A Computing Model for Distributed Processing Systems and Its Application

Yoshitomi Morisawa[1, 2], Hisashi Okada[1], Hiromichi Iwata[1] and Haruo Toyama[1]

[1] Nihon Unisys, Ltd.
1-1-1 Toyosu, Koto-ku
Tokyo 135-8560, Japan
+81 3 5546 4111
Yoshitomi.Morisawa@unisys.co.jp

[2] Graduate School of Information Science,
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma,
Nara 630-0101, Japan

ABSTRACT

When implementing an application system in a distributed computing environment, several architectural questions arise such as, how and where computing resources are distributed, and how the communication among computing resources should be implemented. To simplify the process of making these choices, we have developed a distributed computing model. This model classifies distributed processing systems into seven categories based on the location of data storage and the style of processing between client and server. This paper describes our model and its use in planning the infrastructure of a new system for one of our customers.

1 INTRODUCTION

Currently, there are many software products for implementing Client/Server (C/S) systems available commercially. When a user implements an application utilizing these products, it is common to spend a large amount of time and effort testing the interconnectivity and interoperability, known as conformance testing, among the products to be used. At Nihon Unisys, Ltd.(NUL), we have developed a body of expertise and experience in conformance testing by submitting proposals to customers, providing consultation in Information Technology, and supporting the implementation of our customers' applications. In order to share this expertise and experience as well as reduce the cost of developing new customer applications, we needed to create a framework which includes: a software architecture for implementing C/S systems; an intuitive and simple model of distributed processing, including C/S processing; and combinations of products based on this model. NUL named this framework the Open Solution Framework (OSFW) and the distributed processing model the Client/Server Solution (C/SS) model, and announced it in January 1996.[1-2]

When implementing an application system in a distributed computing environment, several architectural questions arise such as, how and where computing resources are distributed, and how the communication among computing resources should be implemented. A distributed computing model represents a typical configu-ration of computing resources and intercommunication among them. It is useful in answering these questions and in supporting the implementation of application systems. A distributed computing system can be viewed as a multi-tiered environment from the perspective of hardware platforms and as cooperative client/server processing from the viewpoint of functional distribution.[3] An example of the former is the traditional three-tiered platform model that consists of mainframe hosts, servers, and workstations. The three-tiered model has traditionally been used by mainframe vendors to position their mainframe in an information system. Other vendors have used this view to position enterprise level servers and divisional level servers. The cooperative client/server view shows the distribution of presentation functions, application logic and data between client and server. This is the view discussed in this paper.

The main motivation for introducing the C/SS model was to provide a classification of the processing style of C/S systems, to provide proven software tools, called product sets, for implementing C/S systems in each style, to reduce the total system integration cost, and to construct stable systems with lower cost in a shorter time. To achieve the above goals, we developed the OSFW as a framework of C/S system architecture and are using the C/SS model as its kernel reference model. Product sets are periodically revised to reflect latest technology and used with this model for configuring our customers' services.

This paper discusses another motivation for introducing the C/SS model, to use it as a reference model for designing the infrastructure of customer information systems under our Information Technology Infrastructure Planning (ITIP) methodology. This paper reports the use of the C/SS model to design the information infrastructure of a new financial affairs system for a Japanese company. We present the computing model for distributed processing systems, in particular for cooperative client/server processing, and explain our experience in using the model.

2 RELATED WORK

Alex Berson classified cooperative processing systems into five styles: Distributed Presentation, Remote Presentation, Distributed Business Logic, Remote Data Management and Distributed Data Management.[3] The Gartner group has also defined five styles of C/S computing which are very similar to Berson's model and are using the model in their research reports and conference presentations for aiding in the discussion of C/S applications.[4] This model is based on the distribution points of presentation functions, application logic functions, and data management functions. However, this model does not account for asynchronous processing, making it difficult to model groupware applications and E-mail type applications. This is shown in Figure 1.

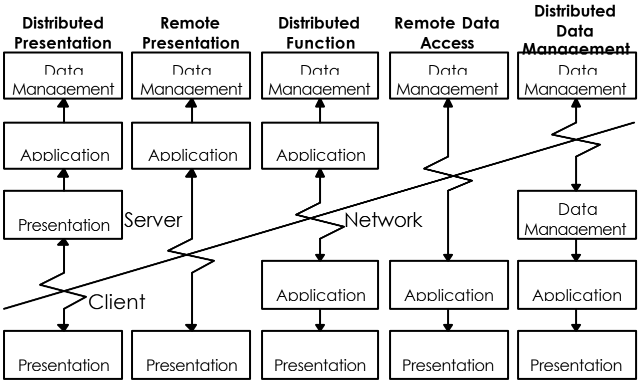

Fig.1 Distributed Computing Model

IBM Corporation classified C/S systems into six templates in their guidance for C/S systems. They are: Front-ending, Resource Centric, Host-Distributed Logic, LAN-Distributed, Data Staging, and Multi-Application. These templates are used as a reference model in C/S engineering.[5] The templates were the result of the survey of approximately 50 real-world solutions designed or implemented by IBM's typical customers until the early in 1990s.[6] However, the criteria for applying these templates are not clear to users and the template does not guarantee coverage of all of type of C/S systems.

Our C/SS model was developed to overcome the weakness in these two models by adding asynchronous processing introducing viewpoints to make the model more understandable.

3 CLIENT/SERVER SOLUTION MODEL

When developing a C/S system, it is necessary to decide the location of data storage based on its administration, its security, and its computer facilities. It is also necessary to decide the processing style between original data in a client and its data storage in server(s). These decisions are required relatively early in the design phase of practical system implementation. Therefore, we classified C/S systems into seven basic models from the viewpoints of the location of data storage and the processing style between client and server. Data location is classified as centralized or distributed. Distributed data is further classified as vertical, e.g. divisional and enterprise data, and horizontal, e.g. data from different divisions. This view is easy to understand for and is using by field system engineers implementing practical systems. The processing style between client and server is classified as synchronous or asynchronous. Synchronous processing is divided into two categories, Transaction and Request/Reply, depending on the characteristics of the messaging between the client and the server. This classification scheme led us to develop seven models of distributed processing which are illustrated in Table 1. The design goal of this model is that it provides an intuitive, easy and simple model for usual field system engineers. Collectively, these seven models are known as the C/SS model.

Table 1 C/SS Model

Data Style / Processing Style	Centralized	Distributed	
		Vertical	Horizontal
Transaction Processing	C/SS Model-1	C/SS Model-2	
Request/Reply Processing	C/SS Model-3	C/SS Model-4	
Delayed Processing	C/SS Model-5	C/SS Model-6	C/SS Model-7

An actual C/S system may be constructed using a combination of one or more of the basic models. This classification introduces the concept of time and vertical/horizontal data distribution in the conventional client/server model. As a general rule, we presume that "Presentation" is in a client side and "Data" is in a server side.

The meanings of the terms used in Table 1 are as follows.

- "Data style" indicates the location of the data storage used. However, personal data or personal database is not included.
- "Centralized" means that the data is stored in only

one place.

- "Distributed, Vertical" means that the data is vertically distributed along an organizational hierarchy. This style focuses on the differences in the processing power of the servers.
- "Distributed, Horizontal" means that the data is horizontally distributed along a horizontal spread of organizations. In this style, data is distributed into multiple servers which have an equivalent level of processing power.
- "Processing Style" indicates the processing style between client(s) and server(s).
- "Transaction Processing" is transaction processing with ACID characteristics, i.e., Atomicity, Consistency, Isolation and Durability.
- "Request/Reply Processing" indicates that a reply from the server is synchronized with a request from a client.
- "Delayed Processing" indicates that the server process is not synchronized with a client request.
- The followings are symbols used in this paper.
  - P: Presentation
  - AP: Application Logic
  - DM: Data Management
  - D: Data

The following sections provide a description of each model.

3.1 Model-1: Centralized Transaction Processing Model

There is a single database held centrally on a single server and the style of processing between a client and the server is Transaction Processing with ACID characteristics.

The typical structure of this model is shown in Figure 2.

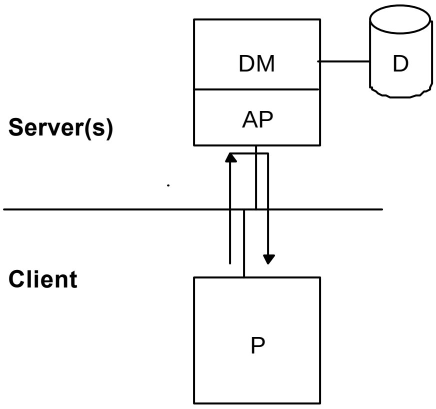

Fig.2 Centralized Transaction Processing Model

An explanation of message and process flow: P on a client sends a transaction to AP on a server. According to the result of AP, DM retrieves, updates, adds, and/or deletes data in D. Then, the result of the process is returned to the client. AP and DM may be on different servers.

This model is suited to transaction processing which uses a transaction management program and usually adds and updates data in a single database. When the transaction has a heavy load, a server side may be divided into an application server and a database server. Examples of this model include an order entry system, a stock management system, a stock ordering system, a production management system, a retail POS system, etc..

3.2 Model-2: Distributed Transaction Processing Model

There are multiple databases on multiple servers and the style of processing between a client and the servers is Transaction Processing with ACID characteristics.

The typical structure of this model is shown in Figure 3.

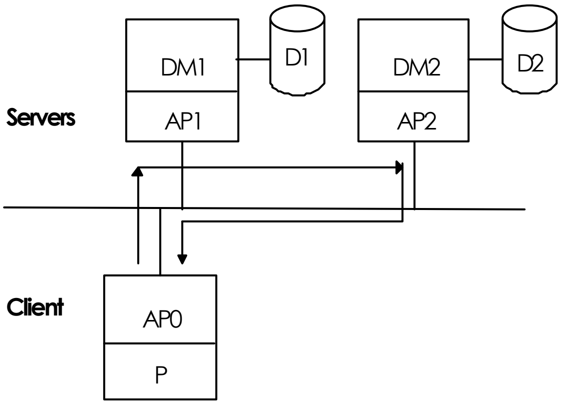

Fig.3 Distributed Transaction Processing Model

An explanation of message and process flow: AP0 on a client sends a transaction to AP1. Using a part of the processing result of AP1, DM1 retrieves, updates, adds, and/or deletes data in D1. Then, the part of this transaction that was not processed, or the processing result of AP1, is sent to AP2. According to the processing result of AP2, DM2 retrieves, updates, adds and/or deletes data in D2. Then, the processing result is sent back to the client. If required, a mechanism is used to maintain the integrity of D1 and D2. There may be different servers for APx and DMx.

This model is suited to transaction processing for a core business which involves complicated update and query of multiple databases. When there is a heavy transaction load, the server side may be divided into an application server and a database server. In the case of asynchronous processing, if message queuing is used by the transaction management program, it is classified as Model-5, Model-6 or Model-7. Examples of this include a core business system of banking, a seats reservation system, a factory production management system, etc..

3.3 Model-3: Remote Data Processing Model

There is a single database held centrally on a single server and the style of processing between a client and the server is Request/Reply.

The typical structure of this model is shown in Figure 4.

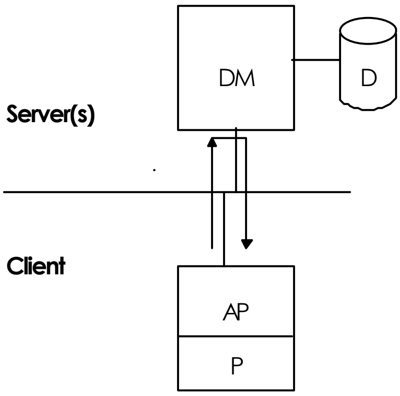

Fig.4 Remote Data Processing Model

An explanation of message and process flow: AP on the client side sends a message(SQL, etc.) to DM. According to the requirements represented by the message format, DM retrieves, updates, adds, and/or deletes data in D. Then, the processing result is sent back to the client.

This model is suited to End User Computing (EUC) such as in a decision support system. AP functions with heavy load and AP facilities common to multiple clients may be put onto a server. Examples of EUC are various statistics, analysis and reporting, such as in budget planning, financial analysis, market research and analysis, sales analysis, capacity planning, demand forecast, etc.. Examples of query and reply processing include customer services, sales support, various inquiry processing, information providing services, etc..

3.4 Model-4: Distributed Data Processing Model

There are multiple databases on multiple servers and the style of processing between a client and the servers is Request/Reply.

The typical structure of this model is shown in Figure 5.

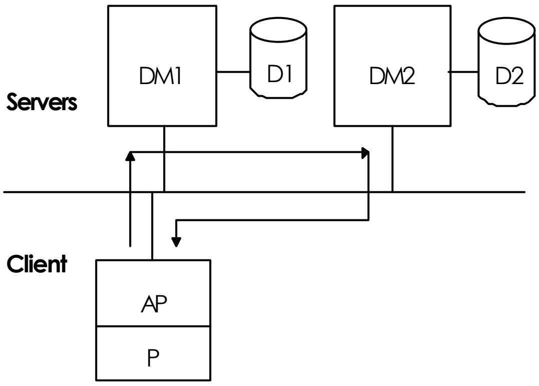

Fig.5 Distributed Data Processing Model

An explanation of message and process flow: AP on the client sends a message(SQL,etc) to DM1. According to (a part of) the requirements of the message, DM1 retrieves, updates, adds, and/or deletes data in D1. The part of the message not processed, or the message of the processing result, is sent to DM2; then, DM2 retrieves, updates, adds, and/or deletes data in D2. The processing result is sent back to the client. If required, a mechanism is adopted to maintain the consistency of D1 and D2.

This model is suited to EUC with simultaneous access to multiple databases and files and to the inquiry-intensive immediate processing. This enables effective utilization of information by sharing existing databases. This includes almost all of applications of Model-3. Other applications are an enterprise sales statistics, an enterprise productivity data analysis, etc..

3.5 Model-5: Centralized Messaging Model

There is a single database held centrally on a single server and the style of processing between a client and the server is Delayed Processing.

The typical structure of this model is shown in Figure 6.

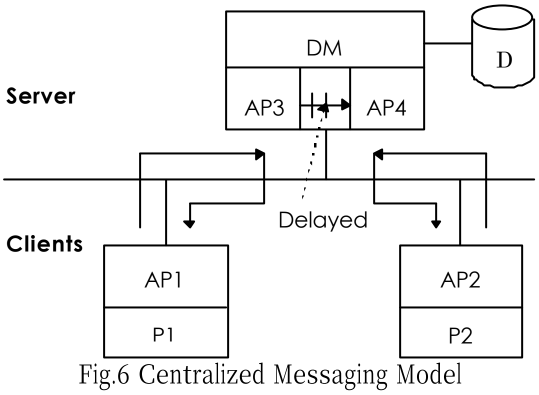

Fig.6 Centralized Messaging Model

An explanation of message and process flow: AP1 on a client sends a message(SQL, document, etc.) to AP3. According to the result of the message, DM retrieves, updates, adds, and/or deletes data in D. After the modification of D, with appropriate timing, AP2 on the other client sends a message to AP4 and its message is processed using D, which has been modified.

This model is suited to the automation of simple workflow within a group or an organization. Examples of this application include shipping and forwarding of internal memos and documents, events notification, internal document filing, execution / monitoring / reporting of business flow, etc..

3.6 Model-6: Data Staging Model

Multi-tiered servers contain vertically distributed databases and the style of processing between a client and the server is Delayed Processing.

The typical structure of this model is shown in Figure 7.

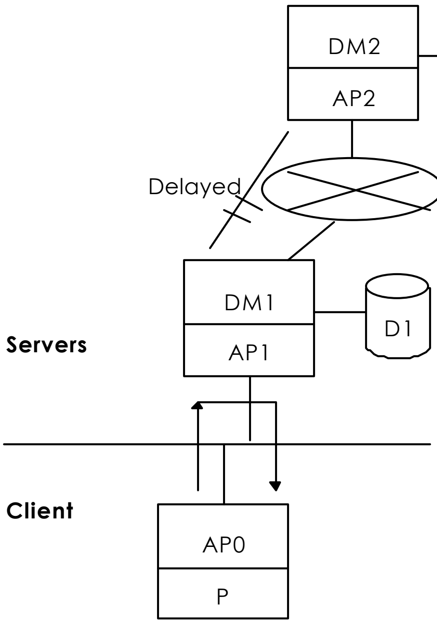

Fig.7 Data Staging Model

An explanation of message and process flow: AP0 on the client sends a message(SQL, etc) to AP1. According to the requirements of the message, DM1 retrieves, updates, adds and/or deletes data in D1. A new message generated by the process of AP1 is sent to AP2. Then, the processing result is sent back to the client of AP0. At appropriate timing, such as kicking from AP1, AP2 and DM2 on another server retrieve, update, add, and/or delete data on D2 depending on the message already sent by AP1.

Typical utilization of this model is in an application such that data in D1 on a lower server is updated (down loaded) using a part (or all) of the data in D2 on an upper server; or, in the opposite way, data in D2 is updated (up loaded) using data in D1.

PDA and mobile computing (which is called a client of client) are treated as a moving object (child client) and are integrated with a parent of the client at a suitable time. These are included in this model.

This model is suited to an application to promote the management and utilization of information by asynchronously sharing of data. Examples are:

1) Asynchronous update of replicated data such as a personal information system or accounting system across multiple branches.
2) Information utilization by downloading a part of database in business domains such as DSS and EIS using Daifukucho (an old-fashioned account book in Japan) and multi-dimensional databases.
3) Uploading of transaction data collected at branches or departments, such as combining the uploading of order data collected by departmental servers and the centralized order processing on an enterprise server.

3.7 Model-7: Chained Messaging Model

Multiple servers contain horizontally distributed databases and the style of processing between a client and the server is Delayed Processing.

The typical structure of this model is shown in Figure 8.

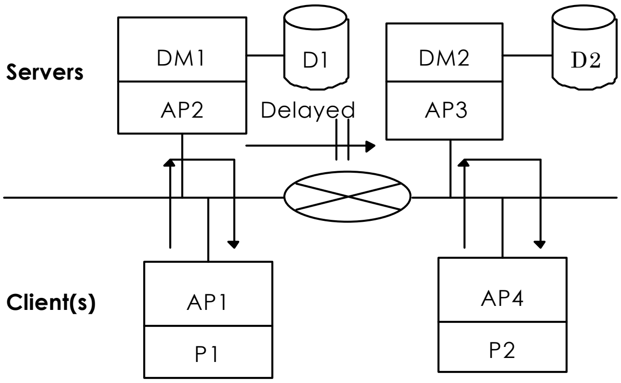

Fig.8 Chained Messaging Model

An explanation of message and process flow: AP1 on the client sends a message(SQL, etc.) to AP2. According to the requirement of this message, DM1 retrieves, updates, adds, and/or deletes data in D1. A new message generated by process AP2 is sent to AP3. The processed result then is sent back to the client of AP1. At appropriate timing such as kicking from AP3, a message is sent from AP4 on another client to AP3, and DM2 retrieves, updates, adds, and/or deletes data on D2 depending on the message already sent by AP2. The processed result is sent back to AP4 for the other client.

This model is suited to the loose integration by cooperations of independent multiple applications or systems. Examples of applications are:

1) Workflow between groups or organizations

- Shipping and forwarding of internal memos and documents, and event notification by E-mail
- Forwarding and authorization of internal Ringi (Circulating the draft for sanction)

2) Integration of cooperating applications between enterprise systems
- Integration of production planning, Stock management, Process control, etc. using message delivery
- Integration of orders, buying, stocks, shipping and billing systems.

3) Cooperating systems between enterprises such as EDI
- Order entry system cooperating with the customer system
- Parts supply management system cooperating with suppliers' systems
- Packed travel reservation system cooperating with related companies' systems

4 EXPERIENCE IN USING C/SS MODEL

4.1 System Requirements of S Company

S company is a construction enterprise that manages construction activities from Hokkaido Prefecture to Okinawa Prefecture in Japan as well as overseas. The goal of this project was to develop a new financial affairs system for domestic use in Japan (approximately 2,000 sites).

Each site is a temporary work office which is set up at the beginning of a construction project and broken down when construction is completed. Half of these temporary work offices are set up and broken down in a year. To manage these remote sites, 1-3 permanent business offices exist in each prefecture. These offices are organized under branch offices of which there are 14 nationwide. These various offices, together with the head office, accomplish the accounts business function.

To take advantage of local availability and pricing, each local work office is responsible for contracting and paying for all materials such as steel frames and fresh concrete, labor costs of steeplejacks and carpenters. Similarly, all accounting information such as orders, order reports, various transfers, etc. which occur in a work office are to input at the local work office, as well.

4.2 Using the C/SS Model

The overall process used in the ITIP methodology to select the recommended C/SS model is:

1) Clarify the data and its administrative organization.
2) Put necessary business processes in order.
3) Describe both a centralized and a distributed data arrangement. Both descriptions are needed to help participants from users' divisions understand the proposed solution since they are usually not familiar with computer systems.

3-1) Clarify C/SS model of each business process for the centralized data arrangement.

3-2) Clarify C/SS model of each business process for the distributed data arrangement.

4) Decide centralized or distributed data arrangement from the points of administration, security, and business requirements.
5) Select simple models under considering its possibilities and stability.

4.3 An Example: New Financial System

The following example illustrated how the C/SS model for the new financial affairs system for S company was derived. We explain only a part of its derivation due to the limited space in this paper.

1) The main data and the usage of data within each organization is classified. Data include entry data, master data, etc. and organization of work offices, business offices, branch offices, the head office, and their relationships.
2) The following business processes are selected.
   - Data-entry process
   - Branch processes (Accounts process of work office, Accounts process of branch, Monthly balance process)
   - Inter-branches process (Transfer process)
   - Enterprise process (Settlement process)
   - Supporting processes (Master data maintenance, etc.)
3) The centralized and distributed data arrangements are clarified. From the points of data administration, data used are as follows and shown in Figure 9.
   - Enterprise Data and Enterprise Common Data (various master data) are centralized.
   - Branch Data are horizontally distributed.
   - Enterprise Data, Enterprise Common Data and Branch Data are vertically distributed.
   - Entry Data are horizontally distributed.
   - Branch Data and Entry Data are vertically distributed.

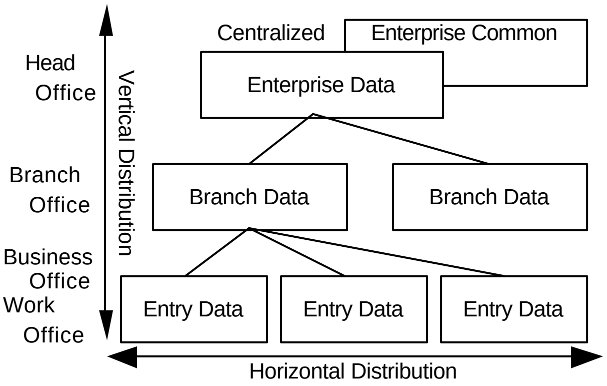

Fig. 9 Relationship between Data

Table 2 indicates the physical arrangement of these data in Figure 9 when they are in centralized arrangement and distributed arrangement.

Table 2 Physical Data Arrangement

	Centralized	Distributed
Head Office	Enterprise Data Enterprise Common Data Entry Data transferred	Enterprise Data Enterprise Common Data
Branch Office		Branch Data Enterprise Common Data(copy) Entry Data transferred
Business Office Work Office	Entry Data	Entry Data Enterprise Common Data(copy)

Using the following premises, we identified the C/SS model for each business process in both of centralized arrangement and distributed arrangement.

- A data entry terminal in a work office is not established permanently, so it is difficult to connect always it into a network. On the other hand, it is required to have the availability for an irregular data entry process in a work office. This process is closer the mobile computing which is included in Model-6: Data Staging model.
- We applied the following criteria for selecting Transaction Processing and Request/Reply Processing.
  - Transaction Processing if there is WAN between a client and a server.
  - Request/Reply Processing if there is no WAN between a client and a server.
- A single database would be large enough to accommodate the assumed quantity of data and transactions. Therefore, when data is centralized in the head office, we assume that it is a single database.

3-1) The C/SS model for each business process in a centralized data arrangement is identified.

- Data-entry process falls under Model-6 because data is vertically distributed in work offices, business offices and a branch office, and processing style is Delayed Processing for uploading data to the head office.
- Branch processes fall under Model-1. These processes are query processes that a client in a branch office queries data centralized in the head office, and processing style is Transaction Processing because there is WAN between a branch office and the head office.
- Inter-branches process falls under Model-1. This process is a query process of data related branch offices. Data is centralized in the head office, and processing style is Transaction Processing.
- Enterprise process falls under Model-3. This process is a query process that a client in the head office queries Enterprise Data centralized in the head office, and processing style is Request/Reply Processing.
- Supporting processes fall under Model-1. These processes are real time process for master file maintenance centralized in the head office from branch offices, and processing style is Transaction Processing.

3-2) The C/SS model for each business process in a distributed data arrangement is identified also.

- Data-entry process falls under Model-6 because data is vertically distributed in work offices, business offices and a branch office, and processing style is Delayed Processing for uploading data to the branch office.
- Branch processes fall under Model-3. These processes are query processes that a client in a branch office queries Branch Data centralized in the branch office, and processing style is Request/Reply Processing.
- Inter-branches process falls under Model-2. This process is a query process that a client in a branch office queries Branch Data horizontally distributed in other branch offices, and processing style is Distributed Transaction Processing.
- Enterprise process falls under Model-3. This process is a query process that a client in the head office queries Enterprise Data centralized in the head office, and processing style is Request/Reply Processing.
- Supporting processes are divided into two categories. One is master file maintenance and other is deployment of Enterprise Common Data.

The processes of master file maintenance are query processes of Enterprise Common Data centralized in the head office from branch offices. Processing style is Transaction Processing. Therefore, these processes fall under Model-1.

The processes of deployment are Delayed Processing for downloading Enterprise Common Data to branch offices, business offices and work offices from the head office. Therefore, data is vertically distributed. These processes fall under Model-6.

4) Due to the following requirements, it was decided that a centralized data arrangement is the best for new financial affairs system.

- The new data entry load on operators could be no more than with the current system.
- Security is important, including protecting against loss or theft in the work place.
- Data input from local sites must be checked against master data to insure integrity.

5) The enterprise process, which will be used by clients within the head office, may use Model-1 and is not restricted on Model-3. Therefore, Model-1: Centralized Transaction Processing model and Model-6: Data Staging model were selected as infrastructure models for the new financial affairs system.

Based on this analysis, we proposed a set of recommended software products necessary for each model and listed the necessary hardware configurations in our proposal.

Our proposal was selected from the following set of recommended software products from the product sets of each C/SS model and additionally included administration products:

For Model-1: VisualBasic for a client application, BEA Tuxedo as a transaction monitor and Oracle for database software on a server,
For Model-6: Oracle Mobile Agents.

Hardware products were selected the following ones:

For servers: Sun Ultra Enterprise 4000 and COMPAC PC servers,
For clients: DOS/V PCs and mobile facilities.

S company has selected the information infrastructure described in our proposal, and is going proceeding to the application development.

5 CONCLUSIONS

NUL has been putting the C/SS model to practical use in selecting a combination of products for implementing client/server systems since the announcement of OSFW in January 1996. Since the available products are evolving rapidly, recommended combinations of products are reviewed as the technology progresses. We are also using the C/SS model as a reference model to develop an information infrastructure in our IT consulting business. This paper has described one of such applications. The responses to this model from our customers and our system engineers have been positive as a reference model to support design stage of business applications.

However, the OSFW, which contains the C/SS model discussed in this paper, doesn't currently cover all application areas. Batch processing and large volume printing, which are important with current information systems and will continue to be important in the future, are not specifically treated in this framework at this time. Currently, we are discussing the enhancement of OSFW to cover these applications also.

ACKNOWLEDGMENT

OSFW is the fruition of discussion with many colleagues inside of NUL. The authors wish to acknowledge many colleagues who have participated in the discussion from the early stage of this framework, and Dr. Lloyd G. Williams and Dr. Koji Torii for reviewing the draft of this paper.

RERERENCES

1. Yoshitomi Morisawa, Hiromichi Iwata, Haruo Toyama: A Proposal of Computing Models for Distributed Processing Systems, *IPSJ-SIG Software Engineering*, 96-SE-109-3, May 1996, in Japanese
2. Yoshitomi Morisawa, Hiromichi Iwata, Haruo Toyama: Open Solution Framework for Implementing Client/Server Systems, *Unisys Technology Review*, Vol.16 No.2, pp.15-33, August 1996, in Japanese
3. Alex Berson: Client/Server Architecture, *McGraw-Hill*, 1992
4.James Cassell: The Total Cost of Client/Server: A Comprehensive Model, *A Gartner Group Conference on the Future of Information Technology Industry*, November 10-11, 1994.
5. A Guide to OPEN CLIENT/SERVER, *Open Enterprise group of IBM Europ in Basingstoke UK*, May 1994
6. J.J.Shedletsky and J.J.Rofrano: Application reference designed for distributed system, *IBM Sys.J.*, Vol.32, No.4, 1993