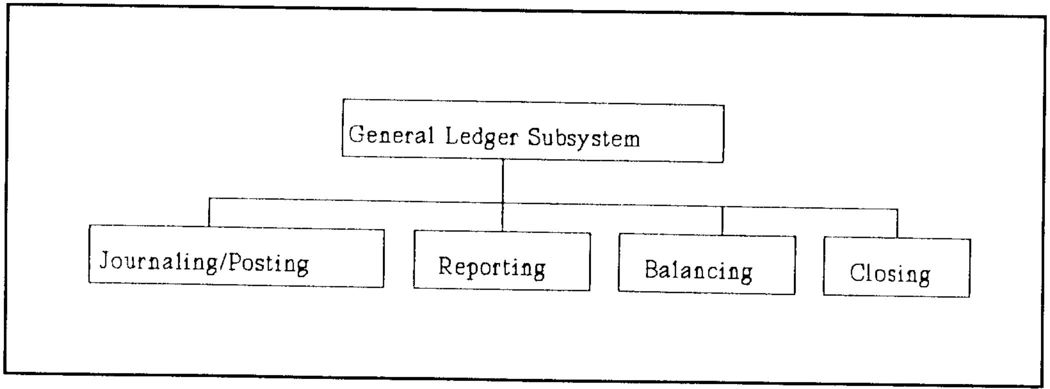

Figure 10: Main Processes of General Ledger Subsystem

F. Prototype

An implementation of the design is conducted in an MIS Senior Project Seminar. We had three semesters to familiarize ourselves with the AIS concepts, development platform and tools, and create the integral design, coding, testing, and implementing. We used the Microsoft Visual Basic 4.0 and Microsoft SQL Server/MT on Pentium PCs. The prototype follows the design closely. Four student teams were assigned and responsible to lead one subsystem development. AIS database is located on SQL Server. AIS Clients were Microsoft Windows 95 and ran OFFICE 95 and Internet applications. Figure 11 shows the prototype architecture.

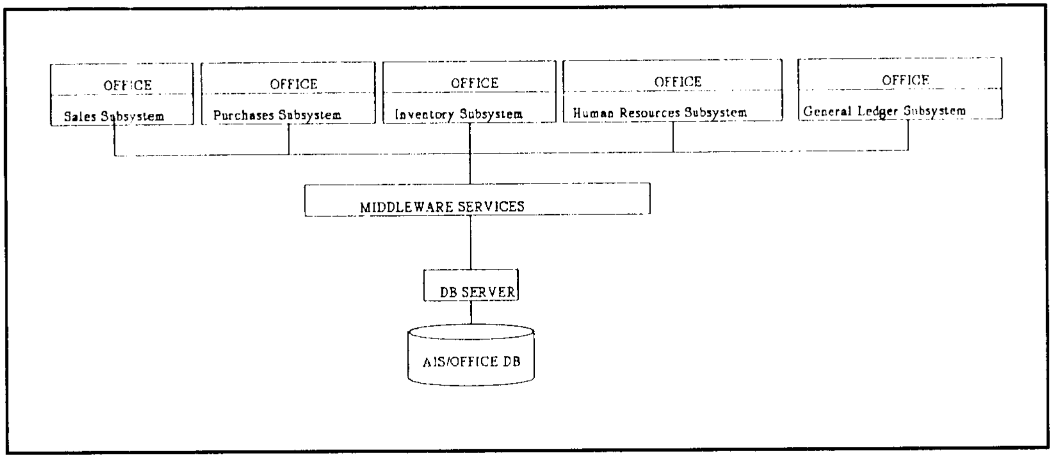

Figure 11: An Integral Design of Client/Server Accounting Information System

4. Summary

In this paper, we have described an integral design of an accounting information system on the client/server platform. AIS is a foundation of the enterprise-wide information technology applications. The goal is to demonstrate the interoperability, scaleability, and portability of an important business application which can collaborate with other applications and databases in use of distributed computing.

Acknowledgement

Special acknowledgement is given to the 18 MIS seniors of Class of 1997 who have taken my 3-semester MIS Project Development Seminar class between the fall 1995 and the fall 1996. They have worked diligently and intelligently to implement this design and have successfully presented the completed system prototype in our annual departmental exhibition.

References

[1] Accounting Research Journal, special issue on accounting information systems in small businesses in Taiwan, Chinese Edition, October 1996.

[2] Adler, R.M., "Emerging Standards for Component Software," IEEE Computer, March 1995, pp. 68-76.

[3] Adler, R.M., "Distributed Coordination Models for Client/Server Computing," IEEE Computer, April 1995, pp. 14-22.

[4] Bernstein, P.A., "Middleware: A Model for Distributed System Services," CACM, vol. 39, no. 2, February 1996, pp. 86-98.

[5] Date, C.J., An Introduction to Database Systems, 6th Ed., Addison Wesley Co., 1995.

[6] Gelinas, U.J. and Oram, A.E., Accounting Information Systems, 3rd Ed., South Western Co., 1996.

[7] Hart, J.M. and Rosenberg, B., Client/Server Computing for Technical Professionals, Addison Wesley Co., 1995.

[8] Institute of Information Industry, Accounting Information Systems Development Practice, Vol. I,II,III, Chinese Edition, 1992.

[9] Khoshafian, S., Chan, A., Wong, A., and Wong, H., Client/Server SQL Applications, Morgan Kaufmann Co., 1993.

[10] Lewis, T. G., "Where is Client/Server Software Headed?" IEEE Computer, April 1995, pp. 49-55.

[11] Wood, A., "Predicting Client/Server Availability," IEEE Computer, April 1995, pp. 41-48.