Architecture Design of Shop Floor Information System --- Using WWW and 3-tier Architecture

Shean-Shyong Chiu , Yih-Ping Luh

Department of Mechanical Engineering
National Taiwan University, Taipei, Taiwan, ROC.
Email: hdd@ms8.hinet.net luhyp@ms19.hinet.net

Abstract

Shop floor information system (SFIS) stores the related activity information of the shop floor. This information is used by management in decision making. Due to lack of complete planning, it has resulted in numerous islands of automation. Traditional 2-tier systems did not scale up and it can be a system management nightmare when you deploy it beyond the departmental LAN. A shop floor information system, which is based on both World Wide Web technology and a 3-tiered architecture, is proposed herein to resolve the above shortcomings. Furthermore, other function modules on the shop floor also update the state information of the common database. In short, the function modules maintain the said information in the SFIS database (tier 3). Similarly, Tier 2 (web server) retrieves data from database and generates the web page dynamically and Tier 1 (web browser) displays the web page. Also, we discuss how to use the related technology to satisfy the real time requirement of SFIS.

Keywords 3-tier architecture, World Wide Web, Shop Floor Information System

1.Introduction

Information technology (IT) has promoted the continuing efforts of industries worldwide to create agile and productive overall manufacturing systems, processes and enterprises. Current IT has changed information retrieval and usage. Furthermore, it increases information flow speed and completes information sources. Also, via speed and flexibility, it improves industry's manufacturing processes. Hence, IT use has become the index of competitive enterprise. [1]

Traditionally, information systems were based on a 2-tier architecture. That is, due to their ease of development, it was very popular. Notably, 2-tier architecture has contributed to current information automation. However, eventually users required a more complex system. Scaling up the original system is difficult, particularly when there are numerous relationships between user interface program and the application logic. Thus, 3-tier architecture was developed to resolve this shortcoming as well as provide greater system flexibility [2,3]. The shortcoming also occurred with the shop floor information system (SFIS).

Although many such information systems exist, they are based primarily on the 2-tier architecture. Hence, due to scalability and maintenance difficulties, users have attempted to apply 3-tier or n-tier architecture to rebuild their information system.

Recently, IT has been improving continually. Two significant technologies are Internet and World Wide Web (WWW). Internet connects computers around the world and WWW provides easy access to Internet resources. The application of these advancements to existing information system and their potential resolutions to the existing problem is discussed herein.

WWW technology and 3-tier architecture was chosen to design the SFIS for the following reasons: Firstly, information systems must provide an open interface so that information is sharable as easily as possible. Restated, openness is a basic requirement. Secondly, the 3-tier architecture is used to reserve the scalability of the information system. Thirdly, to reduce maintenance costs, the web browser performs as the unified user interface. Characteristics of the shop floor system, WWW technology and 3-tier architecture (integration of web browser, web server and database) must be preserved simultaneously. The result represents a SFIS design, which is based on WWW technology as well as 3-tier architecture. This enables users to monitor shop floor activity readily from any location.

A shop floor information system, which uses the WWW technology and 3-tier architecture, is designed herein. New information technology is employed to solve existing shortcomings, and improve SFIS. The remainder of this paper is organized as follows: In Section 1, the shortcomings of existing SFIS are introduced and new information technologies which resolve related shortcomings are discussed. In Section 2, the characteristics of 3-tier architecture and WWW technology are described. Section 3 presents SFIS functions and requirements. That is, the information system is mapped to the 3-tier architecture. Section 4 describes the proposed system architecture and software modules. In Section 5, implementation and results are discussed. Special experience in SFIS design is discussed here. Finally, Section 6 is the Conclusion.

2. 3-Tier Architecture and World Wide Web

2.1 3-tier Architecture

Simplicity is the biggest factor driving the popularity of 2-tier system. 2-tier is great for creating application quickly using visual builder tools. Figure 1 describes 2-tier and 3-tier architecture, which differ significantly. In the former, tire 1 includes two parts, graphic user interface and application logic. In contrast, in the latter, tier 1 manages only a portion of the user interface, whereas tier 2 controls the application logic. That is, the application can be centrally managed on the server --- application logic are made visible to standard management tools. Basically, tier 1 does not involve any application logic. However, if the web browser is used as the user interface, then tier 1 does not require maintenance. Furthermore, Companies implementing 2-tier architecture frequently encounter performance problems when the software demands of the application exceed the capability, or when the number of database users has increased to a point where the application can no longer scale. Table 1 compares these two architectures. Notably, the 3-tier architecture was chosen due to the following reasons: it is less complicated, there is less client-end support required, superior Internet support, and it facilitates application reuse.

3-tier architecture is the new growth area for client/server environment because it meets the requirement of large-scale Internet and intranet client/server application. In theory, 3-tier systems are more scalable, robust, and flexible. In addition, they can integrate data from multiple sources. 3-tier applications are easier to manage and deploy on the network --- most of the code runs on the servers. Also, 3-tier applications minimize network interchanges by creating abstract levels of service. Instead of interacting with the database directly, the client calls application logic on the server. The application logic then accesses the database on the client's behalf. 3-tier substitutes a few server calls for many database queries and updates, so it performs much better than 2-tier [2].

With the Internet, servers can get requests from any of the world's millions of connected browsers. An information system that serves thousands of enterprise clients is demanding. Applications that work perfectly well in prototypes and small installations fall apart when put into large-scale production. Consequently, client/server architecture is moving to 3 tiers or n-tier to deal with the new reality. Also, new SFIS, which are based on 3-tier or n-tier architecture, is being designed.

2.2 World Wide Web

Internet connects the world's computers and WWW facilitates accessing this resource. Both Internet and WWW have affected the world significantly. This effect is also prevalent within SFIS for the following reasons: [4]

(a) It delivers information easily in any format, to/from a wide range of computing formats.

Many types of data formats exist in a manufacturing system. The ability to deliver and integrate information in various formats enriches its presentation. Meanwhile, data within SFIS might exist in numerous formats on many platforms. Integration of information resources has become an important issue. The WWW provides an excellent solution to this issue.

(b) It provides unified user interface.

In 2-tier system, there is one user interface per application, which makes it easier to build information systems. Typically, an information system is not equipped with remote desktops; therefore, it cannot control the software that they execute. These shortcomings can be resolved by using web browser as the unified user interface. Web browser replaces traditional graphic user interface. Notably, a web browser is almost available in most platforms. Therefore, employing the web browser as user interface can also reduce a client's management costs.

(c) WWW supplies information throughout the Internet, that is, it breaks physical boundaries.

Finally, integration of the WWW and 3-tier architecture is discussed. Herein, the web browser, which represents tier 1, acts as the graphical user interface. Notably, it controls data representation and input. Tier 2 includes application logic and web server. It retrieves related data from common database and generates the corresponding web pages as requested by tier 1. Tier 3 is the common database, which stores related shop floor activity information.

The majority of www applications occur in tier 2. The primary technologies used in this layer are portions of a dynamic web page. Dynamic web page can be defined via two different meanings. Firstly, the web page is dynamic and, secondly, the web content is generated dynamically. Furthermore, it can also be divided into two parts, client-side and server-side, based on the process location. The client-side indicates that the client executes related processes. Related technologies include Java Applet, Java scripts and Active X. The server-side indicates that the server executes related processes. The related technologies include CGI Microsoft ASP, Java Servlets, Java Server Pages, and PHP. This study focuses on the server-side technology. Using related technology, application logic and web server can be implemented in tier 2.

3. Shop Floor Information System

In general, a manufacturing system contains some control levels [5,6]. The primary activities in a manufacturing system consist of management and activating information. Usually, the management information, which is located in the first three levels, consists of numerous data, such as production data. The activating information, which is typically located in the last two levels, contains equipment control commands.

In a computer integrated manufacturing system, communication between various levels is difficult. From management to execution, the function requirement is different from each other. Response time requests also differ from each level. Hence, it is impossible to satisfy all requests of all levels simultaneously. Therefore, appropriate data, which satisfies the demand, must be selected. Varying real time request levels also require specific methods. Centralizing all information to a single point is not suitable. That is, function modules such as scheduler and dispatcher must be able to communicate with each other directly.

Control System for Integrated Manufacturing (COSIMA) developed a functional architecture for shop floor control systems. The operational subsystems of this architecture are the production activity control (PAC) and the factory coordination (FC). The five basic software modules that compose the PAC are the scheduler, dispatcher, monitor, mover and producer [7] (Fig. 2). Herein, the monitor module is discussed. The monitor has three primary tasks. First, it captures cell data from the shop floor (device-level), the producer and the monitor. Second, it analyses the cell data. Third, it provides decision support for the scheduler and the dispatcher.

Maglica proposed some recommendations for this architecture [8]. The monitor in the system is very centrally located with regards to handling information flow. Not only does it perform the data collection, it also interprets it and sends decisions based on the data to the other modules of the architecture. Figure 2 illustrates how the monitor can become an information bottleneck, for, when it ceases to function the entire system ceases to function. Figure 3 presents Maglica's proposed new system architecture [8]. The primary difference between the new architecture and the original one is that information flow between different modules does not need to initially pass the monitor, thus changing the one-way information flow to two-way communication. The architecture proposed herein, based on Maglica's model, also solves the bottleneck problem.

In addition to the modification, demands for superior functions accompany the rapid progress of information technologies. Thus, several new requirements are as follows:

(a) SFIS should support with the Internet to enable a user to retrieve related shop floor status or additional information when required.

(b) Using the web browser as the unified graphical user interface is preferred. That is, numerous proprietary user interfaces are not expected within this new information system.

(c) Determining all potential requirements is impossible. Also, requirements change with system use, therefore, the information system must be flexible enough to accommodate these variations or extensions.

(d) Information must be open and proprietary component should be limited. The *de facto* standard is preferred. For instance, the database can be accessed using the standard SQL syntax through ODBC or JDBC interface. That is, as proprietary API is not required to access the database, the system is more flexible.

Furthermore, SFIS has one particular requirement, which is real time. Each level has a unique response time request. The Standard HTTP model is extremely simple; the client establishes a connection to the remote server, and then issues a request. The server then processes the request, returns a response, and closes the connection. Due to the model, SFIS requires modification. Basically, SFIS for real time response request does not close the connection. Subsequent details are described in Section 5.

In short, a good SFIS should have the following characteristics: 1. Information flow must be as smooth as possible, thus preventing system bottleneck. 2. Information must be accessible from any location. 3. It must be flexible. 4. Also, it must meet shop floor application requirements.

4. System Architecture and Software Modules

According to the modified PAC architecture as well as the SFIS requirements that were discussed in the previous section, our system was designed in two parts. Part I is the monitor module, which controls requests from outside the shop floor. Primarily, it receives outside requests, performs data integration and replies to the client. Its data source is a common database. Part II contains the other modules such as scheduler, and internal dispatcher, but not the monitor module. In addition to the application logic of these modules, they also communicate with each other. The proposed system contains two communication methods. The first is direct communication with others, which control information exchanges within local modules. Its only requirement is to establish a communication protocol within related modules. The second method pertains to the common database. That is, if an outside source requests data, it is retrieved from the common database. Therefore, using ODBC or JDBC interfaces related modules exchange information through common database. These related modules must update the common database.

Part I was designed with WWW technology and 3-tier architecture and focuses on accessibility and integration of the shop floor information. This is completed via WWW, which renders information more accessible. Therefore, if a client wants to access the activity status on the shop floor, they require only a web browser. Restated, a proprietary user interface is not required. 3-tier architecture is flexible and scalable. Primarily, Part II concentrates on the SFIS. Part II satisfies communication requirements and updates data within the common database.

Figure 4 depicts the proposed system architecture. Web browser, which is used to monitor shop floor activity status and access data from the database, implements tier 1 as the unified user interface. Tier 2 consists of application logic and web server and connects user interface and shop information database. The application logic retrieves data from the database, integrates it and generates corresponding web contents, dynamically. The web server communicates with the web browser. Additional modules, such as scheduler, dispatcher and producer also exchange information and interface. Also, through ODBC or JDBC interface, they update the database. The system architecture contains the following characteristics:

(a) Shop floor information is employed to analyze the shop floor activity. However, if the data source is not unique, the analysis result cannot be verified. Therefore, to ensure consistency and accuracy, one single source unified database is employed.

(b) For data mining, the shop floor system is divided into modules: scheduler, dispatcher, producer, mover and monitor. Each module has its required proprietary function and contain proprietary data and common data source. These modules only have partial access to the database. The application logic in tier 2 can be designed to retrieve and integrate data in the database based on a user's specific requests.

(c) Within data integration, various modules share common data in the database. Through complete system planning and analysis, this common portion is suitable for all modules. For example, the types of stored data, data retrieval and scheduled updates were all considered.

(d) The shop floor information system must support multiple users. Each user has various data manipulation and representation requirements. For instance, the expectations of a manager differ from that of an operator. Therefore, the user's job role determines the preferred user interface.

In conclusion, via a web browser, tier 1 can access SFIS from anywhere in the world. Middle tier controls data retrieval and integration, generates the web content dynamically, and transfers it to corresponding browsers. The bottom tier is the common database, which stores all data required in SFIS. Each application module updates the database with the corresponding data entry and interchanges data.

5.System Implementation and Results

Apache was employed to construct the web server and MySQL was the relational database management system. PHP programming language was applied to implement the application logic of tier 2. Also, the function calls provided by PHP were employed to access MySQL database and generate web page. Java Applet was employed to accommodate the real time request of the shop floor environment. The appropriate programming language was chosen according to the various requirements of the other modules. However, rather than the proprietary API of MySQL, standard database interfaces including ODBC or JDBC were installed to access the database. Furthermore, if required, this database can be altered.

The application logic language of tier 2 was chosen based on the following reasons: (a) It must be able to implement personalized web page. Web content should be generated dynamically to fulfill a user's specific requirement. (b) It should support various database accesses. (c) After data is retrieved from the database, it must be integrated and generate a corresponding web page, dynamically. (d) It should support programming logic implementation. Hence, PHP, which was designed specifically for web application, was selected as the server programming language. Comprehensive function calls provided by PHP support web page design and database access. It also provides programming logic implementation, such as branches and loops. Based on PHP and related database tables, personal web pages were designed to satisfy each user's requirements. In addition, servers can interpret the PHP language, which, in turn, reduces the software and hardware requirements of a client's computer. In addition to PHP, other server-side languages, such as JSP or Java Servlets, can also be an alternative. Anyway, it should satisfy all the above requirements.

Furthermore, other SFIS modules are discussed. If the module existed previously, then transferring data from original database to the common database is the major task. This transfer centralized data into a common database, which stores all related shop floor data. Due to this centralization, data is more complete, which is the basis of data mining. However, if the module does not exist, an appropriate programming language, which is based on application module characteristics, should be applied. Java is recommended, due to its cross-platform feature. Furthermore, if the operation platform is to be modified, updated in the future, Java is the best choice. Also, shop floor application modules can exchange data directly or through the common database.

In addition, Java Applet was employed to combat the real time demand. That is, Java Applet can maintain the connection after the web page transmission is complete. Figure 5 illustrates the execution of a server program on the server-side, which remains connected to the client-side via Java Applet. Subsequently, when the status requires an update, the connection does not have to be reinitialized. The details are as follows:

(a) Web server closes the connection after the entire web page is transferred to a browser. Hence, a subsequent page transmission requires another connection initialization, which wastes additional network bandwidth. Moreover, if the expected response time decreases, the standard protocol will not achieve the requirement. The control data transmissions occupy excessive network resources. Thus, if a connection can be maintained, it does not require initialization for each use. The network resources can then be applied to actual data transmission. Connections are maintained to reduce the network traffic, which depend on the real time request. However, if this request is not required for some pages, the connection is terminated.

(b) Shop floor activity status changes dynamically, which indicates that web content should be generated dynamically to reflect the change. Another feature considered herein was how and when to update the web page, for which two methods were proposed: The first is to reload the web page. That is, a web browser sends a reload request to the web server at fixed time intervals. When the request is received, the web page can be regenerated and transferred to the browser. Notably, this method is the most popular one that is used in dynamic web page design. Selection of a time interval was difficult. That is, if too much time elapsed, the status update would not be provided in real time. Alternately, if not enough is chosen, the network resources are wasted on connection initialization. Application modules, which exist inside of the shop floor system, change the status. Thus, time intervals were selected based on the server, for which a program was installed in SFIS. Once the status is updated, the server program actively pushes the updated status to the corresponding user interface application.

Java Applet, which was employed herein, maintains the server program connection, thus facilitating economical status updates. Network resource is not required to initialize each process. Furthermore, due to the connection maintained by the Java Applet and server program, server program can actively push the status data to client side.

Finally, SFIS functions are:

(a) Shop floor activity is monitored in real time

That is, shop floor activity data is stored in the common database, which is updated by the related modules. Moreover, when users need to access the shop floor status, the web browser can be employed to forward a request to a web server. The application logic of tier 2 retrieves and integrates related data, then generates the web page. However, if users require real time, through the browser, Java Applet becomes the user interface corresponding to the internal server and fulfills the request.

(b) Message exchange

The system also performs task assignment and message exchange, which enables managers to assign tasks readily. In addition, it also allows users to communicate with each other directly.

(c) Security control

The basic password authorized mechanism was employed to manage user access control. Restated, each user must enter a valid username and password to access the system. Notably, user information is also the foundation of message exchange, task assignment and personal web page.

(d) Data query

Users can forward a query request to obtain the working status and historical data from the database.

(e) Personalized web page

Each user has a unique web page, which is designed, based on the role of the user.

(f) Independent application modules

Basically, each application module is independent. Common data is shared through the common database or exchanged directly.

6.Conclusion

A shop floor information system using WWW technology and 3-tier architecture was proposed herein. Via web browser, which serves as a universal user interface, all users (local and remote) can access shop floor status from anywhere, at anytime. Through the application logic and web server in tier 2, data is integrated and web content generated dynamically, which renders the information more valuable and accessible. 2-tier architecture is limited and inflexible and its application logic and user interface are bound together. However, via a separate application logic and user interface, the 3-tiered proposed system eliminates these shortcomings. In addition, we also discussed how to use the Java Applet to resolve the shortcoming of standard WWW implementation using in real time environment. The proposed architecture could be the information portal of the shop floor environment.

Reference

[1] Shaw, M.I., Seidmann, A., Whinston, A.B., "Information Technology for Automated Manufacturing Enterprises: Recent Developments and Current Research Issues," *International Journal of Flexible Manufacturing Systems*, Vol. 9, 1997, pp.115-120

[2] Robert Orfali, Dan Harkey, Jeri Edwards, "Client/Server Survival Guide", 3rd ed., 1999, John Wiley & Sons, Inc. ISBN 0-471-31615-6

[3] "3- and n- Tier Architectures", http://www.corba.ch/e/3tier.html.

[4] "Is the Web Important?", http://www.cio.com/WebMaster/sem2_start_import.html

[5] Rane, S.P. and Taneja, S.K.,1988 "A Distributed Architecture for Automated Manufacturing Systems," *International Journal of Advanced Manufacturing Technology*, Vol. 3, No. 5, pp. 81-89

[6] Huang, H.P. and Chang, P.C., 1992, "Identification, Modeling and Control of a Flexible Manufacturing Cell,"

International Journal of Production Research, Vol. 30, No. 11 pp.2515-2543

[7] Bauer, Browden, Browne, Duggan, & Lyons (1991): "Shop Floor Control Systems - From design to implementation", Chapman & Hall, USA.

[8] Robert Maglica, "Improving the PAC shop-floor control architecture to better support implementation", *Computer in Industry*, Vol. 33 , 1997, pp. 317-322.

Table 1. 2-tier v.s. 3-tier client/server

	2-tier	3-tier
system administration	complex	less complex
security	low	high
encapsulation of data	low	high
performance	poor	good
scalability	poor	excellent
application reuse	poor	excellent
internet support	poor	excellent
…..		

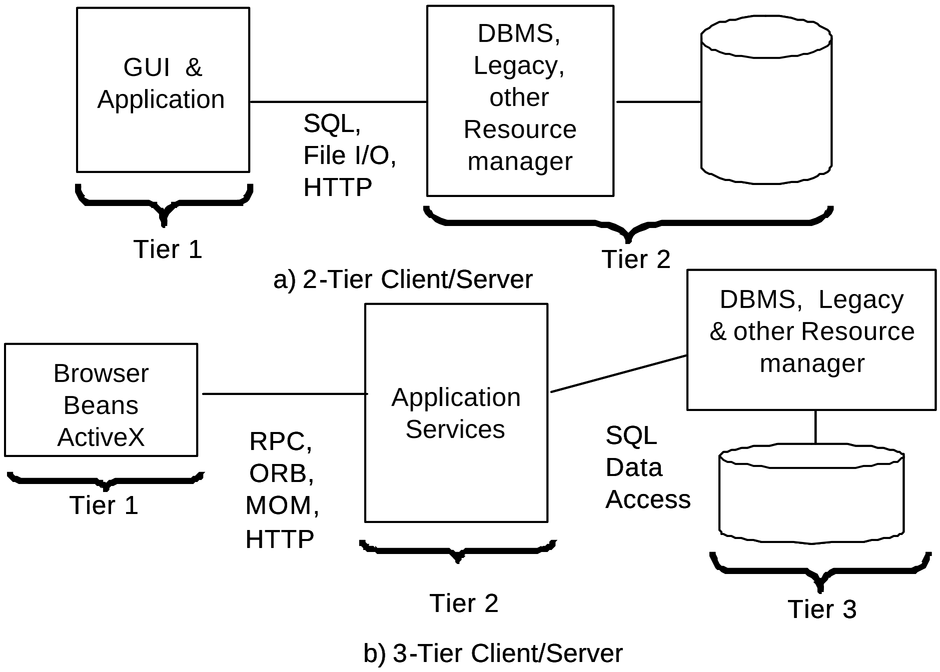

Fig. 1. Client/Server architecture ¡ Ð2-tier v.s. 3-tier

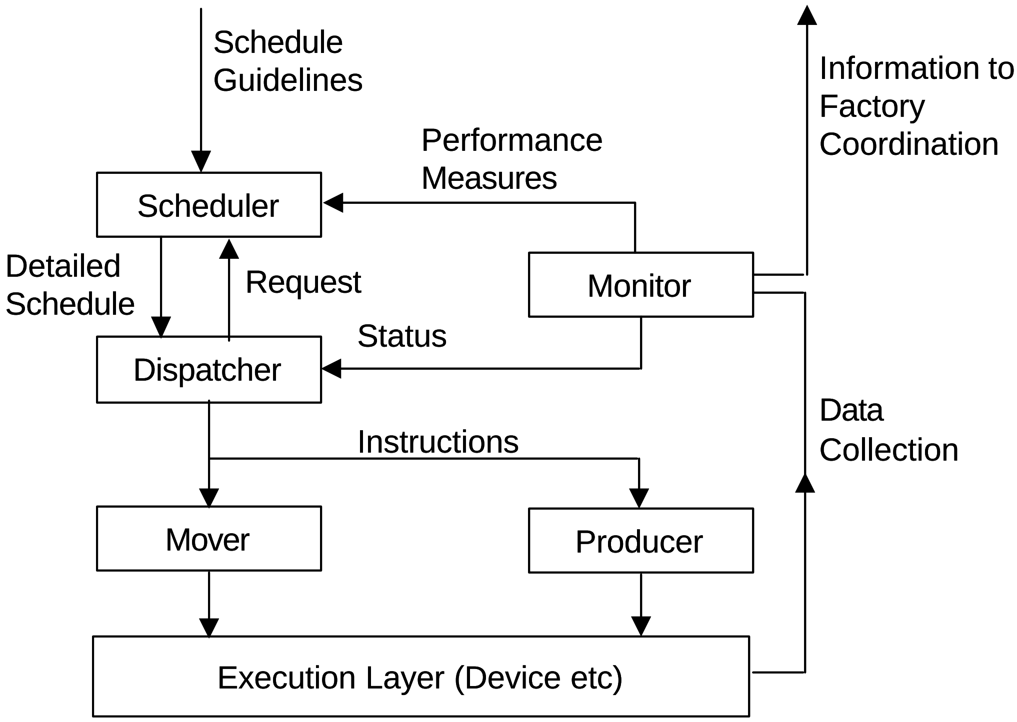

Fig 2. PAC architecture [5]

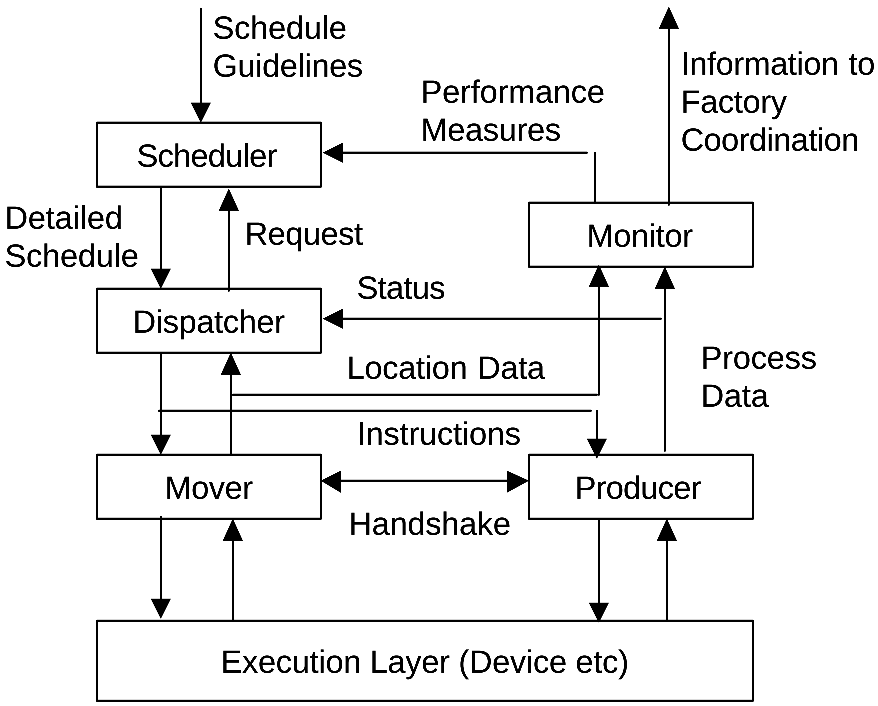

Fig. 3. Modified PAC architecture [5]

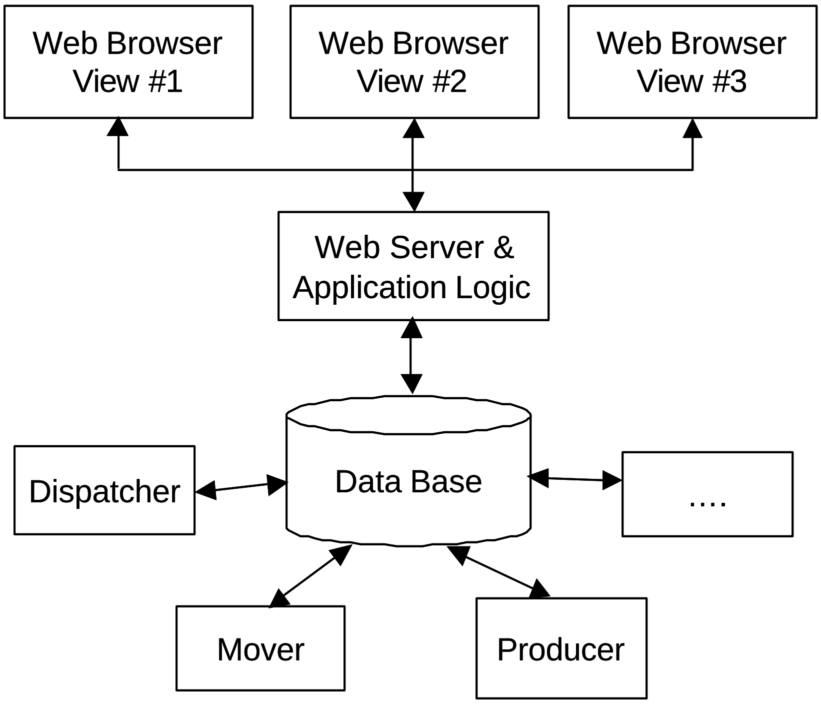

Fig. 4 ¡ BSFIS system architecture and software modules

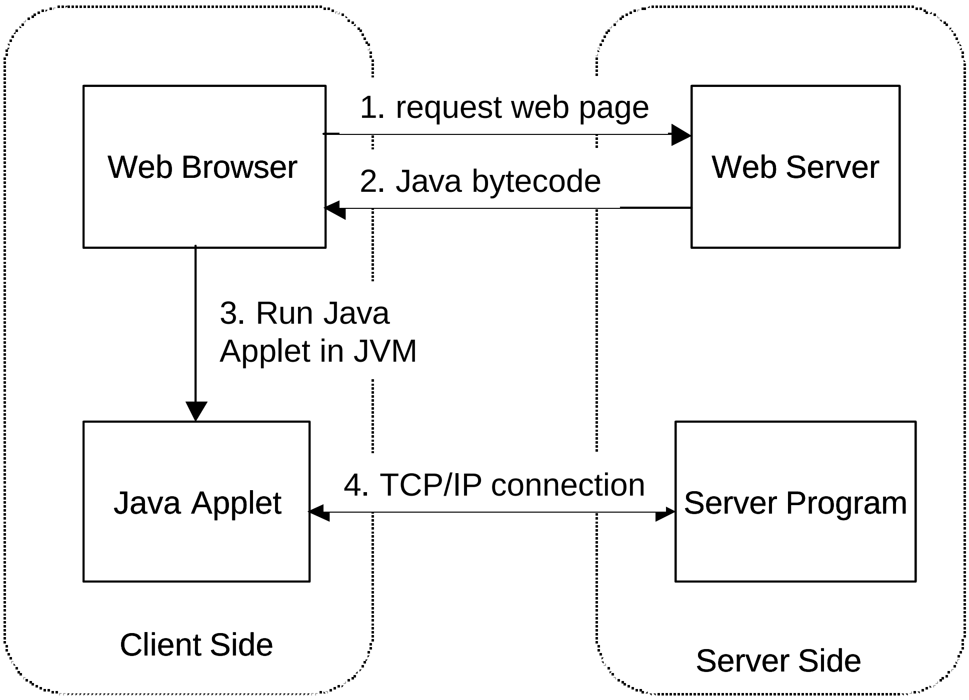

Fig5. Java Applet connection flow