

Layered Design and Collaborative Development of Network System and Its Application in Real-time Information Cluster

Qiudong Sun

School of Electronic and Electrical Engineering,
Shanghai Second Polytechnic University, Shanghai 201209, China
Email: qdsun@ee.sspu.cn

Wenxin Ma

School of Electronic and Electrical Engineering,
Shanghai Second Polytechnic University, Shanghai 201209, China
Email: wxma@ee.sspu.cn

Yongping Qiu

School of Electronic and Electrical Engineering,
Shanghai Second Polytechnic University, Shanghai 201209, China
Email: ypqiu@ee.sspu.cn

Wei Chen

School of Electronic and Electrical Engineering,
Shanghai Second Polytechnic University, Shanghai 201209, China
Email: chenwei@ee.sspu.cn

Abstract—In order to achieve the integrated optimization in developing the big network system, this paper presented the concepts and principles of layered design method and collaborative development using of different tools firstly. Then we gave a recommended layered architecture for the distributed computing network system and also analyzed its pros and cons. At last, we extended the distributed computing network system to an information server cluster based on the task scheduling of dynamic and self-adaptive algorithm. The application of real-time information system designed by layered architecture shows that the method proposed in this paper is practical and efficient.

Index Terms—layered design, collaborative development, distributed computing, clustering technique, task scheduling

I. INTRODUCTION

The modularization and object-orientation are two good local optimization methods for software design. With the rapid development of network technology, the application systems based on network are bigger and bigger, and their functions are also stronger. So in such systems, the local optimization technique is not satisfied the need of integrated optimization. Here the “integrated optimization” means the various optimizations of source

code, design time, design fare, reliability, compatibility and maintainability and so on. In addition, because the various tools for software development have their advantages and disadvantages, it is out of time to develop a big software system only by one tool. And it is more important that the mega-merger of several strong tools for different platforms is used to develop a big software system collaboratively. So it is increasingly imminence that the software development architectures on higher level are called. It is a better choice to develop a big system using a layered design method [1-3].

In this paper, we propose a layered design and collaborative development method for a real-time information system based on distributed computing and clustering technique. This work serves to complement existing approaches [1-4] to the layered architectures of various information systems.

This paper is organized as follows. In section II, we give the ideology and principle of layered design. Section III presents the layered architectures of distributed computing network system and analyzes its advantages and disadvantages. Section IV proposes the layered architecture of information server cluster system and introduces its task scheduling method, and then analyzes its performance. Section V applies this method to a real-time information system, and shows that the proposed

This work was supported by the Key Disciplines of Shanghai Municipal Education Commission (No. J51801).
Corresponding author: Qiudong Sun.

design method brings about more effective. Section VI gives the conclusion of this paper.

II. THE IDEOLOGY AND PRINCIPLE OF LAYERED DESIGN

The Open Systems Interconnection (OSI) Model [3, 5] is an abstract description for layered communications and computer network protocol design. In its most basic form, it divides network architecture into seven layers (as shown in Fig. 1) which, from top to bottom, are the Application, Presentation, Session, Transport, Network, Data-Link, and Physical Layers.

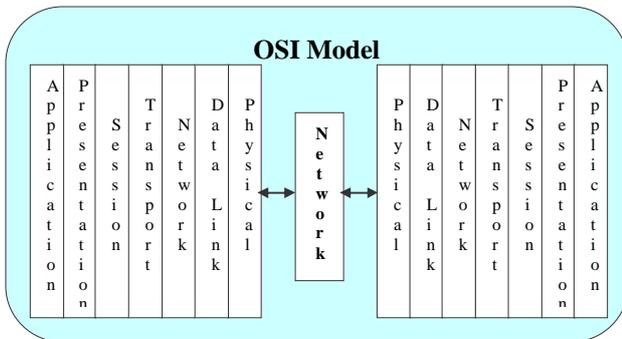


Figure 1. OSI reference model

OSI protocols are used to implement various networks. The OSI protocol stack is split into layers for modularity and orthogonal design. The layers form a hierarchy of functionality starting with the physical hardware components to the user interfaces at the software application level. Each layer receives information from the layer above, processes it and passes it down to the next layer. Not all layers are mandatory. It depends on the protocols that are implemented.

Although the OSI model is constructed for interconnection between different systems in a network and it includes hardware and software, but its ideology of layered design can be borrowed. In order to developing a cross-platform based software system collaboratively by several tools, the layered design method need obey the rules as follows:

Splitting Whole System into Functional Layers

According to its functions and tasks, whole system is split into some independent layers comparatively and encapsulated by modules and objects. Each layer may be an independent process, thread or others procedure module. Sometimes, the layer can be called layer component.

Each Layer Can be Designed Independently

Each layer in the system should be developed collaborative-ly by quite appropriate tools. In this way, the time efficiency of system development is the best. But it must be observed that we can establish the prescriptive communication relation between this layer and its adjacent upper layer and lower layer.

Due to every layer in the system is independent with any other layers, so the modification of any one does not impact others. Thus, the design method like this will

bring convenience for maintenance and upgrade of whole system and any improvement in a layer can also bring relevant benefit to whole system.

Every layer in system can be allowed to design independently, but it is not mandatory. Sometimes, the layers close-by may be designed all together. It depends on what the application style is.

Hold Together Between Different Layers by "Communication"

Here, the "communication" indicates any method to implement the information dispatch, transfer or exchange. The communication between layers is optional. For example, the applications in PC based on NT, the typical communication methods between different computers include Named Pipes, DCOM, Network DDE, RPC and WinSock. In the applications of a computer, the optional communication methods between processes include Anonymous Pipes, Shared Memory/File Mapping, DDE and DLL. But it needs regard that the chosen communication method can be implemented in all of two adjacent layers. Naturally, the general communication method can be realized by various development tools.

III. LAYERED ARCHITECTURES OF DISTRIBUTED COMPUTING NETWORK SYSTEMS

A. Layered Architectures

There is no strict uniform rule for designing the layer architecture of software system. Different application can have its own proper architecture. But according to the ideology of layered design and the characteristics of distributed computing, we can give a reference model for its typical application.

The leading architecture of distributed computing application is N-level one [6], which divides the application into three independent logical parts: Display component, Logic component and Data component. The reason why it is called N-level is that the number of application services (logic component) is unlimited. Here we can also construct its layered architecture as shown in Fig. 2.

In Fig. 2, the whole application system is composed of Display layer, Logic layer, Data layer and Network layer. The Network layer and the communication sub-layers in the other layers can be less considered, because they have been formed basically in network constructing. If the Data layer uses the ODBC supporting database, such as Oracle, Sybase, SQL-server, DB2 or Informix etc, then it is also less considered when programming. The key task in the system is the harmony and cooperation of the User layer in Client and the Application service layer in the Server.

Due to applying the distributed computing, the Client/Server needs no longer too big, too fast CPUs and too large capacity memory. User can optimize the application program to reduce the complexity of tasks to an acceptable degree by using of distributed object if need be. So, the Client can be real thin-client.

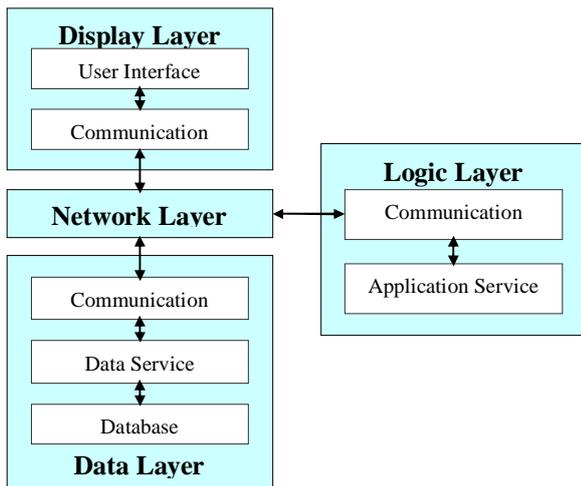


Figure 2. Layered reference architecture of Distributed Computing Network

B. Advantages of Layered Architecture

1) Clear Dividing Lines of Logic

The layered architecture enables the software system having clear logic dividing lines. The modularization and encapsulation are the final aim of any Object Oriented methods. It will be easy to achieve if there are clear logic dividing lines.

2) Optimizing the Development Resources

The functions of an application program can be divided into some layer components. Each layer can be implemented by a special developing team. Therefore, each team can take full advantage of its developer's talent for GUI designing, data modularization and clever transaction processing etc. In addition, the a posteriori viewpoint can usually gain a much more optimization codes.

3) Flexible Code Assembly

Each layer in the system is an independent component. So, the code assembly of whole system is very flexible. If the layer components are the independent processes or threads, the layered architecture will provide the building blocks method for them.

4) Easy Maintenance for Codes

The distinct independent layer component allows its inner modification to not impact other layers. Substantially, each layer is like a dark box to others. It provides a suit of interface functions for other layers. The others would not care about how its task is completed.

5) Support Cross-Platform Collaborative Developing

Some layer components in the system are allowed to be independent processes or threads. Therefore, we can choose proper development platforms and tools to implement them to achieve the best time efficiency.

6) Support Heterogeneous Data Sources

The request layer in the system submits a request order to the service layer, and then the service layer decides to

select which data source in the network to response the request. So, the data source is transparent for the request layer. It is foundation for the system to support heterogeneous data sources.

7) Enhancement Security

The accessing right can be designated to locate at any layers in the system. It will provide a multilevel security mechanism and improve the security level.

C. Disadvantages of Layered Architecture

1) Additional Spending of Communication

The data exchange amongst layers increases the additional communication time. So, the running efficiency will drop down. Sequentially, the running speed of whole system will also be slowed down. This problem can be solved by increasing the computer speed. Fortunately, it is not difficult to do so.

2) The Complexity and Expense of N-level Model

The N-level architecture for the distributed computing application will lead to increase the complexity and expense for system realization to a certain extent. But it enables the maintenance and upgrade easily for the future. So, it also will save the spending. It is worthy of being one-off invested in development phase.

IV. THE LAYERED ARCHITECTURE OF INFORMATION SERVER CLUSTER

D. The Definition of Information Server Cluster

The information server cluster (ISC) is such a system that several information servers work with parallel mode in the system, and each of them plays equal role to tidy and store the data from collectors and provide various services to query terminals, and also every one can be moved away or hung in the system at any moment, that is plug and play (PnP).

Although there are several information servers in the cluster, but as far as data collectors and query terminals are concerned, they should be an integer or a whole component.

E. The Guarantees of Software and Hardware

To want information server cluster operating as normal demands, it is far from enough that just having itself, and it should be supported by other hardware and software systems.

First of all, the implementation of parallel processing of information server cluster depends on the distributed computing. Its architecture is as shown in Fig. 3. There is an application server in the system, which can combine several independent information servers as a whole component and schedule their parallel works. Both the pending data from collectors and the query instructions from user terminals are dispatched to various servers in the cluster by a task scheduler based on the dynamic and self-adaptive algorithm in the application server.

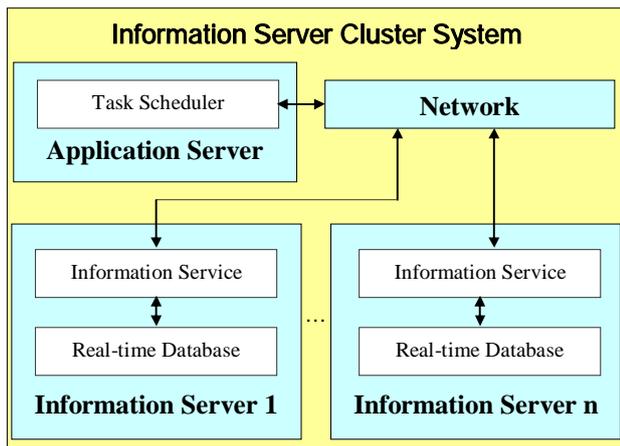


Figure 3. The architecture of information server cluster system

Secondly, the connection between information server cluster and application server are built by communication protocol such as WinSock in windows. Thus, once the one or several servers in cluster disconnect from or connect into the system, the application server will detect them immediately and assure dispatching stored data equably to each server and sending query instructions to all connected servers, i.e. realizing PnP.

F. The Dynamic and Self-adaptive Algorithm for Task Scheduling

The task scheduling is one of top challenges in parallel distributed computing. Its general form and several limited forms are all NP-complete problems. Accompanying with widely using of the large-scale parallel distributed processing system especially the network workstation cluster system, it becomes a key problem to improve the utilizing rate of resources how the efficient scheduling tactic can be designed to balance the loads in various node computers [7-11].

This paper uses a dynamic centralized self-adaptive scheduling tactic (as shown in Fig. 4) [7], which collects the load status from the common processes in various information servers by information collection processes (ICP) in the application server and sends them to a scheduling process in the same server to dispatch the tasks of data storing into the most appropriate information server.

The ICP does not participate in decision-making of task scheduling but is responsible for gathering load status and transferring the instructions from query terminals. When the number of node computers located in parallel platform are great than an enactment threshold, the application server will automatically create an ICP, which is responsible for gathering the load status of its own node computers and sending them to the scheduling process wholly to decrease the access times to application server by every node computer and lighten the system spending. When all of its node computers are removed, the ICP will be destroyed too.

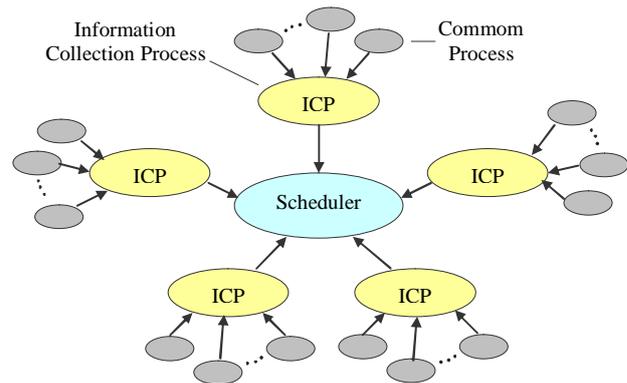


Figure 4. The architecture of load information collection process [7]

G. The Advantages and Disadvantages of ISC

There are quite a lot of advantages to the ISC, this method can provide information servers much flexibility for their maintenance, data backup and query of backup data.

1) Easy Maintenance

Since the ISC supports PnP, if there is an information server out of order and needed to repair, it can be taken off the system immediately and also can be put in this system when repaired.

2) Easy Backup

It is unnecessary to prepare backup hardware and software sedulously in such an ISC. There are two reasons supporting the point of view. First, the current process of data writing is, namely, backup. When the store medium (hard disk or re-write DVD) is full, you can take away from the system and change a new one put in. Thus, the data in the full medium has become the backup data. Second, the data in the real-time database usually has certain time validity, the data outdated can be circularly covered with new data. As enough as the capacity of store medium in the existent ISC, therefore, the data covering method can be applied to data writing process. Here, the backup process will be no longer needed.

3) Easy Query for Backup Data

The PnP function of ISC can assure the backup information server linking to the system optionally just like a general information server. This method can easily solve the query problem of backup data. But is must be considered that you may make out a distinctive mark for backup information server distinguishing from general information server to avoid the system covering its backup data.

4) Real-time Assuring

The parallel processing for multi-servers can assure system's real-time capability of huge data collection and information query. In principle, if the data capacity increases rapidly, the speed of data writing and inquiring can still be ensured only through adding some information servers.

5) Some Disadvantages

Although ISC has many advantages, its disadvantages are also obvious. On one hand, the cost of hardware of system may be raised due to increasing the number of servers. However, using the technology of distributed computing in the system can lower the requirement of single server' capability. From this point of view, the part of raising cost of hardware can be compensated. On the other hand, the capacity of communication in the system will be increased because of distributed computing. But this issue can be solved by improving the network speed.

V. THE APPLICATION IN REAL-TIME INFORMATION SYSTEM

Generally, a real-time information system is composed of real-time data collectors, real-time databases, data service programs, system database, data processing, information query, system management and network communication services. However, to have high flexibility in the design scheme, the system should be re-bedded according to its logical functions. As a result, the system can be segmented into network (NT), system database server (SDS), real-time information servers (RIS), whether using a single server or using multi-servers in parallel mode regarding as its scale), application servers (AS), clients (C) and data collectors (DC). And then, a real-time information system as shown in Fig. 5 is formed after connecting those virtual layers by network communication. In the Fig. 5, all virtual layers can be connected by virtual network and each of them can be divided into some sub-layers or modules in functional basis, which generally include communication sub-layer and functional sub-layer. In principle, all layers and sub-layers in the system can be designed independently, but they should obey the normative interface.

H. Network(NT)

Today's information system is usually a network system, so a network layer should be involved in the system. But considering the small-scale system or special applications, there is probably no network in it. Therefore, the network layer should be designed as a virtual one for practical applications making choices.

I. System Database Server(SDS)

The system database server SDS is actually a database system, which is used to answer for providing the services of management, maintenance and query to those relatively steady system data. It can be located in a real server alone or together with other service procedures in a computer in form of process. The database operations in the system can be implemented by instructions among SQL Server and SQL Clients [6].

J. Real-time Information Servers(RIS)

As the database servers, the real-time information servers are data stores for whole system, which are response for information processing and being written and red, and also providing various information query services. To get a rapid response, the RIS is normally not a current general database system such as Oracle or Sybase, it should be a special database created by

ourselves. The operations to this special database can be completed by instructions among WinSock Server and WinSock Clients [12].

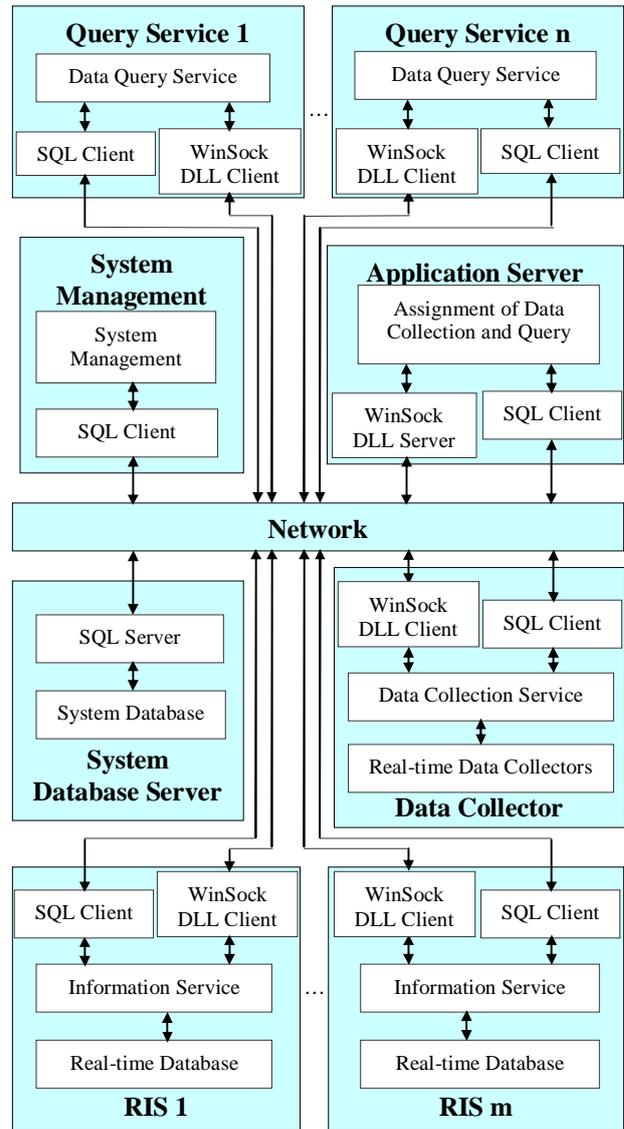


Figure 5. The layered architecture of real-time information system

K. Application Server(AS)

The application server is a service centre in the system, which is applied to solve those issues like one-to-many (such as one real-time data collector to multi real-time database servers) and many-to-one (such as multi clients to one real-time database server) in the communication services of database operations to reduce the pressures in clients, data collector and real-time data server. The all operations for data storing and information query are accomplished by AS. The instructions of query operations from clients and the written data from collectors are detached to appropriate real-time database servers by AS. And the query results from those real-time database servers are put together to return to the client.

L. Clients

The clients are mainly used to manage the system data and query the information and so on.

M. Data Collectors

The data collectors are response for collecting data and packing up, and then sending them to AS assigning to relative unoccupied real-time database servers.

The whole system is developed by three tools: PowerBuilder, Visual C++ and Assembly. PowerBuilder is applied to develop the data query in user-ends and the setup of various system tables, because its DataWindow Control has particular advantage during database application development [12]. Visual C++ is applied to develop the real-time data service program and the dynamic link library of communication based on WinSock protocol [13], because it has plentiful API functions, class libraries and various application development guides. Assembly is applied to develop the real-time data collection module, because it has advantage of reduced instruction and rapid execution speed. We take full advantage of the characteristics of three tools to develop the system collaboratively, so it is obvious that the time efficiency is improved.

VI. CONCLUSION

This paper described a layered design rules and methods for the distributed computing network system, recommended a layered architecture for it and analyzed its advantages and disadvantages. In order to construct a huge information system, the information server cluster based on task optimization scheduling had been introduced. The application instance of real-time information system shows that the layered design method has extremely agility and validity in aspect of developing network application system and the proposed architectures are efficient for using different tools to develop different layers collaboratively.

ACKNOWLEDGMENT

This work was supported by the Key Disciplines of Shanghai Municipal Education Commission (No. J51801)

REFERENCES

- [1] E. Stoyanov, A. MacWilliams and Dieter Roller, "Architecture for Distributed Component Management in Heterogeneous Software Environments," *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*, 2007, pp. 167-172.
- [2] Stefan Kirn, Otthein Herzog, Peter Lockemann and Otto Spaniol. *International Handbooks on Information Systems—Multiagent Engineering (Architectural Design)*. Springer Berlin Heidelberg, 2006, pp. 405-429.
- [3] Peter McBrien, "Design of distributed applications based on the OSI model," *Lecture Notes in Computer Science*, 1997, vol. 1250, pp. 361-373.
- [4] P. K. Rajesh and B. MageshBabu, "Web-based collaborative conceptual design: an XML approach," *The International Journal of Advanced Manufacturing Technology*, vol. 38, no. 5-6, pp. 433-440.
- [5] Jan Verschuren, René Govaerts and Joos Vandewalle, "ISO-OSI security architecture," *Lecture Notes in Computer Science*, 1993, vol. 741, pp. 179-192.
- [6] Simon Gallagher, Simon Herbert, "PowerBuilder 6.0 Unleashed," Indianapolis, IN : Sams Pub., 1996.
- [7] X. R. Wu, "A Dynamic and Self-adaptive Task Scheduling Mechanism," *Computer and Modernization*, 2007, no. 148, pp. 65-67.
- [8] X. H. Kong, W. B. Xu and J. Sun, "Research on Distributed Multiprocessor Scheduling Based on the Ant Colony Algorithm," *Computer Engineering & Science*, 2007, vol. 29, no. 3, pp. 63-65, 83.
- [9] A. Z. Liu, J. Z. Wang, H. L. Jia, S. Z. Wang and L. Y. Chen, "Genetic algorithm for mobile Agent tasks scheduling," *Computer Applications*, 2007, vol. 27, no. 11, pp. 2830-2833.
- [10] Y. Zhang and X. H. Zhang, "Research of multiprocessors scheduling policy based on improved ant colony algorithm," *Computer Engineering and Applications*, 2007, vol. 43, no. 35, pp. 74-76.
- [11] X. N. Tong, W. N. Shu and Z. M. Li, "Load balancing and task scheduling of heterogeneous multiprocessor system," *Optics and Precision Engineering*, 2007, vol. 15, no. 12, pp. 1969-1973.
- [12] X. D. Jiang, "Windows Sockets Network Programming," Beijing: Publish House of Tsinghua University, 1999.
- [13] Wayne Robertson, Edward Koup, "Windows 95 and NT Networking," McGraw-Hill Companies, 1997.
- [14] Q. D. Sun, Y. P. Qiu, W. X. Ma and Y. J. Gu, "Collaborative Development of Network Application System Based on Layered Conceptual Model," *Proceedings of 2nd International Workshop on Database Technology and Applications*, 2010, pp. 296-299.

Qiudong Sun was born in P. R. China in 1963, received B. S. degree in the field of electronic engineering in 1985 from Shanghai Jiao-tong University, Shanghai, P. R. China, earned M. S. degree in the field of communication and electronics in 1988 from Shanghai University of Science and Technology, Shanghai, P. R. China.

He was an Assistant Lecturer in the Department of Electronic and Electrical Engineering of Shanghai Second Polytechnic University (SSPU), China, from 1988 to 1990. From 1990 to 1999, he was a Lecturer in the Department of Electronic and Electrical Engineering of SSPU. He was a Manager of Strawberry Software (Taiwan) Shanghai Co from 1991 to 1994, and a Director of Software Developing Department of Shanghai EverNew Applied Technology Research Institute from 1995 to 2005. He has been an Associate Professor at the School of Electronic and Electrical Engineering of SSPU since 1999. His current research interests are computer engineering, electronic engineering, and image processing and pattern recognition.

Prof. Sun is now a Director of Department of Communication and Information of School of Electronic and Electrical Engineering of SSPU, a Director of Shanghai Images and Graphics Association, a Researcher of Academic Committee of Management Science Academy of China engaged by special arrangement, and an Expert of Science and Technology Committee of Pudong District of Shanghai. He won the Third Prize of Science and Technology Progress Awards of Shanghai in 1998, the Second Prize of Science and Technology Progress Awards of Shanghai in 2004, and the First Prize of Teaching Achievement Awards of Shanghai in 2005.

Wenxin Ma received his B. S. degree in Electronic Engineering from East China Normal University in 1984.

He has been a Senior Engineer at the School of Electronic and Electrical Engineering of SSPU since 1997. His current research interests are electronic engineering and computer science.

Mr. Ma is now a Director of Department of Electronic and Information Engineering of School of Electronic and Electrical Engineering of SSPU, and a member of Shanghai Information Association. He took part in a key scientific and technological project of China in 1990. He won the Third Prize of Teaching Achievement Awards of Shanghai in 2005.

Yongping Qiu graduated from Shanghai University, Shanghai, P. R. China, where he received the B. S. degree in Microelectronics and Automation in 1985.

Mr. Qiu is now a Lecturer in the School of Electronic and Electrical Engineering of SSPU. His current research areas include computer science and software engineering.

Wei Chen received her junior college diploma in Electronic Science Technology from East China Normal University in 1986, B. S. degree in Electronic and Information Engineering from Shanghai Second Polytechnic University in 2008.

Ms. Chen is now a Dean's Office Clerk in the School of Electronic and Electrical Engineering of SSPU. Her current research areas include computer science and software engineering.