Promotion of Development Efficiency by Using Frameworks

● Toshiya Hanamori ● Noboru Kurumai ● Takashi Shima

(Manuscript received April 3, 2006)

The framework is the core SDAS technology for reducing the time needed to develop a system and improve development efficiency and quality. Fujitsu provides the B².Sframework, which consists of three frameworks, as its application framework system. The first framework, Web Front Framework, can make development and maintenance more efficient because it separates screens, business logics, and data. It can also improve the security of Web applications. The second framework, Mission-critical System Framework, is used for mission-critical systems. Mission-critical System Framework ensures the efficient construction of mission-critical systems by providing a multi-language container and batch framework functions. The third framework, Client J Framework (CJF), uses Java to establish a rich client system. This paper introduces the latest facilities for using these frameworks and cites some examples of how these frameworks are applied.

1. Introduction

In open systems, there is no middleware such as Advanced Information Manager (AIM) [note 1)] of the mainframes used in mission-critical areas. Therefore, it is necessary to perform mission-critical control processing as well as business processing in business applications. To address this problem, the common components of control processing, simple control functions, and frameworks that consist of these items are created and the frameworks of vendors and open source software are used.

Fujitsu provides the B².Sframework as an application framework system. B².Sframework consists of the following three products:

1) Web Front Framework
2) Mission-critical System Framework
3) Client J Framework (CJF), which is a framework using Java for constructing a rich-client system

note 1) AIM is Fujitsu's integrated online database system for mainframes.

This paper introduces the latest facilities for using these frameworks and cites some examples of how these frameworks are applied.

2. Web Front Framework

Web Front Framework included in Interstage Application Server and Interstage Business Application Server is a framework function for developing Web applications based on J2EE.

A Web application utilizing Web Front Framework consists of three elements: data, screens, and business logics. Developers design screens and business logics after defining the data. The screens and business logics are separated from the data to minimize the effects of modifying one element on another. As a result, Web applications offering high expandability and maintainability can be developed. Moreover, Web Front Framework provides useful functions for building systems, for example, frameworks for EJB, and a function for extracting logs between the Web server and AP server.

Thus far, the development efficiency and maintainability of Web applications have been improved using Web Front Framework. On the other hand, there is growing interest in the vulnerabilities of Web application[1)] development. Because measures taken to address the vulnerabilities of Web applications increase the cost of system development, it is desirable to apply secure frameworks that have been verified in advance. This section describes the vulnerabilities of Web applications and then explains the security functions provided by Web Front Framework.

2.1 Vulnerabilities of Web applications

Web applications are exposed to non-validated input values, cross-site scripting (XSS), session management, cross-site request forgery (CSRF), and other vulnerabilities.

In particular, the failure to validate input values is considered the most serious vulnerability in the Open Web Application Security Project (OWASP).[2)] The following outlines the causes of these vulnerabilities.

1) Non-validated input values

This vulnerability causes an attack on system components when request parameters from a client are used without being validated. SQL injection is an attack method specific to this vulnerability. To prevent this vulnerability, all input values must be validated on the server and all illegal values eliminated.

2) XSS

Due to this vulnerability, malicious scripts are executed as a result of displaying data input to forms directly on Web browsers. To prevent this, unsafe characters must be replaced or removed when displayed on the screen as strings. This processing is called sanitizing.

3) Session management

For Web applications including login authentication, a Web site may be attacked if third parties know the login session ID. To prevent this problem, a new session ID should be assigned at login. Because unnecessary sessions left after logout may be misused, they must be quickly removed.

4) CSRF

CSRF is an attack by a third party who browses a fraudulent Web site and unintentionally sends requests to another site. As a result, the third party makes online shopping purchases without even realizing it. A method of preventing this problem is to validate screen transitions. Such attacks can be prevented by using an application that blocks requests sent from an unintended screen (Web site).

2.2 Web Front Framework security functions

As described in the previous section, the typical vulnerabilities of Web applications can be avoided by proper application design and development, coupled with validation by the system. However, this will increase development cost. Web Front Framework provides functions as a framework to deal with these vulnerabilities. The following describes the security functions provided by Web Front Framework.

1) Input value validation

Web Front Framework provides a function for validating input values on the server. To validate input values, validation rules are described in eXtensible Markup Language (XML) format and called through a dedicated Application Program Interface (API) from the business logic. These validation rules in XML format include input value conversion (int, long, string, etc.) and built-in validation logic and can check the correlation between input values. An example of validation logic is shown below.

- Check 2 values (big or small, equivalence)
- Character length check
- Comparison with enumerated values
- Check by regular expression

It is also possible to incorporate user-defined validation logics described in Java.

2) Sanitizing

Web Front Framework provides JSP extension tags called Unified JSP Interface (UJI) tags.

These tags can be roughly classified into those that control frameworks and those that are used as screen components. All screen components support sanitizing, and five characters (&, <, >, ", ') that are considered unsafe in HTML are displayed by escaping. As a result, even if scripts are output with screen components, they do not work.

Table 1 lists the UJI tags of screen components.

3) Session ID renew

As previously described, one of the methods for preventing third parties from knowing logged-in session IDs is assigning a new session ID at login. Normally, the following procedure is required to reassign session IDs:

- Save all objects associated with sessions.
- Discard sessions.
- Create new sessions.
- Reset all saved objects to new sessions.

Web Front Framework provides an API to perform the above procedure. By making use of this API, the object of a session scope uniquely defined for applications can be automatically inherited by a new session. A function that discards session objects at logout is also provided.

Table 1
UJI tags (screen components).

Basic tag	Character string display
	Resource
Component tags (19 types)	Character string input
	Integer input
	Date input
	Decimal, decimal point input
	Check box
	Radio button
	Push button
	Combo box
	List
	Anchor
	Label, etc.
Screen part tag	Table
	Tree
	List

With this function, unnecessary resources can be released after sessions are terminated, which reduces the risk of session misuse by third parties.

4) Request validity verification

In Web Front Framework, the framework manages the status of screen transitions to validate requests. When a request is sent from an illegal screen, an exception is thrown to the application, the screen can be switched to an error screen, and arbitrary processing such as logout processing can be executed.

This section described the security functions provided by Web Front Framework. By using Web Front Framework, the basic vulnerabilities of Web applications can be prevented so developers can focus on implementing measures against the vulnerabilities specific to applications.

3. Mission-critical System Framework

In mission-critical systems that were conventionally centered on mainframes, peripheral work and core work are now being decentralized and shifted to open systems as open technologies such as Internet technology advance. The flow of mission-critical work on open systems begins with online processing, resulting in the shifting of databases to open systems with greater coordination with decentralized systems and the shifting of online and batch processes associated with databases to open systems.

In such an environment, an Information System Department considers the following as being the most important issues for building and operating systems:

1) Improvement in system development efficiency/acceleration/stable operation
2) Flexible system expansion/modification
3) Total Cost of Ownership (TCO) reduction

Mission-critical System Framework is included in Interstage Business Application Server to 1) support the creation and operation of highly flexible and extensible systems with high produc-

tivity and robustness by providing common technologies needed to build mission-critical systems (online, batch jobs) utilizing open platforms, 2) share the application structures of mission-critical systems, 3) offer linking technologies, and 4) provide functions that support work management.

Figure 1 shows an overview of Mission-critical System Framework.

Mission-critical System Framework provides functions to create online work (synchronous and asynchronous) and batch work. Some typical functions are described below.

3.1 Framework for online work

1) Provision of common technologies needed to build mission-critical systems

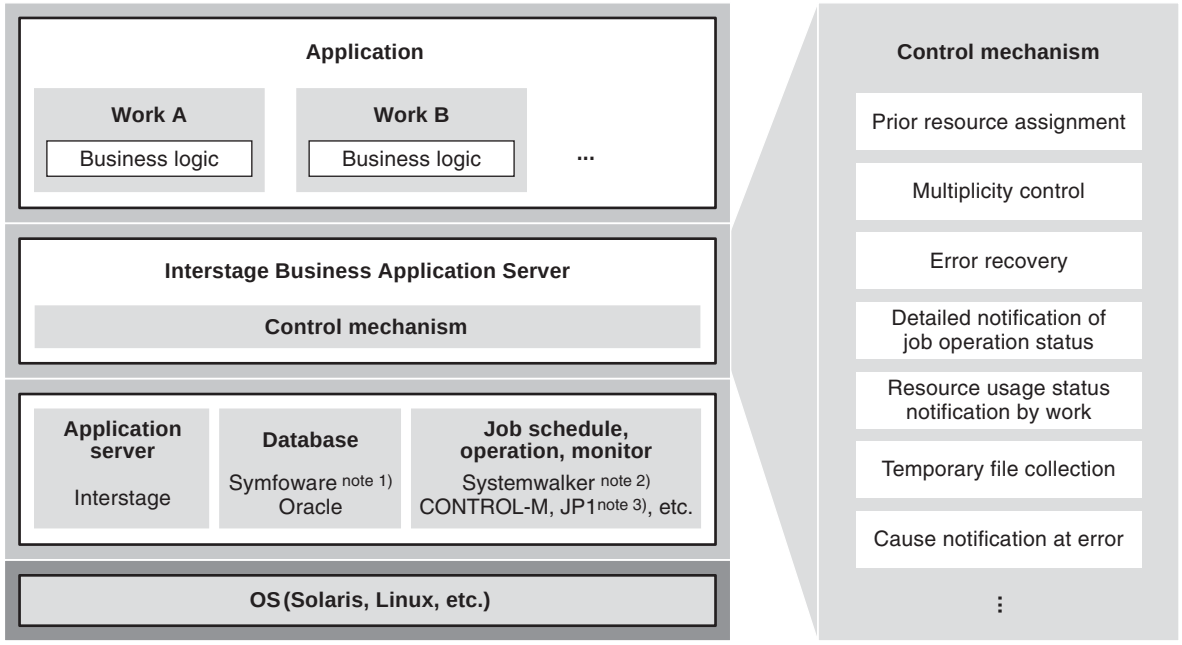
Applications can be roughly classified into control logics and business logics. Control logics are classified as application execution control in online processing, common control not dependent on processing form (e.g., transaction management and database access control), queue control of asynchronous processing and specific control for each processing form (e.g., advanced assignment of resources in batch processing), and a program for linking applications and managing resources. Business logics are programs that execute control specialized for work, for example, reflecting customer information and the contents of orders in databases.

However, high reliability and quality are demanded of control logic, thus requiring time and expenditure for design and development.

In Mission-critical System Framework, control logics and business logics are developed separately, and Mission-critical System Framework provides control logic as an execution environment to improve the productivity of applications and quickly create reliable systems.

- Execution control of applications

The following are provided: a function (called the preload function) that collectively loads all



note 1) Fujitsu's RDBMS
note 2) Fujitsu's operation management software
note 3) Hitachi's operation management software

Figure 1
Outline of Mission-critical System Framework.

business logics in an application when started and a function (called the demand load function) that loads a business logic when a processing request is made to that business logic and unloads it when processing ends. Systems can be created according to a usage pattern of applications, ensuring a quick response and effective use of memory resources with these functions.

- Simplification control of database processing

By using a database application function that makes database connections in advance and remakes them when they are broken, database access can be accelerated and processing simplified when connection errors occur.

Transactions, such as startup and termination, should not be considered in applications due to the transaction control function that commits and rolls back databases based on the processing results (termination code) of business logic.

- Consistent guarantee of queues and databases in asynchronous processing

When coordination is performed between applications by asynchronous processing, the sender writes messages to a queue and the recipient retrieves the messages from the queue. When an application accesses a database, consistency must be ensured when retrieving messages from a queue and accessing the database. If consistency is not assured, processing request messages may be deleted from the queues even if an error occurs when updating databases. Developing these processes entails both added cost and more time. As described earlier, Mission-critical System Framework provides processing in the form of control logic to access queues. The consistency of access to queues and databases is guaranteed using the transaction control function by providing a function that saves messages in queues to databases.

- Error processing in asynchronous processing

Although messages are normally retrieved from a queue and processed by applications, a message may remain in a queue due to various processing errors in the application and may affect subsequent work processing.

Mission-critical System Framework can automatically move error messages to another queue and localize problems by using the error message move function.

2) Techniques for sharing and coordinating application structures

Highly flexible and extensible systems must be created to respond to sudden market changes and demands on companies to meet their customers' needs.

Mission-critical System Framework defines the structures of business logics separately from control logics; improves versatility and reusability as work components that do not depend on platforms, middleware, or online forms (synchronous/asynchronous processing); and can develop business logics by employing the design methods used in different development languages (C language, COBOL, Java).

A function that converts character codes between client and server applications and converts data types for online synchronous processing is provided to develop client and server applications in different languages.

For asynchronous processing, a flow tool is provided to define the order in which business applications are called, define the messages to be transferred, and define the actions to be taken when errors occur. This enables new services to be easily created through a flexible combination of business logics.

3) Provision of the operation support function

In open systems, the functions of the Web server, AP server, and DB server are divided and decentralized on the servers to build systems. This brings the advantages of flexible support of increases in work volume and improved reliability. However, it also increases the maintenance costs for administrators due to the complexity of multiple servers and work management.

Mission-critical System Framework provides a logging function that outputs operating states, for example, the time from calling to the response

of applications, parameters, and error details, by automatically adding related user identification information for each server.

Output logs can easily be analyzed using various tools.

To track data for accounting and failure analysis, an API used to record the processing status (user log) of applications is also provided. This API is used to obtain user logs easily and properly.

These functions make it possible to visualize work, quickly detect performance bottlenecks, and reduce the costs of maintenance, such as for performance tuning and capacity planning.

3.2 Framework for batches

1) Provision of common technologies needed to build mission-critical systems

Assigning the resources used by jobs in advance and aggregating processing delay factors at the start of a job suppress the following events and stabilize turnaround time:

- Delay in waiting for resources to be released while jobs are being executed
- Error termination due to lack of resources at the latter stages of jobs
- Congestion of jobs due to deadlocks

Many jobs can be stably executed even on a single server by coordinating with resource management software such as Class-based Kernel Resource Management (CKRM) and supporting the following functions:

- Classification by job characteristics
- Multiplicity control by execution class
- Assignment of resources by execution class

2) Techniques for sharing and coordinating application structures

Because frameworks control file assignment and error collection processing, jobs can easily be created simply by defining the business applications and resources to be used, without having to create a control application.

Therefore, applying this batch framework improves development productivity fivefold to tenfold as compared with shells and definition steps.

In addition, programs coded in different languages (COBOL, C language, Java, etc.) can also be used along with existing assets and know-how.

3) Provision of the operation support function

Even when using operation management middleware, the operating states of processing steps in jobs and the usage states of resources could not be recognized. However, by applying this batch framework, the following information can be obtained in real time:

- Operating status for each processing step (elapsed time, etc.)
- File usage status for each processing step

When operation analysis and capacity planning of work were conducted, the resource usage status could be aggregated for each process and server. However, this could not be done for each work because the correspondence between work and processes had to be separately managed. Batch framework solves this problem by providing functions for the following:

- Obtaining the usage status log of resources for each job
- Aggregation of the resource usage status log for each work and a visual analysis tool (coordinated with Systemwalker[note 2])

If a job terminates abnormally, it takes time to identify the point where the termination occurred and the reason for the termination, as well as time to reexecute the job.

To solve this problem, the following functions are provided to report detailed processing results for each job and reduce the time needed to complete recovery:

- A job log that outputs the steps executed, the resources used, and the causes of abnormal job termination
- A function for restarting from the step where

note 2) Systemwalker is an integrated operation management software product based on the concept of Policy-based Systems Management (PSM).

an error occurred

If an application or shell terminates abnormally in conventional batch processing, unreclaimed resources must be periodically deleted in system maintenance to avoid trouble due to an accumulation of unreclaimed resources.

Batch framework provides a function that completely deletes unnecessary resources based on the defined necessity of files. Therefore, the load of system operations associated with abnormal termination can be reduced.

4. Client J Framework (CJF)

Recently, there has been increasing demand for introducing rich-client systems when implementing in-house, mission-critical business systems. The results of a customer questionnaire[3] conducted in Japan by Nomura Research Institute, Ltd. show that the percentage of rich-client systems will increase from about 13% in 2004 to 28% in two years (more than double). In-house mission-critical business systems account for 33% of the systems being migrated from the current fat-client systems to rich-client systems. Under these circumstances, many products and techniques associated with rich-client systems are emerging in the software product market.

Many users are considering implementation in Java. According to a survey covering about 1200 projects for system integrations handled by Fujitsu from the start of fiscal 2004 to the end of September 2005, users seeking Web systems accounted for 52% in 2005 (an 8% increase over the previous year). As shown in **Figure 2**, 27% of users (a 2% increase over the previous year) are considering Java (Applet or Java application) as a client technology.

There is growing demand for Java when building client systems, though its existing design method is somewhat immature. Especially, when building mission-critical business systems, the problem posed by the specialized knowledge and technology required to implement asynchronous control and assure performance must be addressed.

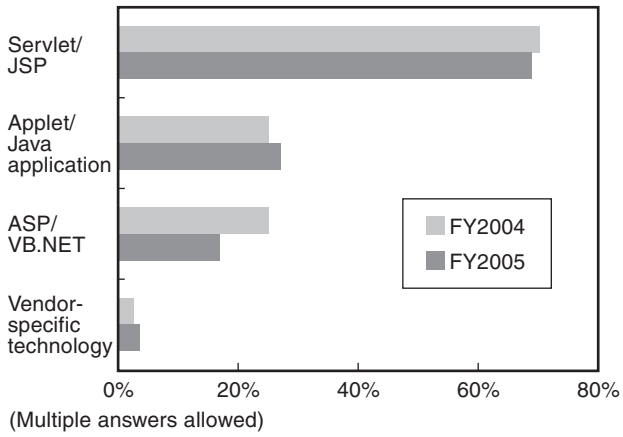


Figure 2
Candidate client technologies for Web systems in Japan.

To solve this problem, Fujitsu provides frameworks through CJF based on Java to offer rich-client solutions. By implementing rich-client systems that adapt CJF, both development scale and person-hours can be dramatically reduced.

An overview of CJF and examples of adaptation work are given below.

4.1 What is CJF?

CJF is a framework used to implement business applications for rich-client systems in Java.

The features of CJF are as follows:

1) Adoption of open technologies

CJF uses open technologies such as XML based on Java, which is a de facto standard in Web environments, without using proprietary scripting languages. This allows for a smooth appropriation and expansion of assets. Moreover, it makes it easy for development engineers to obtain the necessary skills and improve development productivity.

2) Adoption of the MVC model[note 3)]

Developers can concentrate on the development of screens and business logics by dividing development into business logic (Model), screen (View), and control logic (controller provided by the framework), thus facilitating high-quality system development. Screens and business logics can also be developed in parallel to shorten the development period.

3) Minimum necessary development

Desired operations can be implemented by simply minimizing the necessary development because some logics are defined externally. For example, screen transitions and screen layouts can be implemented just by using simple external definitions, and user confirmation using actual screens at an early stage of development is made possible.

4) Easy event management

In some cases, when a lengthy process (e.g., search or printing) is waiting for a response, the next process or a specific process is executed when the response arrives. In CJF, a framework is used to perform asynchronous event handling, so these situations can be easily managed.

4.2 Problems and solutions for conventional development techniques

Conventionally, screens and logics were developed together for each work screen. When employing this development method, processing must be implemented for each screen even if similar processing is executed. As a result, as the number of screens increases, the amount of development work also increases, thus requiring the modification of multiple logics. For parts that require advanced skill for screen transitions, screen layouts, event handling, and data processing other than screen and work processing, system quality tends to depend on the skills of developers.

Conversely, CJF employs a technique of developing processing for each event instead of for each work screen to improve development efficiency. As a mechanism for this purpose, screens and logics are separated, with logics separated into control logic, business logic, and definition information (**Figure 3**). The control logic provides functions for the following:

1) Storing necessary events from among events in a queue.
2) Recalling business logic corresponding to the first event in the queue by referencing definition information.

With this mechanism, if the same processing is executed on different screens, only one business logic needs to be described by the definition setting to recall the same logic for events, thus minimizing the effects on business logics even if a screen is added or modified. Moreover, because screens and business logics are separated, operations suitable for the skills of developers can be performed in parallel and the development period shortened.

4.3 CJF functions

CJF provides many functions as frameworks to support developers. GUI components provided by CJF are based on GUI components for the clients of Interstage Apworks (Apworks)[note 4)] (**Table 2**).

4.4 User-friendly development environment

The development environment has a significant effect on refining the skills of development engineers and on development productivity. CJF enables developers to use Apworks based on

note 3) Three programming design patterns of M (Model: part that describes business logic), V (View: part that describes screen design), and C (Controller: part that describes coordination control of work processing) comprise a method of creating programs corresponding to each role by dividing them.

note 4) Fujitsu component-oriented Java integrated development environment that enables efficient system construction.

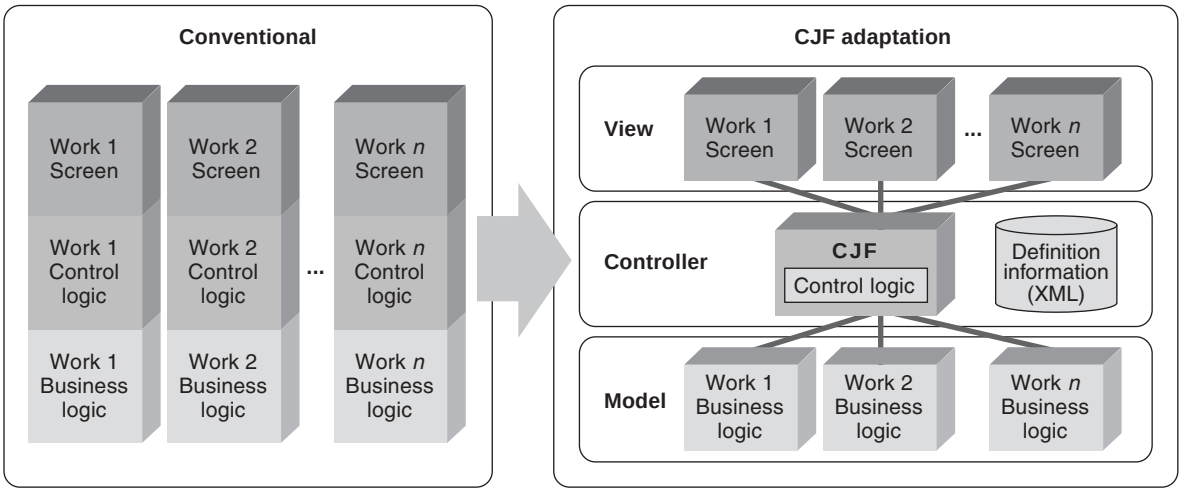


Figure 3
Conventional development method and CJF adaptation development method.

Table 2
Outline of CJF functions.

Function	Description
System control	• Performs initialization and termination processing to initiate start/end procedures and business logic. • Handles unexpected exceptions. • Can speed up initial start and switch operating environments according to operations by dynamically reading definition information.
Event control	• Collectively controls events received from screens, devices, etc. • Associates various events with business logic based on definition information. • Automatically registers event listeners associated with screen components.
Screen control	• Determines layout configurations and screen transitions according to definition information. • Automatically inherits data between screens. • Can display multiple screens and execute independent works (multiple windows).
Data area management	• Obtains and releases I/O data areas.
Device control	• Hides individual interfaces for server communication control and peripheral device control and provides consistent interfaces.
Screen component control	• Hides interfaces specific to screen components and provides consistent interfaces.
Message control	• Manages the content of messages independently of business logic for easy content management. • Can automatically display and clear message popup screens.
Utilities	• Provides a utility function, e.g., log output and screen image output.

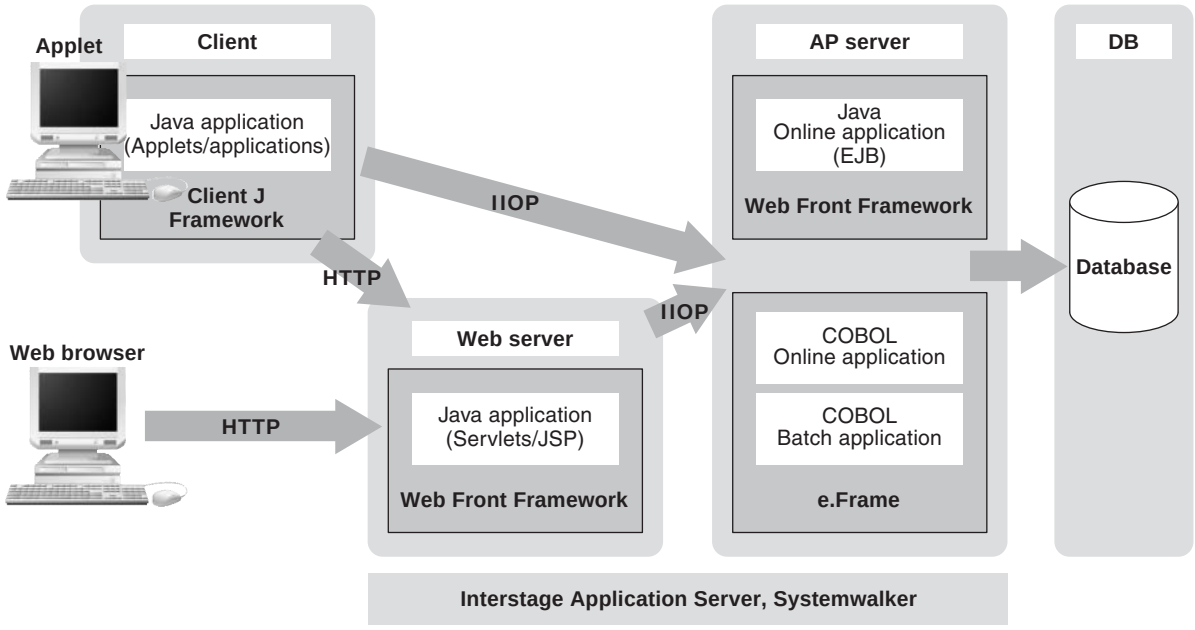


Figure 4
Standard system configuration when adapting CJF.

Eclipse,[note 5)] which is a de facto standard Java integrated development environment and the preferred environment of developers.

Screens can be developed using the graphical editor of Apworks while confirming actual images. Because the frameworks provide control logics, only business logics need to be described and developed. Definition information such as screen transition definitions and action mapping definitions that define the correspondence between events and business logics can be easily defined using a general-purpose tool (e.g., Excel) to set and convert these definitions into XML files.

4.5 Standard system configuration

Because CJF runs on the client side, it can coordinate directly with Application (AP) servers through Internet Inter-ORB Protocol (IIOP) communication and also coordinate with Web servers through HTTP communication. The assets on the server side can be used even if clients are migrated to CJF, thus minimizing the changes associated with system migration.

An integrated adapter interface is provided for parts that coordinate with servers.

Figure 4 shows a standard system configuration when adapting CJF.

4.6 Examples of adaptation work

CJF is adapted extensively to the development of entry type mission-critical systems, regardless of business category. It is also used in the solution business that targets rich-client systems. **Figure 5** shows the main needs in terms of current system form and some examples of adaptation in CJF. CJF is often adapted for in-house mission-critical business systems that require speedy operations in entry type work.

1) Example of adaptation in the financial sector

In the financial sector, CJF is adapted to

note 5) One of the open source integrated software development environments (IDE). It is expected to become widely used by Java developers and a standard common platform for software development.

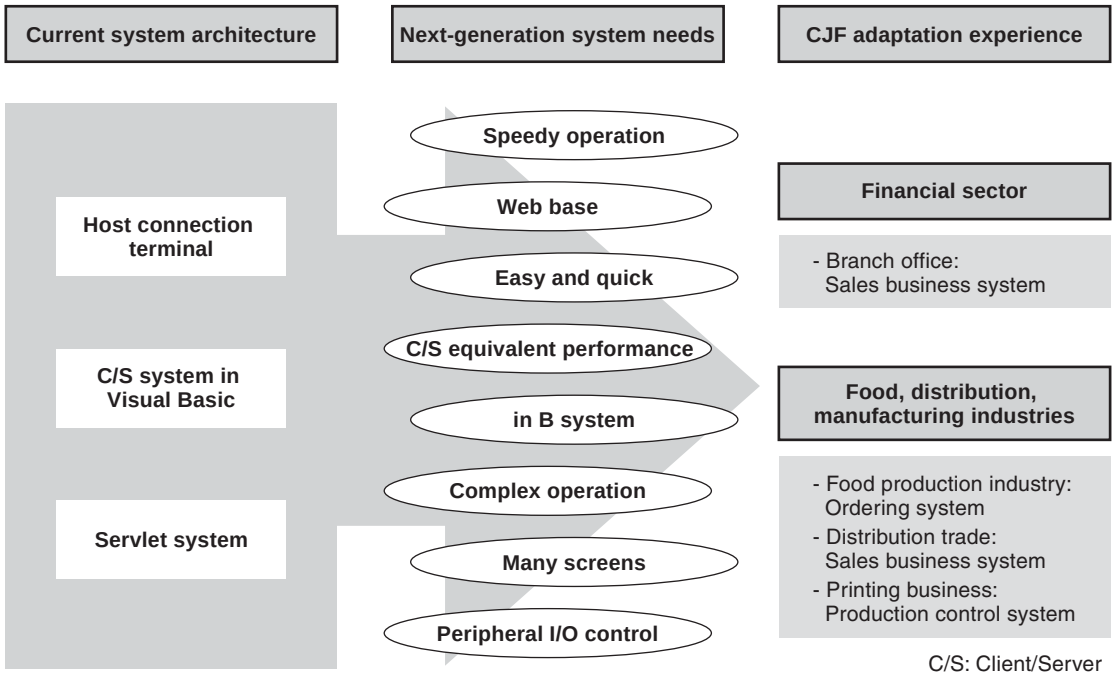


Figure 5
Architecture of current system and adaptation experience.

work that requires a high-performance GUI and various screen transitions (**Figure 6**) in the sales business systems of branch offices.

As a result of analyzing the productivity data of certain financial systems, the development person-hours were reduced by 38% when CJF was applied (**Table 3**). Data on actual developments was adopted as the productivity data when applying CJF. Calculations of productivity when CJF was not applied were based on the results of similar development performance in the past. The effects of application are due to the fact that developers were released from complicated event processing and could concentrate on implementing business logics because CJF provides complex, high-performance control logics.

The subject of this comparison is development of a basis that can be commonly used in the category of business. Although both system and structural design are more difficult than general business logics design, a reduction was achieved by applying CJF. A further reduction is expected in the development of general business logics.

2) Example of adaptation in manufacturing and food production industries

In the manufacturing and food production industries, CJF is adapted to work in the ordering, production management, and sales business systems that requires complex screen compositions (**Figure 7**), frequent use of PF keys, and control of peripheral equipment.

Currently, there are needs for rich-client systems in the distribution trade and in government and other public offices, for which the application of CJF is now being studied.

3) Example of adaptation in solution businesses

CJF is increasingly being used in various solution businesses. The TransMigration Services described elsewhere in this special issue provides a function that automatically generates source files written in Java from screen definitions, for example, the Presentation Service Access Method (PSAM), when host terminal screens are migrated to Web screens. **Figure 8** shows an

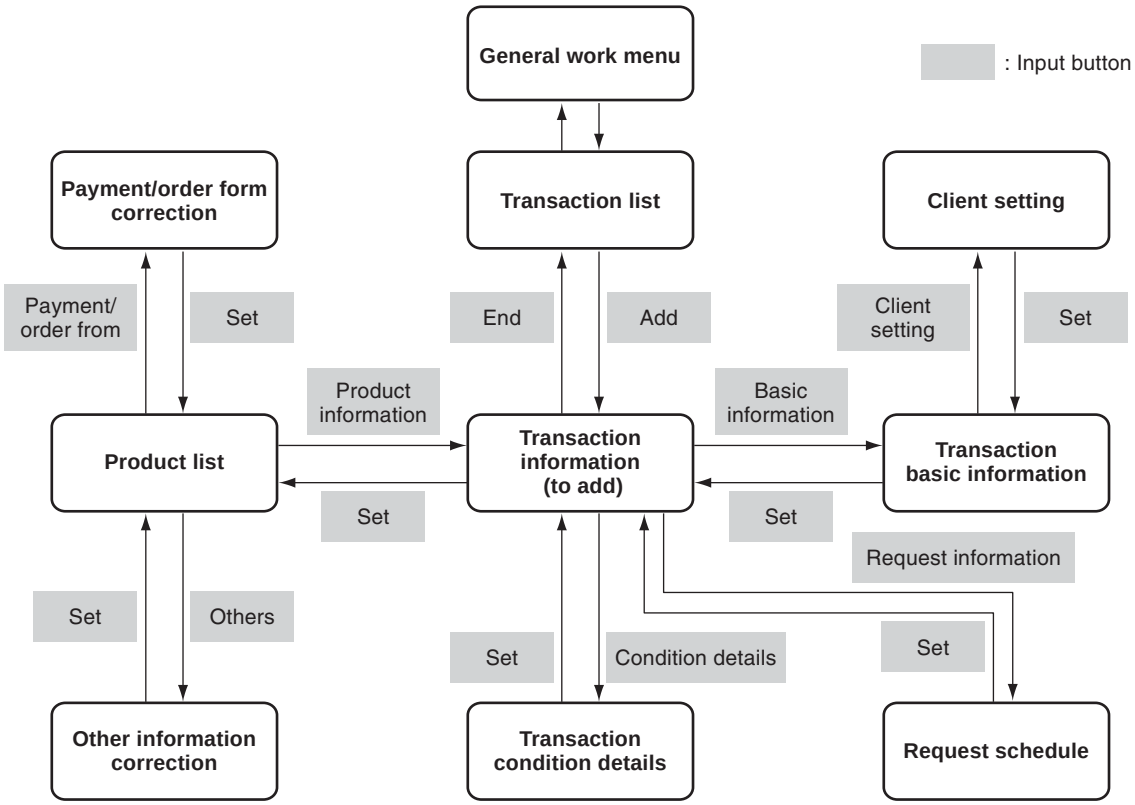


Figure 6
Example of screen transition.

Table 3
Effects of adaptation.

Comparison item	CJF not adapted	CJF adapted	Reduction effect
Development scale	70.0 k lines	54.1 k lines	-23%
Development person-power	128 person-hours/ month	79 person-hours/ month	-38%
Development period	18 months	15 months	-17%
Total number of bugs	960	693	-28%

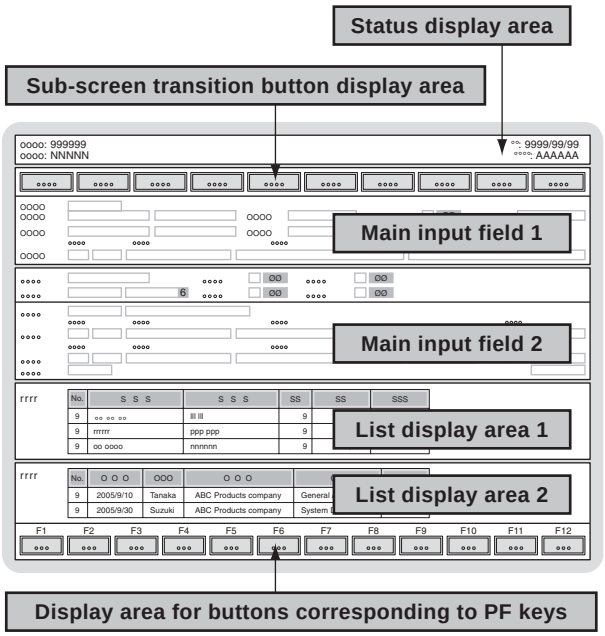


Figure 7
Example of screen composition.

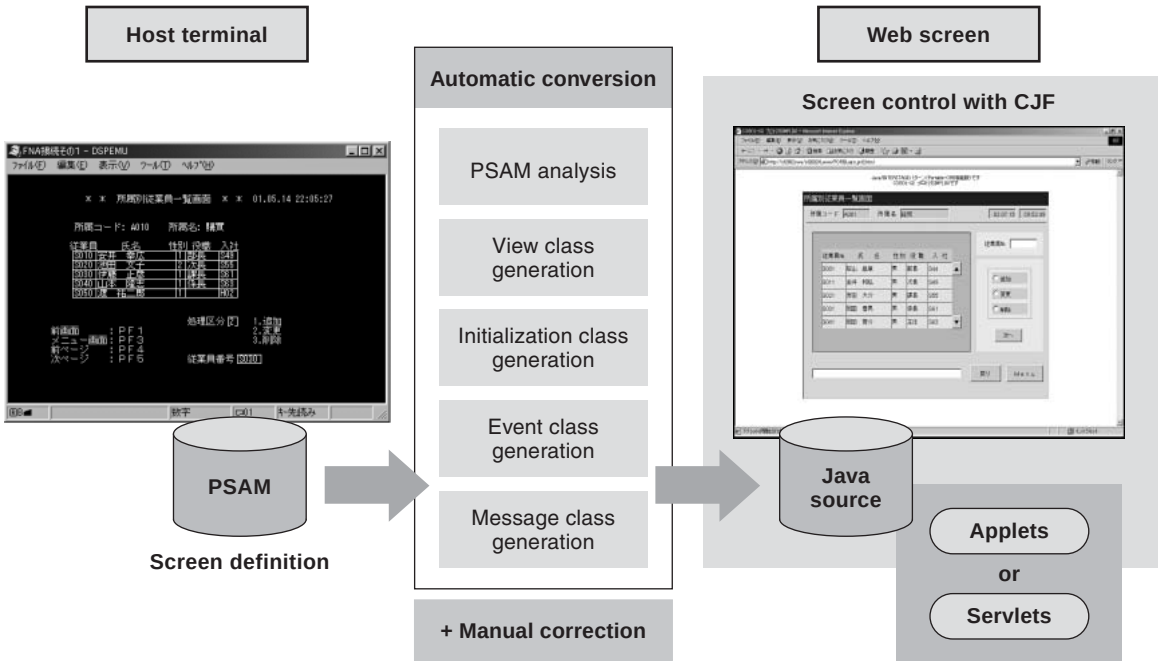


Figure 8
Example of adapting CJF in TransMigration Services.

example of applying CJF to the control layer of a Web screen. In image entry package products, CJF is adapted to realize a variety of image display controls.

5. Conclusion

This paper described Web Front Framework, Mission-critical System Framework, and CJF, which are essential application frameworks for implementing Web-based systems.

Learning time is required for the initial introduction of development by applying frameworks. However, because frameworks are used repeatedly, know-how about improving productivity is obtained through learning and maintaining parts, and this know-how dramatically improves development productivity. Frameworks represent a technology that developers are strongly recommended to use if they have not already done so.

Fujitsu uses these framework products in its system construction services, feeds back related results as required, and deals with the vulnerabilities of the latest technologies and Web applications. Fujitsu also plans to refine the frameworks that can be used as a base of system development offering higher quality and productivity.

References

1) H. Takagi: Security Vulnerabilities in Web Applications. (in Japanese), *Information Processing*, **46**, 6, p.636-642 (2005).
2) OWASP: Top Ten Most Critical Web Application Security Vulnerabilities. *http://www.owasp.org/documentation/topten.html*
3) Nomura Research Institute: NEWS RELEASE. Clients of next-generation corporate systems moving from stage of examination to that of introduction. (in Japanese), 2004/05/27. *http://www.nri.co.jp/news/2004/040527.html*

Toshiya Hanamori, *Fujitsu Ltd.*
Mr. Hanamori received the B.E. degree in Information and Knowledge Engineering from Shizuoka University, Shizuoka, Japan in 1993. Later that year, he joined Fujitsu Ltd., Numazu, Japan where he has since been engaged in research and development of software for mission-critical frameworks.

Noboru Kurumai, *Fujitsu Ltd.*
Mr. Kurumai received the B.E. and M.E. degrees in Information and Knowledge Engineering from Tottori University, Tottori, Japan in 1999 and 2001, respectively. Later that year, he joined Fujitsu Ltd., Yokohama, Japan, where he has since been engaged in research and development of software for Web application frameworks.

Takashi Shima, *Fujitsu Ltd.*
Mr. Shima graduated from the Mechanical Engineering Department of Nagaoka National College of Technology, Niigata, Japan in 1981. Later that year, he joined Fujitsu Ltd., Tokyo, Japan, where he worked on the development of financial terminal systems. Later, he has been engaged in research and development of Java rich-client frameworks.