

# System for Distributed Project Management over the Internet: PI-CEE

●Takao Okubo   ●Takahide Matsutsuka   ●Hirotaka Hara  
(Manuscript received June 21, 2000)

**Rapid information sharing using Internet technologies such as the WWW and e-mail is enabling software development to be performed collaboratively at widespread locations. However, this requires efficient project management at these locations. In this paper, we describe a new system which supports distributed project management on the Internet. This new system, the Project Information-Cooperative Enterprise Environment (PI-CEE) is based on the Server-Integrated architecture, which is based on a business model for constructing efficient enterprise systems. We implemented the PI-CEE system between distributed locations via the Internet using Java. The system enables project information sharing using a business-customizable notification function, and replication, which ensure efficient information sharing between distributed locations.**

## 1. Introduction

The structure of software development is changing as Internet technology progresses. Rapid information sharing by e-mail, by FTP, and over the World Wide Web (WWW) has enabled collaborative development at widely separated locations. However, this requires efficient project management at these locations. To meet this need, we have developed a project management system, the Project Information-Cooperative Enterprise Environment (PI-CEE) that enables software developers in multiple locations to share information about projects, tasks, documents, etc., via the Internet.

The PI-CEE system is based on an advanced concept we have developed for efficient distributed enterprise systems called the CEE.<sup>1)</sup> The main components of the CEE are a model and an architecture. The CEE provides a business model that describes business objects and relationships such as processes, artifacts, persons, and organization units. The mainstream approach of current dis-

tributed system architectures is the three-tier architecture, which separates the application logic from the graphical user interface (GUI). However, it cannot adapt quickly to changes in business and infrastructure technology. The CEE architecture is based on this business model and integrates middleware and servers to protect them against the effects of these changes. In addition to the CEE, we have implemented two functions on the PI-CEE for smooth project management: notification and replication.

- Notification  
Information such as a delay in work should be quickly reported to the person in charge. The contents and destinations of such reports depend on the structure and application of the business concerned. We therefore developed a notification architecture using customizable rules and an event-driven execution system and then implemented it in the PI-CEE.

- Replication  
 Because of the ongoing decentralization and globalization of work, more and more software development projects are including software developers working in other countries. This raises the problem that when data is kept in a central location, the server-access response may be slow, causing a reduction in efficiency. In this case, data replication, in which copies of the data are placed on local servers at each location, becomes necessary. However, with the existing replication technology, it is difficult to maintain data consistency when two or more persons update the same data at around the same time. To solve this problem, we have developed a new replication method that maintains data consistency by detecting and resolving update conflicts.

In the next chapter, we discuss the CEE business model, the Server-Integrated architecture (which is based on the CEE business model), the need for an event-driven notification system, and the multiple server replication method used in the PI-CEE.

## 2. Concept

### 2.1 CEE business model

We use the business model concept to abstract business data and business functions. This abstraction increases the modularity and reusability of business systems. The importance of the business model in the construction of an enterprise system has been pointed out by J. Martin.<sup>2)</sup> The business model of the CEE is based on J. Martin's model but is more detailed. The CEE model defines the data flow between business processes explicitly, which is important for the construction and improvement of a business system. It also encapsulates the data and business of a business object using an object-oriented methodology. The standard business model of the CEE is shown in **Figure 1**. It conforms to the Unified Modeling Language (UML). In the CEE business

model, documents, for example, are related to processes, organization units, and persons. The input and output relationships can also be defined between the documents and processes in the same way they are defined with IDEF0,<sup>3)</sup> etc.

### 2.2 Server-integrated architecture based on CEE business model

The three-tier architecture is composed of a presentation layer corresponding to the GUI, a function layer which processes applications, and a data layer which accesses a server and databases. However, a change in a database or the GUI could affect the whole system because of the application design and development of the function layer. This hinders the efficient development and improvement of business systems.

We therefore propose a new architecture called the Server-Integrated architecture (**Figure 2**) which solves this problem. It has four layers.

- 1) Presentation layer
- 2) Service layer
- 3) Business model layer
- 4) Data integration layer

The business model layer is the core part of this architecture. It implements the business model defined in Section 2.1. Changes in the GUI, a database, or the middleware have no affect on the other layers because the business model layer abstracts the data and functions for business systems.

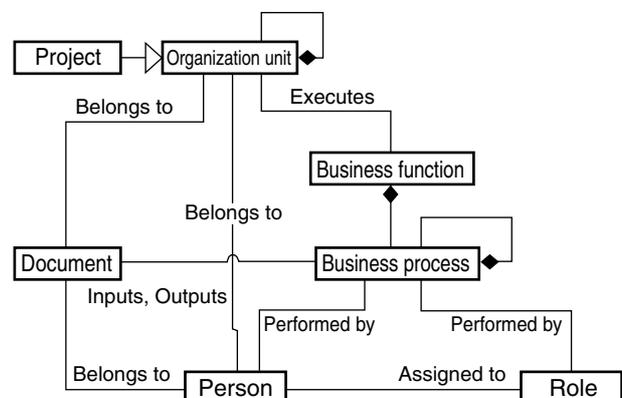


Figure 1 Standard business model of CEE.

### 2.3 Event-driven notification system

In project management, it is necessary to report changes in work information. For example, when a task is finished, the persons in charge of the following tasks should be notified as soon as possible so that they can quickly start them. If a task is delayed, it might be necessary to notify the project manager and the persons responsible for the following tasks. The contents and destinations of such notifications depend on the structure and application of the business, so it is appropriate to define rules for every application case. We therefore need an event-driven notification system in which the user can easily customize the notification rules.

### 2.4 Multiple server replication

Because of the diffusion of the WWW, it has become easier to concentrate and manage information on a single server. However, it is difficult to access a server over narrow-bandwidth network lines or from behind a firewall when the project

includes cooperating companies or offices in other countries.

Distributing servers and data to every location enables users to access a nearby server to get information, and the key technology for achieving information sharing in a distributed system is data replication. We have implemented a replication function based on the CEE business model and the Server-Integrated architecture.<sup>4)</sup> The architecture of our replication system is shown in **Figure 3**. If a user accesses a server and updates some of the data for a model object, the server stores the update in the database and then sends copies of the updated object to other servers. The other servers receive the object replicas and store the update in their databases. This method enables information sharing among distributed locations.

The multiple server replication function used in the PI-CEE also provides a method for detection and resolution of data conflicts and provides various lock services which are superior to existing replication methods.

Although this method is independent of the communication protocol used, we implemented the replication using SMTP for projects that include offices in other countries. The merit of SMTP is

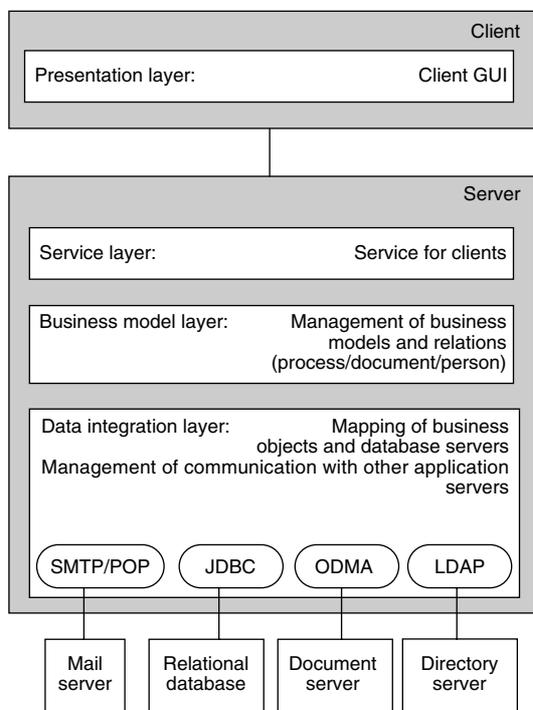


Figure 2 Server-Integrated architecture.

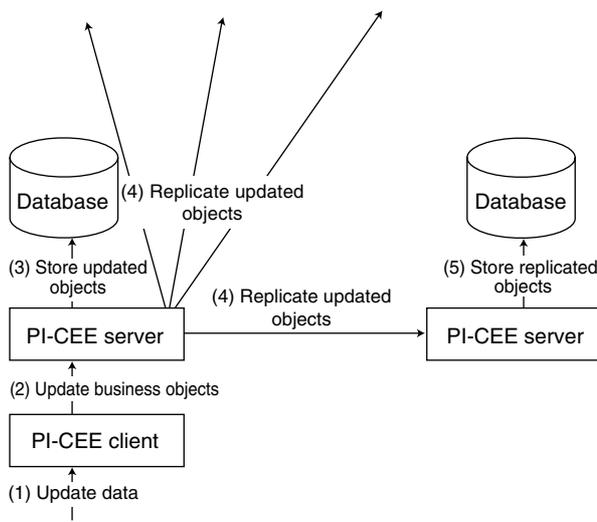


Figure 3 Multiple server replication architecture.

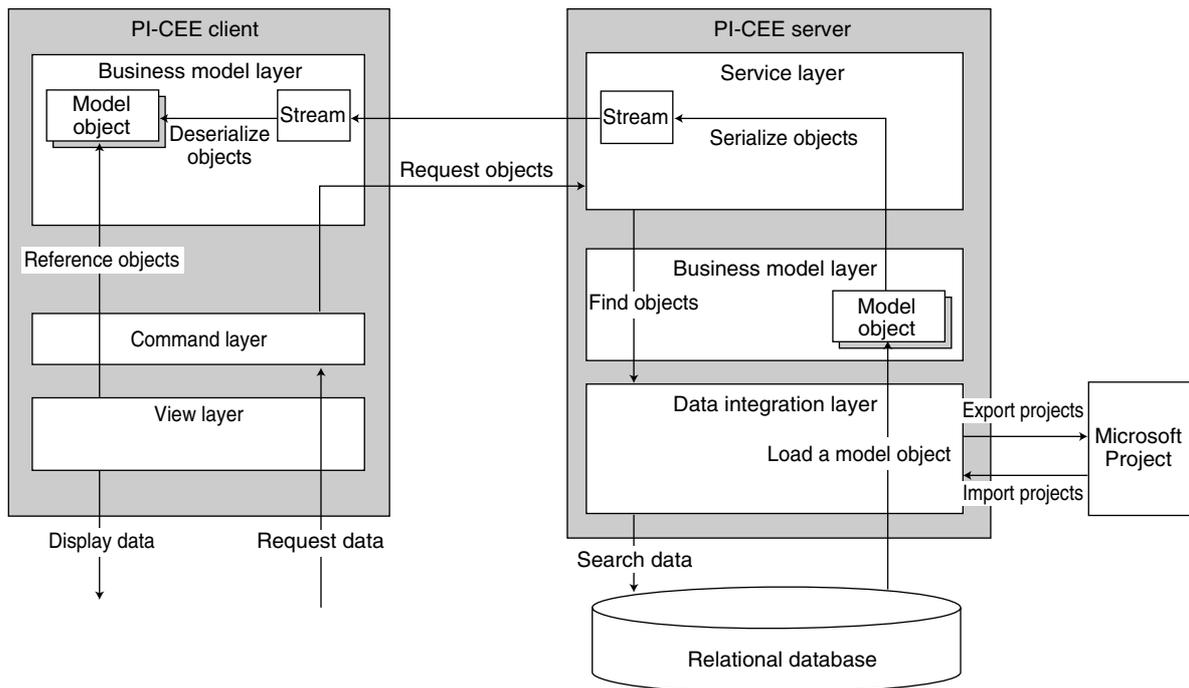


Figure 4  
System configuration of PI-CEE.

that because it is a multiple server system, it can work through a firewall and is therefore easy to apply.

### 3. Information sharing using the PI-CEE

The PI-CEE is an Internet-based project management system we have developed based on the CEE business model and the Server-Integrated architecture. The system configuration of the PI-CEE is shown in **Figure 4**. The whole system (including the servers and clients) is implemented in Java. On the PI-CEE server, project information can be registered in Microsoft Project, which is the de facto standard for project management tools. The registered project and task information is stored in the database and can be referenced by the PI-CEE clients. The project information in the database is transformed into a business model object called a “project object” in the data integration layer and is put in the business model layer. The service layer references objects in the business model layer and provides services requested by a PI-CEE client.

A PI-CEE client can be downloaded as an applet and executed with a WWW browser. A PI-CEE client contains the business model layer, which handles the same models as the server, the view layer, and the command layer (which enables the views to report their operation results to the models). The client requests data from the server using the model object. We used a communication infrastructure based on the socket for server-client communication. The server serializes the requested model object and transmits it to the client as a data stream. The client deserializes the received data, restores it to the model object, and displays it in a view.

Based on this architecture, the PI-CEE offers the following functions.

- 1) Sharing and managing of project work information:  
Software developers can confirm work progress by referencing client work lists. Also, the progress of an individual or the whole project can be monitored by inputting the progress situation.

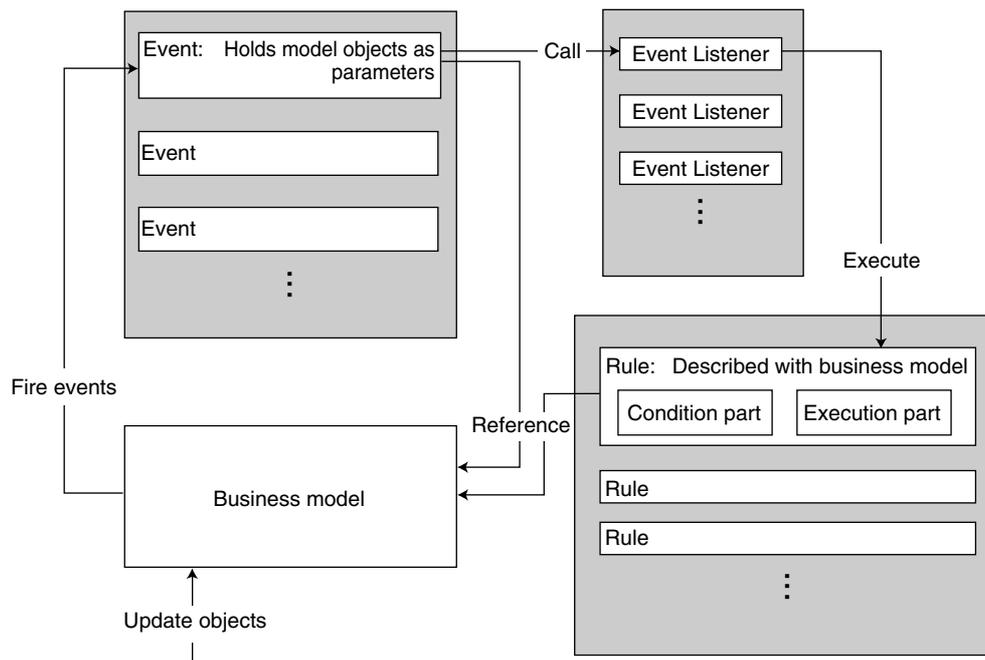


Figure 5  
Notification architecture.

2) Registration and referencing of artifacts related to each task:

Documents related to any task can be referenced and updated by registering them as task artifacts.

3) Progress notification functions:

Various progress notices are generated from the reliance relationships between tasks; these are then sent to the relevant persons in charge. The following are examples of notices.

- Notice of task completion (sent to the persons in charge of the following tasks)
- Warning that a task is not on schedule (sent to the persons in charge of the following tasks and to the project manager)
- Notice of an approaching task deadline

#### 4. Implementation issues

##### 4.1 Notification function using rule definition

As we mentioned in Section 2.3, various notifications are needed when managing projects.

The notification form differs according to the project conditions and user requirements. Therefore, we applied an event-driven rule execution architecture to provide a notification function. The architecture is shown in **Figure 5**. First, events corresponding to changes in the status of the business model are prepared. Each event holds business model objects related to the status change. Next, rule classes corresponding to each kind of notification are described. A rule consists of two parts: a condition part and an execution part. The contents and destinations are described in the execution part. Although rules cannot be customized dynamically, customization is facilitated by describing rules using relationships abstracted from the business model.

When a project manager starts the system, he or she chooses the rules that are required for the application. The chosen rules are added to the list of listeners of the events. When a data modification occurs, an event is triggered and the conditions of the rules which are triggered by the event are checked. If they meet the conditions for

executing the rule, notification is performed as described in the execution part.

## 4.2 Multiple server replication with conflict resolution

When two or more users update replicas of the same data at around the same time, a data conflict occurs. To maintain data consistency, we need a method to detect and resolve such conflicts.

### 4.2.1 Detection and resolution of data conflicts

The conflict detection method is shown in **Figure 6**. First, a record of the modification is written to the object when the object is updated. When an object is replicated, the server compares the replica that it already has with the one it has just received. Any difference in the corresponding parts of the updated records indicates a conflict.

To resolve such conflicts, we assign a priority to each update in advance so that we can determine which replica is correct. Users can select the priority rule which activates on the server in advance, but they cannot change the rule dynamically. When a conflict occurs, the system compares the priorities of each server in which

conflicting data objects are updated. The object of the highest priority is adopted as the update result. Thus, the system is able to maintain data consistency between servers.

### 4.2.2 Locking function corresponding to an application

Users can combine this replication method with a locking function according to the object model. We were able to apply the fundamental replication method without a locking function for task progress replication because such updates occur frequently and the quantity of data involved is small. The locking function should be used when replicating documents that are infrequently updated or when the update quantity is large.

## 5. Application and effects

We have applied the PI-CEE system to several projects that involve other companies and requested the persons in charge to provide daily progress information. Typically, these companies manage a large-scale project involving 1000 tasks with 500 people in charge of them. Through these applications, we have shown that the PI-CEE enables real-time information sharing on large-scale projects between widely dispersed locations and enables efficient project management.

Another software development project at Fujitsu involves users in a development office outside Japan. The users need to share project-related documents that can be updated at many different locations. Previously, this would be done using FTP; but by using the PI-CEE with the replication function, it is now possible to share progress information and documents updated at multiple locations.

## 6. Related work

There are several project management tools on the Internet. For example, ActionPlan<sup>5)</sup> enables distributed project management on the WWW with Java applets. Although it is very similar to the PI-CEE, it does not manage documents related to tasks. Microsoft (MS) Project is a suite

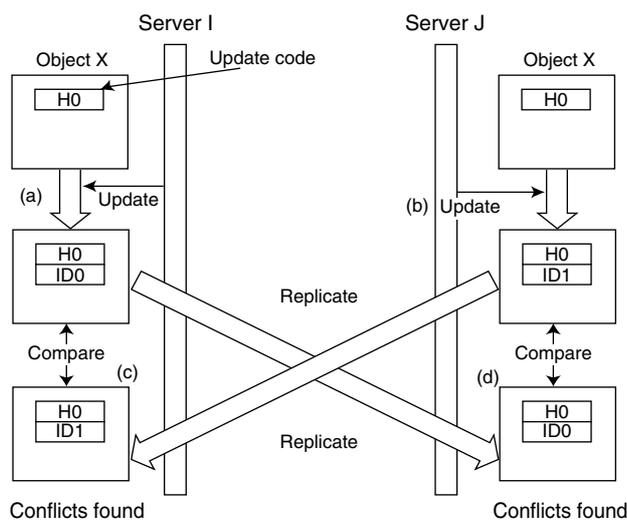


Figure 6  
Conflict detection.

of project management tools that share and manage project schedules. The latest version "Project 200" enables a schedule to be referenced and updated from a WWW client. Actually, the PI-CEE integrates MS Project for editors of project schedules. The PI-CEE enables easy customization of user interfaces with the CEE concept-business model and Server-Integrated architecture. Especially, the PI-CEE can define and manage relations between tasks and documents.

Several replication tools have been implemented in databases and document management tools. Oracle is a database management system (DBMS) which provides multiple server replication.<sup>6)</sup> It also provides a conflict-resolution method. However, these solutions are applied to the database structure, not to the object models. Lotus Notes Domino is a document management system which performs document-based replication using conflict-detection and conflict-resolution methods<sup>7)</sup> similar to those used in the PI-CEE. The conflict-resolution of Notes Domino uses the data's time stamps. On the other hand, the PI-CEE allows a user-defined rule to be used for conflict-resolution. Also, unlike Oracle and Notes Domino, the PI-CEE provides a user lock service to decrease the conflict ratio.

## 7. Conclusion

The PI-CEE enables efficient distributed project management via the Internet using the CEE business model and the Server-Integrated architecture. To improve the efficiency, we have added two functions to the PI-CEE. First, we added an easy-to-customize notification function using business models and an event-driven rule system.

Second, we added a function for multiple server replication that maintains data consistency by detecting and removing update conflicts.

At present, we are developing a version of the PI-CEE that is integrated with ProjectWEB, which is a standard Web application used by systems engineers at Fujitsu. This integration will provide more global information to support the work of systems engineers.

In the future, more and more cooperative work will be done by sharing information over the Internet using systems such as the PI-CEE.

## References

- 1) S. Uehara et al.: Enterprise Model-based Software Architecture with Server Component Integration. EDOC98, pp.356-363, 1998.
- 2) J. Martin: Information Engineering Book 2: Planning & Analysis. Prentice-Hall, 1990.
- 3) The Air Force Wright Aeronautical Laboratories: Integrated Computer Aided Manufacturing (ICAM) Architecture. Part 2, Vol.6, Function Modeling Manual (IDEF0), June 1981.
- 4) T. Okubo et al.: An Optimistic Method for Updating Information in Distributed Collaborative Work. COMPSAC'98, pp.400-405, 1998.
- 5) L. Y. Eric : Distributed Java Applets for Project Management on the Web. IEEE Internet Computing No.3, pp.21-26, 1997.
- 6) Oracle8i Replication Release 8.1.5, Oracle. <http://technet.oracle.com/doc/server.815/a67791/toc.htm>
- 7) Notes Domino Administration Help. Lotus. [http://doc.notes.net/domino\\_notes/5.0.3/help5\\_admin.nsf](http://doc.notes.net/domino_notes/5.0.3/help5_admin.nsf)



**Takao Okubo** received the B.S. and M.S. degrees in Computer Science from Tokyo Institute of Technology, Tokyo, Japan in 1989 and 1991, respectively. He joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 1991 and has been engaged in research and development of enterprise systems supporting distributed collaborative work. He is a member of the Institute of Electronics, Information and Communication Engi-

neers (IEICE).



**Hirotaka Hara** received the B.S. degree in Information Science in 1984 and the Ph.D. in Information Engineering in 1992 from the University of Tokyo, Tokyo, Japan. He joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 1984 and has been engaged in research and development of artificial intelligence and distributed enterprise systems. He is a member of the IEEE. He received the IPSJ Convention Award and the JSAI

Annual Conference Award in 1993.



**Takahide Matsutsuka** received the B.S. and M.S. degrees in Computer Science from Tokyo Institute of Technology, Tokyo, Japan in 1994 and 1996, respectively. He joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 1996 and has been engaged in research and development of distributed enterprise systems and software architecture.