

## **Practical Virtual Use Method of CGI Program for User Creation Personal Portal Page Using DACS Web Service**

**Kazuya Odagiri<sup>1</sup>, Shogo Shimizu<sup>2</sup> and Naohiro Ishii<sup>3</sup>**

<sup>1</sup> *Yamaguchi University, Ymaguchi-shi, Ymaguchi, Japan*

<sup>2</sup> *Gakushuin Women's College, Software, Shinjuku, Tokyo, Japan*

<sup>3</sup> *Aichi Institute of Technology, Aichi, Japan*

<sup>1</sup>odagiri@yamaguchi-u.ac.jp; <sup>2</sup>shogo.shimizu@gakushuin.ac.jp; <sup>3</sup>ishii@aitech.ac.jp

### **ABSTRACT**

A personal portal, which is an entrance wherein each user can acquire the information that s/he is interested in on a network, is often used as an alternative means of communication. However, there were a number of problems in the existing personal portals. For example, because the Web page as a personal portal was generated by the program located on the specific Web server which is managed by a system administrator, it was not always ideal for all users. To solve this kind of problems, we developed two Web Service functions, which are realized on the network by introducing the Destination Addressing Control System (DACS) Scheme. These two Web Service functions are as next. The first is the function to extract the data for each user from a database and display it on the Web browser. The second is the function to retrieve the data for each user from a document medium and display it on the Web browser. Through these Web Service functions, each user can easily create a customized personal portal that displays personal information. Then, the above two functions were extended to manage information not only for each user but also for each group of users and for all users, and the extended two functions were integrated as a DACS Web Service. By using the DACS Web Service, each user can create and customize the Web page as a personal portal for practical usage in an individual organization. However, when the use of it in a real network is assumed, it is difficult to make the Web page as the personal portal with a complicated function by the present web page as the personal portal using DACS Web Service. Therefore, in this study, functional extension is performed. To be concrete, we realize the method of using the Common Gateway Interface (CGI) program opened to Internet by a user unit, and improve the present Web page as the personal portal using DACS Web Service by the method

**Keywords:** Web Services; Common Gateway Interface; Destination Addressing Control System

## 1. INTRODUCTION

Static Web pages are often used as an alternative means of communication in addition to e-mail by an unspecified number of users. However, they are unsuitable for communication among individual users for their individual purposes or interests. As an alternative means of communication, the personal portal is realized by a Web Service; it allows each user to change the contents of the Web page dynamically. The term “Web Service” refers to a service that is provided to users through a Web server, and the term “portal” often indicates a Web page for searching information [1][2] such as with Google or Yahoo. The meaning of the phrase “personal portal” used in this paper is different from that of the above “portal” as just described. Here, “personal portal” refers to an entrance where each user can acquire information that s/he is interested in on the network; it can display different information for individual users dynamically on a Web browser by using a program such as the Common Gateway Interface (CGI) [3]. As examples, Netvibes [4] and Google [5] are introduced. Therefore, personal portals are suitable for communication among individual users for their respective purposes or interests. To display the necessary information for an individual user on a Web browser, the information is searched and extracted from databases on the network, after which each user is notified by a program such as CGI. If each database is distributed on the network, the program used to retrieve the information is large and complicated. Given that the program is installed by a system administrator and cannot be freely changed by a user, using a Web page as a personal portal is not always ideal for all users. To solve this problem, a new form of personal portal, which each user can create and customize easily, is necessary. The Destination Addressing Control System (DACS) Web Service is proposed and examined to realize such a personal portal. This DACS Web Service is implemented by extending two types of Web Service functions, which are realized on the network introducing the DACS Scheme. Using this DACS Web Service, each user can easily create and customize a Web page as a personal portal for oneself.

The DACS Scheme is a network management scheme. The basic principle of the DACS Scheme is that the entire network system is managed through communication control for each user on the client machine [6]. Moreover, functional extension is achieved by allowing the communication control for each user to coexist with communication control for each client machine to use in a user group [7-8]. In addition, Secure DACS Scheme is proposed to solve a security problem [9]. Two types of Web Service functions are primitive functions as fundamental elements of DACS Web Service, which are also realized on the network introducing the DACS Scheme. However, they have following two functions. In the first function, data that is stored in the database and is dispersed on the network can be used efficiently [10]. In the second function, data that is stored in a document medium such as PDF or simple text files can be used efficiently. Using both functions, when different users input the same URL into the Web browser, the different information for each user is searched and extracted from the database or document medium and displayed on the Web browser. By

incorporating various kinds of URLs into a static HTML, it has become possible for each user to create and customize a Web page as a personal portal easily [11]. In addition, by extending and integrating two types of Web Service functions, we realized the DACS Web Service [12]. The DACS Web Service uses data stored in a database or document medium to enable individual users, groups of users, and for all users to send and receive information through the integrated user interface. However, when the use of it in a real network is assumed, it is difficult to make the Web page as the personal portal with a complicated function by the present web page using DACS Web Service. Therefore, in this study, functional extension is performed. To be concrete, we realize the method of using the Common Gateway Interface (CGI) program opened to Internet by a user unit, and improve the present Web page using DACS Web Service by the method.

## 2. RELATED WORKS AND MOTIVATION

Existing personal portals are Web pages that serve as unified windows for data stored in the information systems of organizations. In the field of the personal portals, various studies have examined [13], [14], [15]. In addition, personal portals are developed in individual organizations for practical purposes, and commercial software packages are also used. The system known as “HInT” is an example developed in a university [16]. “Blackboard” [17] is an example of a software package often used in a university. In these cases, the basic mechanism of personal portals is described by the sequence of processes from (1) to (9) shown in Fig. 1. The arrows with the dotted lines show the actions that the user performs in the Web browser, whereas those with solid lines show the flow of processing. This mechanism is explained in sequence as follows. First, the user enters a URL as input into the Web browser (1). Next, the Web server corresponding to the URL is accessed (2), and the program corresponding to the URL is executed. At this point in time, the program on the Web server side does not have the user information (user name and password). Therefore, an input demand for user information is executed on the Web browser side (3). Subsequently, the user enters the user information as input (4), which is sent to the program on the Web server (5). In existing mechanisms, user authentication is performed at this point in time. If access is permitted, information related to the user is searched from data that is accumulated in advance (6). To accumulate the data, there are methods using relational databases or document medium. Following search, the data related to each user are extracted (7). Programs on the Web server side programs that receive the data, such as CGI, generate a Web page dynamically from the data. Then, the Web page is sent to the Web browser side (8). Thus, each user can view information which is related to him/herself and displayed in the form of a Web page (9).

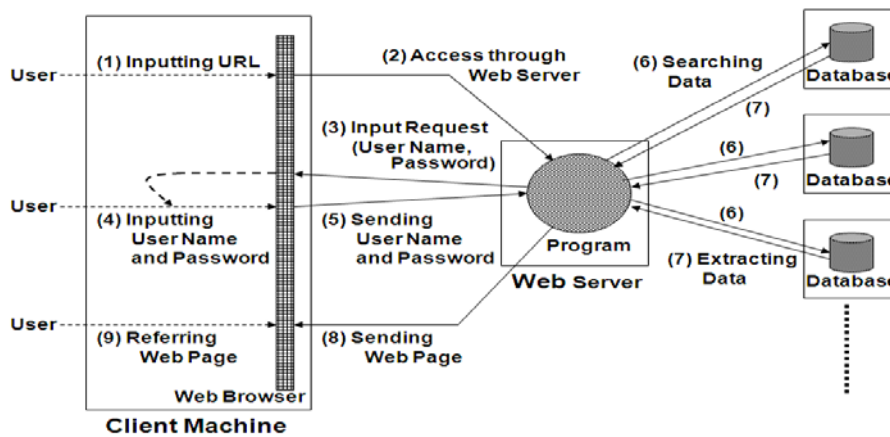


Fig. 1 Basic Mechanism of Existing Personal Portals

Moreover, as one of the methods to realize the personal portal, there is a method by application frontend integration in Service Oriented Architecture (SOA) [18][19]. As shown in Fig. 2, SOA, which is the study of information system integration in organizations such as enterprises and universities, is realized by using WebAPIs. The mechanism is different from that of Fig. 1 in at least two points.

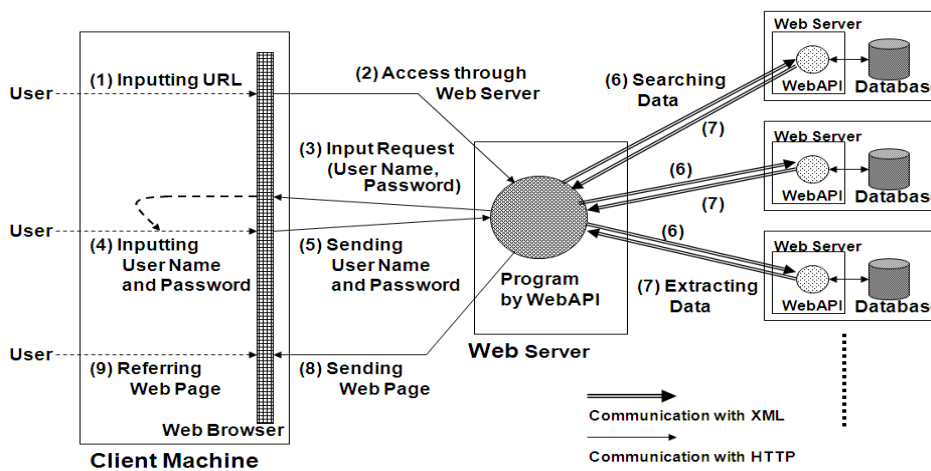


Fig. 2 Mechanism of Personal Portals by WebAPI

- (1) The program on the Web server that the user accesses first is implemented with Web APIs such as Simple Access Object Protocol (SOAP) [20][21][22] and Representational State Transfer (REST) [23][24][25].
- (2) Communication between (6) and (7) is performed with use of Extensible Markup Language (XML) [26][27] through each Web server distributed on the local area network.

The essential commonality of the mechanisms explained by here in Section 2 is that the program on the Web server extracts data from some database and generates the Web page as a personal portal. However, there were two problems with this general scheme.

(Problem 1) The program on the Web server is installed by a system administrator, and usually cannot be changed. Given that this program generates the Web page, it is not always easy for the user to customize the personal portal. Although this can be done in some cases within the specifications of the program, not all users can customize the Web page because the customization beyond these specifications is impossible.

(Problem 2) Although data from a database are used in existing personal portals, the data are often stored in the form of particular document medium in individual organizations. However, existing personal portals cannot use such data easily.

To solve these problems, two types of Web Service functions based on the DACS Scheme were proposed. By using the first function [10], problem 1 was solved. This enabled each user to create a Web page as a personal portal using data from the database. By using the second function [11], problem 2 was solved. This allowed the user to generate a Web page using data in the document medium. Using these functions, when different users enter same URL as input into Web browser, different information can be searched and extracted from the database or document medium and displayed on the Web browser. However, these functions allow one to deal only with information for individual users. In practice, these functions are insufficient in an organization.

Therefore, we proposed the DACS Web Service for improving the above problems. It has the functions of using information for groups of users and all users based on two functions of existing Web Service. In addition, the scheme integrated these two functions, and generated unified and personal user interfaces for each user. The personal portal for practical use in an individual organization was thus realized.

The following were improvements obtained over existing personal portal by incorporating the DACS Web Service into practical personal portals.

(1) Given that each user can create a Web page as a personal portal that fits his/her preference without the limitation of the program on the Web server, a user interface that is easy to use and suitable for practical use is realized

(2) Data stored not only in a database but also in a document medium, are used easily.

Next, we examined a new element for functional extension by access control technology for the data corresponding to position and the role of the user. Role-based Access Control (RBAC) [28], [29], [30] is available for this kind of access control. As RBAC enables access control corresponding to the role of individual user, access control for groups of users as well as all users is functionally possible.

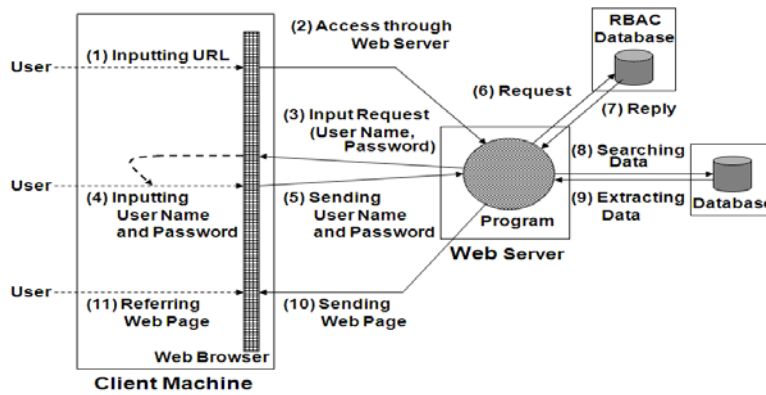


Fig. 3 Mechanism of Personal Portals with RBAC

The mechanism of personal portals when applying RBAC is shown in Fig. 3. The essential difference between Fig. 3 and Figs. 1, and 2 is that processes (6) and (7) are added in Fig. 3. These processes, which involve requesting and extracting access control information, must be performed between the gateway program on the Web server and the RBAC server that stores the access control information. When RBAC is applied to the two types of Web Service functions based on the DACS Scheme, the system configuration does not have the gateway program. Therefore, when an inquiry to a database is sent, the DACS Client must be extended to request and extract access control information from RBAC before the inquiry. In addition, because the DACS Scheme is the method used to maintain access control information in the DACS Server, but not in RBAC Server, access control information must be managed in two servers. This extension goes against the concept of the DACS Scheme, which is to control an entire network using a simple mechanism. Therefore, it is inappropriate to apply RBAC to two types of Web Service function based on the DACS Scheme. The extension was carried out in a form suitable to the DACS Scheme. As the result, we realized the Web page as a personal portal using the DACS Web Service, which was realized on the network introducing the DACS Scheme. *However, when the use of the Web page as the personal portal in a real network is assumed, it is difficult to make the Web page as the personal portal with complicated functions by the present web page using DACS Web Service. Therefore, in this study, functional extension is performed.*

In Section 3, a synopsis of the following is provided to explain the DACS Web Service: the DACS Scheme and the two types of functions based on the scheme, as well as the system configuration by these functions to realize the customized personal portal. In Section 4, the actual DACS Web Service is proposed. In particular, the features that are improved relative to the two types of the conventional Web Service functions are explained. In Section 5, after *the function of using the Common Gateway Interface (CGI) Program opened to Internet by a user unit, the virtual use method of the CGI using the function is shown.*

### 3. EXISTING DACS SCHEME

#### 3.1 Functions of the DACS Scheme

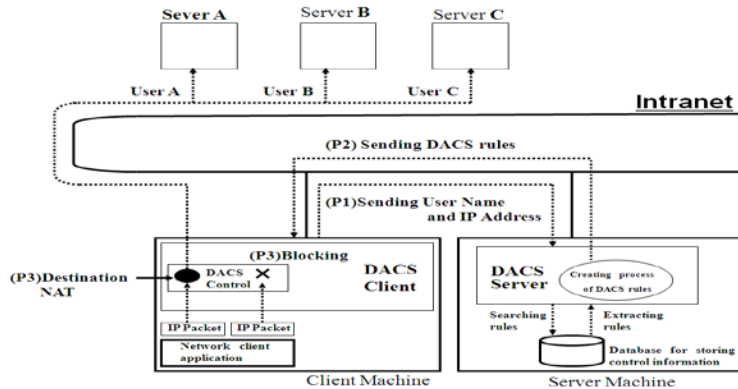


Fig. 4 Function of the DACS Scheme (1)

First, we provide a summary of the DACS Scheme. Figs. 4 and 5 show the functions of the network services according to the DACS Scheme. The DACS Server, which is located on a server machine, refers to the server software of the DACS Scheme. The DACS Client, which is located on each client machine, is the client software of the DACS Scheme. The DACS Control and DACS SControl in the DACS Client denote the control mechanisms for communications that are sent from the client software such as a Web browser. The DACS Control controls the normal communication from client software. On the other hand, the DACS SControl converts normal communication from the client software to Virtual Private Network (VPN) [31], [32], [33], [34] communications.

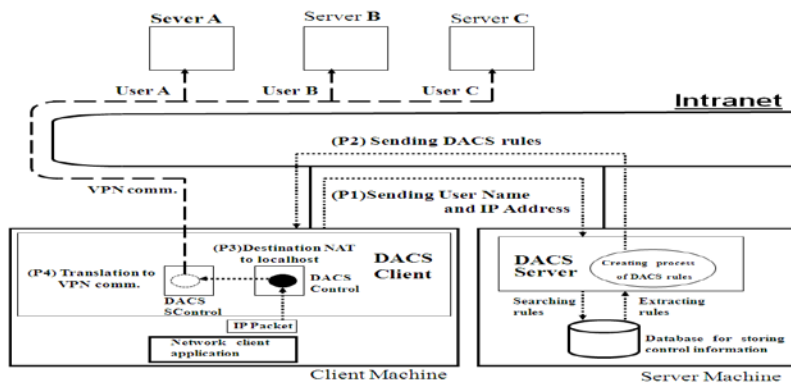


Fig. 5 Function of the DACS Scheme (2)

The DACS rules are the rules for controlling the DACS Control and DACS SControl. At the time of (a) or (b) below, the DACS rules (rules defined for each user) are distributed from the DACS Server to the DACS Client.

- (a) When the user logs into the client machine
- (b) When a notice is delivered from the system administrator

According to the distributed DACS rules, the DACS Client performs operation (1), (2) or (3) operations. Thereafter, communication control of the client machine is performed for each user who has logged in.

- (1) The destination information on the IP Packet, which is sent from the client application, is changed by Destination Network Address Translation (NAT).
- (2) The packet from the client machine, which is sent from the client application to out of the client machine, is blocked by a packet filtering mechanism.
- (3) Communication between a client machine and a network server is supported by VPN with the port forward function of Secure SHell (SSH) [35], after the destination of the communication is changed to localhost (127.0.0.1) by function of (1).

An example of case (1) is shown in Fig. 4. Here, communication control by the system process (P3) is performed after the system proceeds from (P1) to (P2). Thereafter, the system administrator can distribute a communication of the user who has logged in to a specified server among servers A, B, or C. An example of case (2) is also shown. When the system administrator wishes to forbid a user to use Mail User Agent (MUA) as the client software of the control target, this is achieved by blocking the IP Packet with specific destination information.

Next, an example of case (3) is shown in Fig. 5. Here, communication control by the system proceeds (P3) and then, (P4) is performed after the system proceeds from (P1) to (P2). The communication is supported by VPN, and the system administrator can distribute the VPN communication of the user who has logged in to a specific server (A, B, or C). In the DACS Scheme, when a network service that handles personal and confidential information is controlled, the communication between a network server and client machine must be encrypted for protection. This function is used at that time.

In order to realize the DACS Scheme, communication controls on the client machine are performed by the DACS Protocol, as shown in Fig. 6. The DACS rules are distributed from the DACS Server to the DACS Client in (a) of Fig. 6, and applied to the DACS Control and the DACS S Control in (b) and (c) of Fig. 6. Normal communication control, such as modification of the destination information or communication blocking, is performed at the network layer in (d) of Fig. 6.



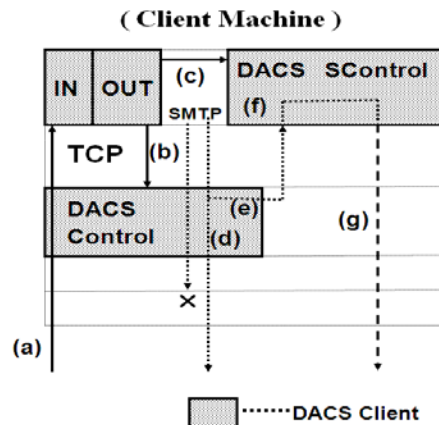


Fig. 6 Layer setting of the DACS Scheme

In case (1), as the function of destination change, the DACS rules are only applied to the DACS Control. The DACS rules consist of both destination-a (the destination IP address-a, destination port-a) before destination change and destination-b (destination IP address-b, destination port-b) after destination change. The communication sent from the client software is sent to destination-a. Thereafter, the destination of the communication is changed to destination-b by the control of destination NAT in the DACS Control, and communications are sent to the network server with this destination. In case (2), as the function of communication blocking, the DACS rules only are applied to the DACS Control similarly. The content of the DACS rules consists of destination-c (destination IP address-c, destination port-c) as the communication-blocking target. When the communication sent from the client software is destination-c, the communication is blocked by packet filtering in the DACS Control. In case (3), as the function of VPN communication, the DACS rules are applied to both the DACS Control and the DACS SControl. The content of the DACS rules consists of the following two rules.

(r1) Rules with both destination-d (destination IP address-d, destination port-d) before destination change and destination-e (127.0.0.1, port-e) after destination change

(r2) Rules with destination-f (destination IP address-f, destination port-f)

The DACS rules as (r1) are applied to the DACS Control and the DACS rules as (r2) are applied to the DACS SControl. Then, when a communication is supported by VPN, it is sent from (f) to (g) via (e). The VPN communication of (g) is sent by the DACS SControl. Using the port forwarding function of SSH, VPN communication which tunnels and encrypts the communication between a network server and a client machine with the DACS Client is realized. Normally, to communicate from the client software to a network server using port forwarding of SSH, it is necessary for the local host (127.0.0.1) to be indicated on that software as the

communicating server. Using this function, transparent use of a client machine as a characteristic of the DACS Scheme is not failed. That the use of a client machine is transparent means that even if the configuration of the network servers is changed, the client machine can be used continuously without changing its setups. Communication control for this function is performed with the DACS SControl by SSH port forwarding. Through the use of these two functions, VPN or non-VPN communication for each network service can be selected for each user. In the case where non-VPN communication is selected, the communication control is performed by the DACS Control, as shown in (d) of Fig. 6. In the case where VPN communication is selected, the destination of the communication is changed by the DACS Control to the localhost. Then, the port number is changed to the number assigned for each communication. Subsequently, the communicating server is determined and the VPN communication is sent by the DACS SControl, as shown in (g) of Fig. 6. In the DACS rules applied to the DACS Control, the localhost is indicated as the destination of communication. In the DACS rules applied to the DACS SControl, the network server is indicated as the destination of communication. Thereafter, by changing the content of the DACS rules applied to the DACS Control and the DACS SControl, the control in the case of VPN communication or non-VPN communication is distinguished.

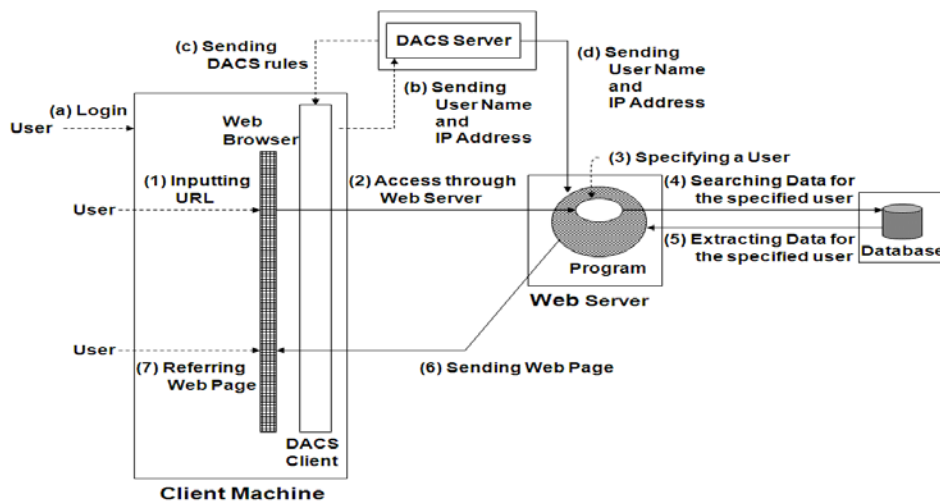


Fig. 7 Function Using Data from Database

### 3.2 Two Types of Web Service Functions based on DACS Scheme

In this Section, the network service corresponding to the DACS Scheme is explained. In the existing DACS Scheme, the communication control information for each user and each client machine has been maintained in the DACS rules on the DACS Server. By applying that information for communication control to the DACS Client (DACS Control) located on the client machine, communication from the client machine is controlled. As a result, the communication control mechanism on the network server is not required. However, on a network introducing

the DACS Scheme, if a correspondence list of a client machine's IP address and user name is passed to the network server, it becomes possible to identify which user is sending the communication from a client machine. As a result, it becomes possible for a program on the network server to perform different processing for each user. A concrete example is the correspondence of the Web Service to the DACS Scheme. As the example, two types of Web Service functions based on the DACS Scheme are described as follows.

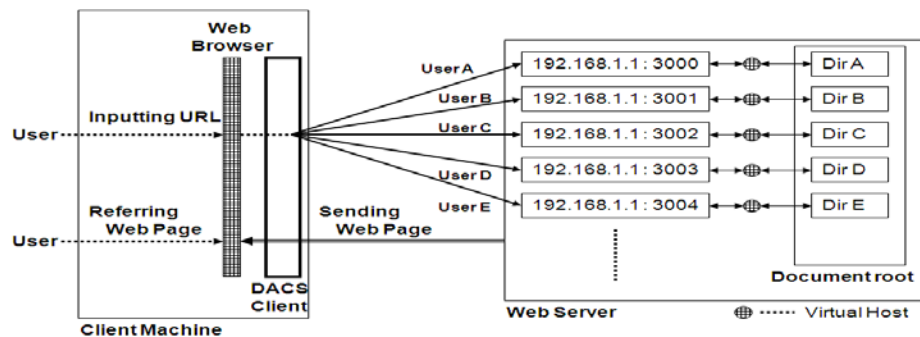


Fig. 8 Function Using Data from Document Medium

First, the function to retrieve data from the database is developed. To realize this function, the DACS Scheme must be extended, and the program on the Web server must be implemented in correspondence to the extended DACS Scheme, as shown in Fig. 7. In the existing DACS Scheme, after a user logs in to a client machine (a), the user name and IP address are sent to the DACS Server (b). Then, the DACS rules are sent back to the DACS Client (c). In the extended DACS Scheme, the user name and IP address are sent to the program on the Web server. A characteristic of the extended DACS Scheme is that the server side program on the Web server identifies the user by checking the login information and the source IP address from the client machine, and changes the processing of the program for each user. When different users access the program with the same URL, different information for each user can be searched and extracted from the database, and can be displayed on the Web browser. On the extended DACS Scheme, this new function is performed through the processing from (1) to (7).

Next, the function to retrieve data from the document medium for each user is developed. In the network with the DACS scheme, a different IP address and TCP port can be assigned for each user using the same host name. Therefore, a different document medium with the same file name on a different Web server can be referred for each user by entering the same URL into the Web browser as input. When this principle is combined with the function of a virtual host, which is equipped as a Web server, it is possible to use the Web server in the manner shown in Fig. 8. By the function of the virtual host, multiple sockets (IP address and TCP port) can be assigned to one Web server. The referred document can be changed for each socket.

First, in the document root of the Web server in Fig. 8, directories (i.e., Dir A, B, C, D....) are prepared for each user. By the function of the virtual host, each directory is connected to each socket as a pair. By changing the TCP port number (3000, 3001, 3002....) for an IP address (192.168.1.1), the sockets corresponding to each directory are prepared. Next, movement on this mechanism is described. First, a user enters a URL into a Web browser as input. When the URL is entered as input by User A, the file in Dir A that is connected to the socket (192.168.1.1:3000) is referred. When User B enters the URL as input, the file in Dir B that is connected to the socket (192.168.1.1:3001) is referred. For User C, the file in Dir C that is connected to the socket (192.168.1.1:3002) is referred. When a document medium with the same name exists in each directory (i.e., Dir A, B, C....), each user views different contents by entering the same URL into the Web browser as input. From the user's point of view, the same function as that of the Web Service based on the extended DACS Scheme is realized. For the information sender, because it is possible to deliver information to the specific user by uploading the document medium to the predetermined directory, the range of information usage broadens greatly. Because the information sender can easily describe the content of the document medium, it is possible to communicate information with much expressive power and impact.

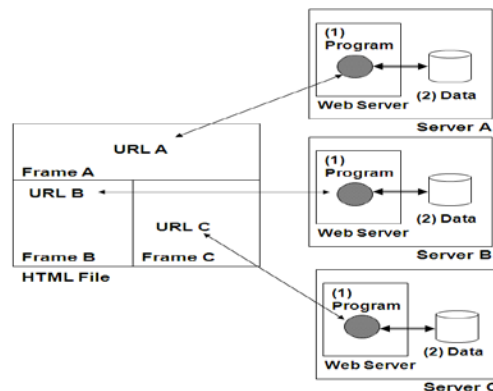


Fig. 9 Type1 System Configuration

### 3.3 System Configuration to Realize Personal Portals

Use of the function to retrieve data from the database enables the Type1 system configuration, which is shown in Fig. 9. First, server A-C, which have programs based on the extended DACS Scheme, as shown in (1) of Fig. 9, and data in the database as shown in (2) of Fig. 9, are distributed on the network. In the extended DACS Scheme, information related to each user is displayed on the Web browser by inputting the URL. One window of the Web browser is divided, for example, into three frames (Frames A-C). A static HTML file with each URL (URLs A-C) in each frame is created for displaying the Web page as a personal portal. The static HTML file is placed on the Web server or the client machine. When the static HTML file is opened through the Web browser, the information extracted from each server is distributed on the Web browser. In the extended DACS Scheme, the URL corresponding to each server is only incorporated in the static HTML file. Thus, when the static Web page is created, the user can easily create the customized Web page as a personal portal.

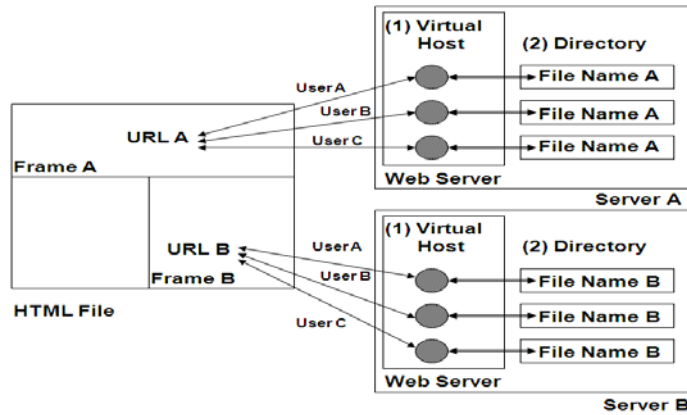


Fig. 10 Type2 System Configuration

As shown in Fig. 10, the Type 2 system configuration can be realized by using the function to retrieve data from the document medium. Each URL (URLs A, B) is prepared. The URL and the information that can be viewed after entering it in the Web browser as input are sent to the users. For example, the URLs for acquiring the homework in one classroom and for communicating from the office to each individual user are enumerated as different kinds of URL. By entering this URL into the Web browser as input, the file with the same name (File Name A or B) is referred. The file is stored in each directory for each user, and each user can view the stored file in the directory. Therefore, a customized personal portal with a static HTML File that is the same as Type 1 configuration can be created. As a result, by allowing both system configurations to coexist, a Web Service is realized in which a user can use information on the network regardless of the form of its storage. (Fig.11)

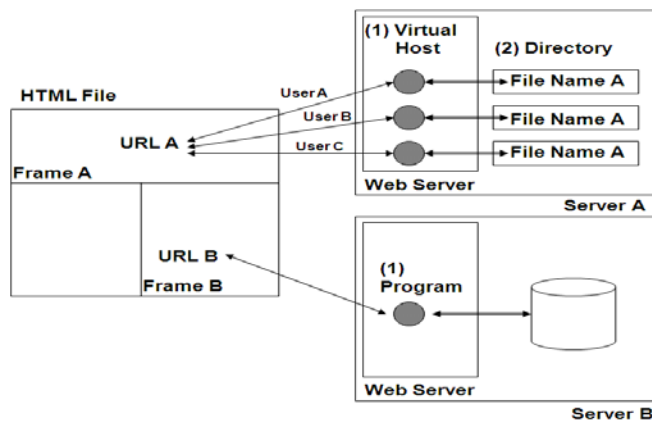


Fig. 11 Web Service by Two Functions

## 4. DACS WEB SERVICE

In Section 3, we provided a synopsis of the DACS Scheme. In addition, we described two types of Web Service functions that could be realized in the DACS Scheme or extended DACS Scheme, and the system configuration according to these two types of functions for realizing a customized personal portal. As these two functions are used to manage the information of each user, it is insufficient to create a customized personal portal for use in one group of users. Therefore, the following three functions are necessary.

- (1) A function to manage the information of each user.
- (2) A function to manage the information of user groups.
- (3) A function to manage the information of all users.

Among these three functions, the latter two do not exist. Therefore, in this section, these two types of Web Service functions are integrated after extending the function to compensate for this insufficiency. The DACS Web Service, which is achieved by this integration, is proposed to realize the personal portal. Moreover, we explain the method to acquire the data they seek.

### 4.1 Operational Procedures for Personal Portal

In this section, operational procedures to create a web page for a personal portal are explained. To be concrete, a system administrator's procedure and a user's procedure are described.

(System Administrator's Procedure)

- (Step1) After a system administrator assesses where data in information systems on a LAN exists, s/he make a list with contents of the data and the URLs to acquire it.
- (Step2) The list is uploaded and released on a Web Server to let users know where the data they seek exists.
- (Step3) The URL to view the lists notified to users through some methods such as the following.
  - (1) Notice on a bulletin board
  - (2) E-mail
  - (3) Notice on a Web page that many users view
  - (4) Oral notification by telephone

In Fig.12, an example of such a Web page is described. In this page, content of the data and URLs for viewing the data are described side by side on one line. When the DACS Client is installed on the client machine, the user can view this page through the function of the DACS SControl. That is, this page is accessed securely by VPN communication. When the DACS Client is not installed on the client machine, the user can not view this page. By clicking the URL of the

page on a client machine with DACS Client, the user can easily acquire the data they seek. By user's procedure as follows, each user can create the web page for the personal portal.

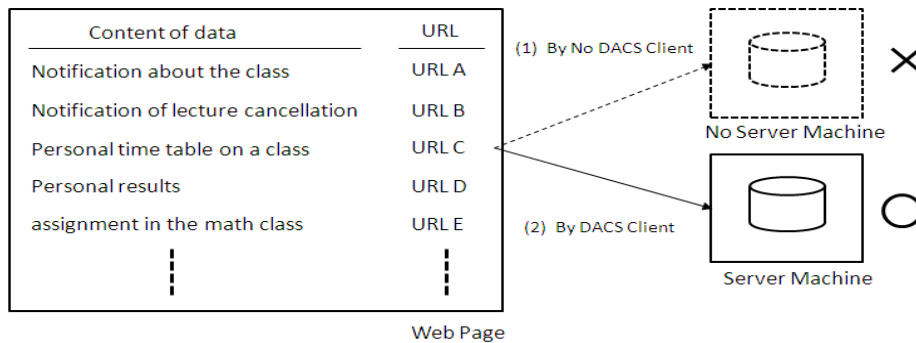


Fig. 12 Web Page to show Where Data and URLs Exist

(User's Procedure)

(Setp4) Each user inputs access the above list by inputting the specified URL with the DACS Client installed in a client.

(Step5) Each user views the list, and acquires the URL for getting the necessary information.

(Step6) Each user creates a static HTML file for the personal portal by implementing the above URL.

(Step7) Each user uploads the above HTML file.

After these steps, it is possible to view the web page for the personal portal.

## 4.2 Synopsis of DACS Web Service

In Fig.13, an overview of the DACS Web Service is shown. In this figure, the function to retrieve data from a database of an information system is shown as Function  $\alpha$ , and the function to retrieve data from a document medium such as a simple text or PDF file is shown as Function  $\beta$ .

From here, a series of DACS Web Service's movement is explained. First, as URLs to use the DACS Web Service, three kinds of URLs as follows are prepared for Function  $\alpha$  and Function  $\beta$ .

- (1) URL for acquiring data of each user
- (2) URL for acquiring data of each group
- (3) URL for acquiring data of all users

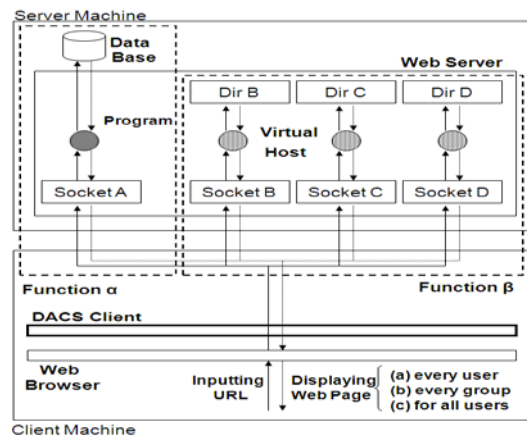


Fig. 13 DACS Web Service

By using these URLs, necessary information is acquired. The concrete functions by using these URL are described in Fig.14. By the URL (1), the existing two functions described in the Fig.7 and Fig.8 are used. By the URL (3), normal web access is done. In the case of Function  $\alpha$ , the data is extracted from a database. For example, web access without three processes of (3),(4) and (5) in Fig.1 is given. In the case of Function  $\beta$ , the data is extracted from a static document medium. This is a normal web access that we do in accessing a home page on Internet. Because the above cases are realized by using the existing technology, it is not necessary to explain it especially.

URL	Function $\alpha$	Function $\beta$
(1)	Acquiring data by using the function of Fig.7	Acquiring data by using the function of Fig.8
(2)	Acquiring data by using the extended function of Fig.7	Acquiring data by using the extended function of Fig.8
(3)	Normal web access by extracting data form a database	Normal web access by using a static document medium

Fig. 14 Details of Function  $\alpha$  and Function  $\beta$

However, in the case of using the URL (2), the extended functions from the functions of Fig.7 and Fig.8 are used. Therefore, technological explanation needs to be described. In Fig.15 the extended function from the function of Fig.7 is described. Processes from (4) to (7) in Fig.17 are additional processes from the function of Fig.7. First, a user name, which is sent from the DACS Sever by the process (d), is sent to an authentication server such as LDAP (5). The authentication server receives it, and returns the group name that the user belongs to by extracting it form an authentication database in process (5). The program on the Web server receives the group name, and sends it to the database in process (6). The database receives the



group name, and extracts data for the group from the database. By process (7), the data is sent to the program on the Web server. The program receives the data, and sends a Web page to the Web browser on the client. In Fig.16, the extended function from the function of Fig.8 is described. This extended function is realized by changing the setups of the DACS rules. In Fig. 7, each user passes through the specified sockets and accesses the specified directory. Different multiple users do not access the same directory. However, by assigning the same DACS rules to each user belonging to the same group, the function described in Fig.16 is realized. To be concrete, when multiple users from UserA1 to UserA5 inputs same URL to the Web Browser, the same static document medium in the directory of GroupA is accessed through the specified sockets (192.168.1.1:4001). In the same way, When multiple users from UserB1 to UserB5 inputs same URL to the Web Browser, the same static document medium in the directory of GroupB is accessed through the specified sockets (192.168.1.1:4002). After these above access, each Web page for each user's group is sent to the Web browser on the client.

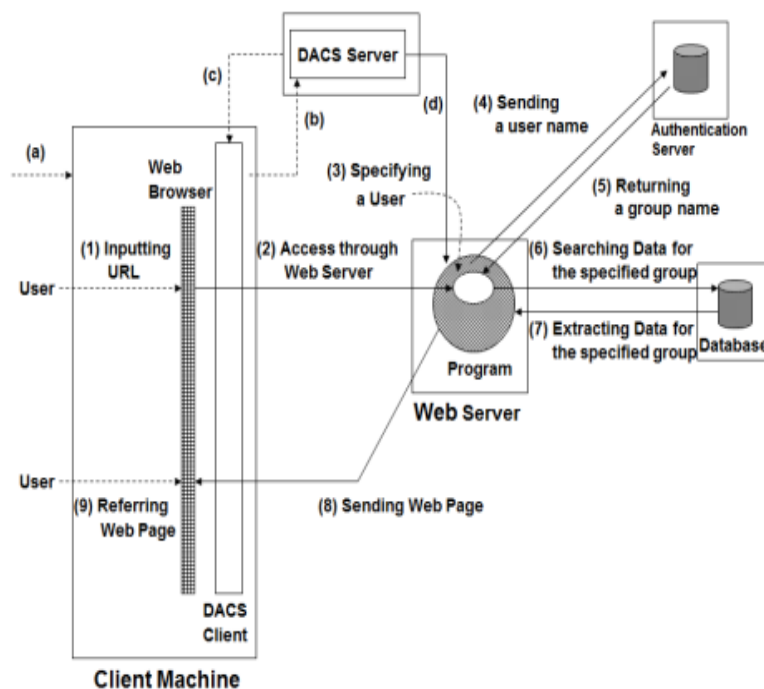
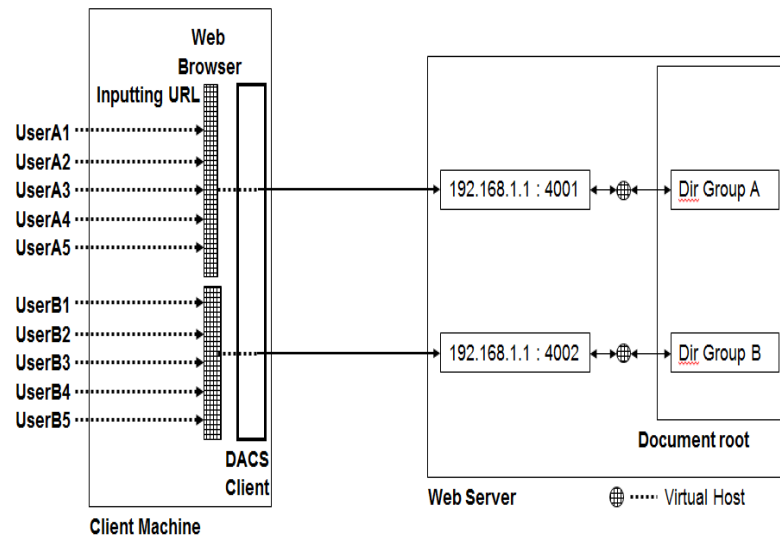


Fig. 15 Extended function of Function alpha

In this way, a user can use data from an information system or document medium dispersed on the network, without being aware of which function is being used. In other words, a user can use information regardless of its form or place of storage, if a user knows the URL and the kind of information acquired by that URL. Regardless of whether Function  $\alpha$  or Function  $\beta$  is used, data are displayed on the Web browser after entering the URL as input. Three kinds of data which are sent to each user (a), each group of users (b), and all users, are displayed.

Fig. 16 Extended function of Function  $\beta$ 

### 4.3 Characteristics of DACS Web Service

#### (1) Unifying Access Control

To manage access control information that is stored in one place, a network administrator can unify the access control for multiple Web servers distributed on the local area network. In a conventional network, access control for Web servers is performed by the mechanisms for access control that are distributed on the local area network, such as packet filtering of the fire-wall and router as well as access control of the Web server.

#### (2) Detailed Access Control

To change access control information that is stored in one place, a network administrator can manage the access control for the Web server according to the users and the kind of network (e.g., for office work, for students, public network). To be more specific, when a user moves multiple networks with a personal notebook computer and is permitted access to one Web server, the network administrator can refuse the user's access depending on the network configuration. In a conventional network, the same access control can be performed functionally by arranging a mechanism that combines access control by user authentication with that by use of the source IP address. However, this is a complicated and troublesome method because a network administrator needs to manage multiple Web servers individually.

#### (3) Change of the Referred Web Server

Using the same URL, the referred Web server can be changed according to the user and the type of network. When the Web Server, which is used by presenting the necessary information for each network (e.g., the guidance information of the network use), and the URL for

information reference is decided as a promise, a user can refer to that information by automatically accessing the Web server located in that network.

#### 4.4 Personal Portal Creation by Using Prototype System

```

<html>
<head>
<title>Personal Portal</title>
</head>

<frameset cols="50%,50%">

<frameset rows="30%,30%,40%">
<frame src="https://www.test.ac.jp/userdataget.php" name="frame1"           ←(a)
<frame src="https://www.test.ac.jp/groupdataget.php" name="frame2"       ←(b)
<frame src="https://www.test.ac.jp/allaverageget.php" name="frame3"      ←(c)
</frameset>

<frameset rows="30%,30%,40%">
<frame src="https://www.test.ac.jp/index.html" name="frame4"           ←(d)
<frame src="https://www2.test.ac.jp/" name="frame5"                   ←(e)
<frame src="https://www.test.ac.jp/allusers/index.html" name="frame6"   ←(f)
</frameset>

</frameset>

</html>

```

Fig.17 HTML File as Personal Portal

Using this system, the experiments were performed. First, the content of the static HTML file as a personal portal (portal page) is described in Fig.17. This personal portal is divided into 6 frames. At each point from (a) to (f), each URLs for using 6 functions described in Fig.14 were set into each frame.

<p>Result of userA</p> <table border="1"> <thead> <tr> <th>English</th> <th>Mathematics</th> <th>Chemistry</th> <th>Physic</th> <th>Geography</th> <th>World history</th> </tr> </thead> <tbody> <tr> <td>80</td> <td>85</td> <td>85</td> <td>90</td> <td>70</td> <td>75</td> </tr> </tbody> </table>	English	Mathematics	Chemistry	Physic	Geography	World history	80	85	85	90	70	75	<p>Examination paper</p> <p><a href="#">English</a> <a href="#">Mathematics</a> <a href="#">Chemistry</a> <a href="#">Physic</a> <a href="#">Geography</a> <a href="#">World history</a></p>								
English	Mathematics	Chemistry	Physic	Geography	World history																
80	85	85	90	70	75																
<p>Average of groupA</p> <table border="1"> <thead> <tr> <th>English</th> <th>Mathematics</th> <th>Chemistry</th> <th>Physic</th> <th>Geography</th> <th>World history</th> </tr> </thead> <tbody> <tr> <td>65</td> <td>70</td> <td>60</td> <td>75</td> <td>55</td> <td>65</td> </tr> </tbody> </table>	English	Mathematics	Chemistry	Physic	Geography	World history	65	70	60	75	55	65	<p>Teaching materials</p> <p><a href="#">English</a> <a href="#">Mathematics</a> <a href="#">Chemistry</a> <a href="#">Physic</a> <a href="#">Geography</a> <a href="#">World history</a></p>								
English	Mathematics	Chemistry	Physic	Geography	World history																
65	70	60	75	55	65																
<p>Average of all subjects</p> <table border="1"> <thead> <tr> <th>English</th> <th>Mathematics</th> <th>Chemistry</th> <th>Physic</th> <th>Geography</th> </tr> </thead> <tbody> <tr> <td>60</td> <td>70</td> <td>60</td> <td>65</td> <td>60</td> </tr> <tr> <th>World history</th> <th>Japanese history</th> <th>philosophy</th> <th>art</th> <th>music</th> </tr> <tr> <td>60</td> <td>70</td> <td>75</td> <td>55</td> <td>65</td> </tr> </tbody> </table>	English	Mathematics	Chemistry	Physic	Geography	60	70	60	65	60	World history	Japanese history	philosophy	art	music	60	70	75	55	65	<p>Notice (All students)</p> <p><a href="#">Method of taking attendance</a></p> <p><a href="#">About examinations</a></p> <p><a href="#">Manner for classromm usage</a></p>
English	Mathematics	Chemistry	Physic	Geography																	
60	70	60	65	60																	
World history	Japanese history	philosophy	art	music																	
60	70	75	55	65																	

Fig.18 Portal Page for User A

Next, when the portal page was opened after user A logged in on a client machine, the page shown in Fig. 18 was displayed on the Web browser. The three frames on the left were as follows. Data stored in the database for user A, i.e., personal result extracted from the table which stored the results of all students were displayed in the top frame, which was realized with the Function  $\alpha$  used by the URL (1). Data stored in the database for group A of which user A is a member, i.e., average results of the classes that user A attended which were extracted

from the table with the average results for each class were displayed in the middle frame, which was realized with the Function  $\alpha$  used by the URL (2). Data stored in the database for all users, i.e., average results of all classes were displayed in the bottom frame, which was realized with the Function  $\alpha$  used by the URL (3). In addition, the right three frames on the right were as follows. The content of the static HTML file stored in the home directory for user A, i.e., marked examination papers of the attendance subjects were displayed in the top frame, which was realized with the Function  $\beta$  used by the URL (1). The content of the static HTML file stored in the home directory which is accessible by group A of which user A is a member, i.e., teaching materials of each attendance class were displayed in the middle frame. In this frame, URLs prepared for each subject are implemented, and each user is accessible to each home directory prepared for each class in each subject.

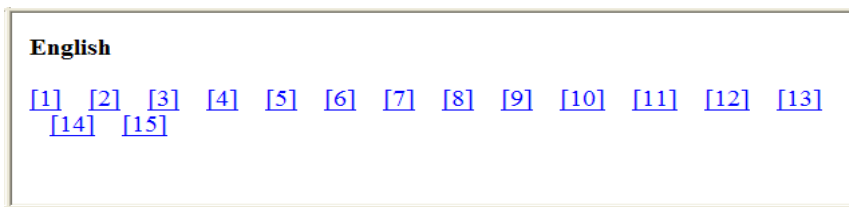


Fig. 19 Changed Frame

Therefore, user A became accessible to teaching materials of each attended class. In this case, when input the link of “English”, the content of the frame was changed to the page of the attended class with links of each materials in Fig.19. These were realized with the Function  $\beta$  used by the URL (2). The content of the static HTML file stored in home directory for all users, i.e., notices for all students were displayed in the bottom frame, which was realized with the Function  $\beta$  used by the URL (3). Then, when the HTML file serving as the personal portal was opened after user B logged in on a client machine, the data related to user B was displayed on each frame of the Web browser in the same way as the above-mentioned case (Fig. 20).

<p><b>Result of userB</b></p> <table border="1"> <tr> <td>English</td> <td>Mathematics</td> <td>Chemistry</td> <td>Philosophy</td> <td>art</td> <td>music</td> </tr> <tr> <td>75</td> <td>60</td> <td>80</td> <td>70</td> <td>90</td> <td>95</td> </tr> </table>	English	Mathematics	Chemistry	Philosophy	art	music	75	60	80	70	90	95	<p><b>Examination paper</b></p> <p><a href="#">English</a> <a href="#">Mathematics</a> <a href="#">Chemistry</a> <a href="#">Philosophy</a> <a href="#">art</a> <a href="#">music</a></p>								
English	Mathematics	Chemistry	Philosophy	art	music																
75	60	80	70	90	95																
<p><b>Average of groupB</b></p> <table border="1"> <tr> <td>English</td> <td>Mathematics</td> <td>Chemistry</td> <td>Philosophy</td> <td>art</td> <td>music</td> </tr> <tr> <td>70</td> <td>75</td> <td>65</td> <td>60</td> <td>75</td> <td>80</td> </tr> </table>	English	Mathematics	Chemistry	Philosophy	art	music	70	75	65	60	75	80	<p><b>Teaching materials</b></p> <p><a href="#">English</a> <a href="#">Mathematics</a> <a href="#">Chemistry</a> <a href="#">Philosophy</a> <a href="#">art</a> <a href="#">music</a></p>								
English	Mathematics	Chemistry	Philosophy	art	music																
70	75	65	60	75	80																
<p><b>Average of all subjects</b></p> <table border="1"> <tr> <td>English</td> <td>Mathematics</td> <td>Chemistry</td> <td>Physic</td> <td>Geography</td> </tr> <tr> <td>60</td> <td>70</td> <td>60</td> <td>65</td> <td>60</td> </tr> <tr> <td>World history</td> <td>Japanese history</td> <td>philosophy</td> <td>art</td> <td>music</td> </tr> <tr> <td>60</td> <td>70</td> <td>75</td> <td>55</td> <td>65</td> </tr> </table>	English	Mathematics	Chemistry	Physic	Geography	60	70	60	65	60	World history	Japanese history	philosophy	art	music	60	70	75	55	65	<p><b>Notice (All students)</b></p> <p><a href="#">Method of taking attendance</a></p> <p><a href="#">About examinations</a></p> <p><a href="#">Manner for classromm usage</a></p>
English	Mathematics	Chemistry	Physic	Geography																	
60	70	60	65	60																	
World history	Japanese history	philosophy	art	music																	
60	70	75	55	65																	

Fig. 20 Portal Page for User B

## 5. VIRTUAL USE METHOD OF THE CGI PROGRAM

### 5.1 Principle of Virtual Use method of the CGI program

In this paper, the method that is realized by the Function  $\alpha 1$  is proposed. By using this function, CGI program  $I$  is accessed virtually through same URL from users in each group. To be concrete, this method is realized by the following procedure.

#### (Step1) First setting of the CGI program

First, the CGI program is set by a normal procedure. For example, the program files as the CGI program are placed on the Web Server, and initial setting is performed. For example, the setting of initial parameter of the CGI program and permission of the program files. As the result, users can use the programs of the CGI program by inputting one URL into a Web Browser.

#### (Step2) Setting for virtual use of the CGI program programs

After copying the directory that stores the programs, it is pasted as another directory with another name. By repeating a similar operation, multiple directories for each group are prepared. At the same time, the content of the DACS rules is changed. As the result, users that belong to same group become possible to access the programs of same directory by use of a URL. On the other hand, users that belong to other group become impossible to access the above programs by use of same URL.

By these procedures, in the form of using same URL, users in each group can access the programs of the directory in each group, and can not access the programs of other group. Virtual use of the CGI program is realized without a special mechanism.

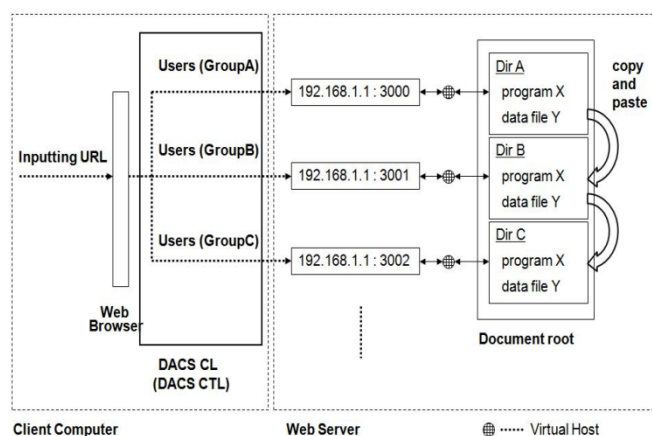


Fig. 21 Virtual Usage of the CGI Program

A concrete example of it is shown in Fig. 21. As first step, the program X as the CGI program and other files such as data file are placed in directory A (Dir A in Fig. 21), and initial setting of it is performed. As the result, users can access and use it. Next, second step is as follows. At first, Dir A is copied and pasted with another name. In Fig. 21, Dir B and Dir C are the pasted directories. Each directory is named with the regularity. Though each socket is connected to each directory through the virtual host by the system setting, each name is allocated to be easy to automate the setting. At the same time, by changing the DACS rules, the host name in URL and the communication port is converted to each socket every group. In Fig. 21, when users in Group A inputs one URL into a Web Browser, they access the program X in Dir A through by way of 192.168.1.1:3000. In the case of users in Group B, they access the program X in Dir B through by way of 192.168.1.1:3001. In the case of users in Group C, they access the program X in Dir C by way of 192.168.1.1:3002. Then, users in Group A can not access the program in Dir B and Dir C. Users in Group B can not access the program in Dir A and Dir C. Users in Group C can not access the program in Dir A and Dir B. In this way, virtual use of the CGI program is realized simply.

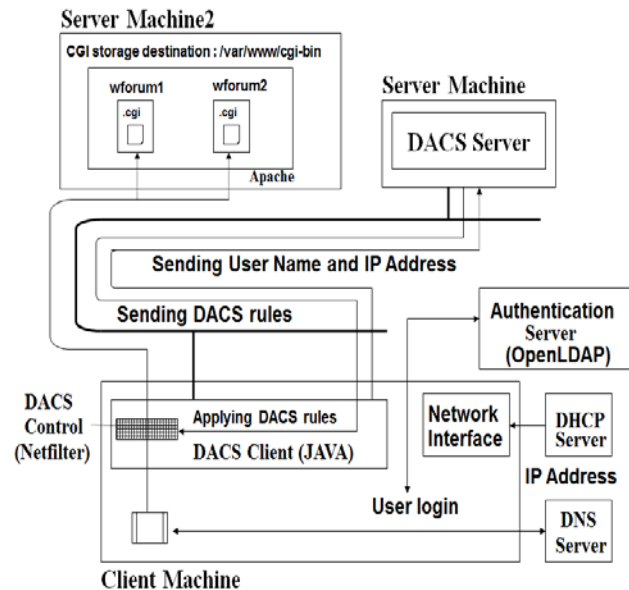


Fig. 22 System Configuration

## 5.2 Evaluations by experiments with the real CGI Program

Here, experimental results by using a bulletin board CGI program [17] open to Internet are shown. The CGI programs were operated on the experimental network systems described in Fig. 22, which were built as follows.

### (1) Server Machine

- CPU: Celeron M Processor340 (1.5GHz)
- OS: CentOS6
- DACS Server:

- Language: Visual C++ 7.1
- Database: postgresql-8.4.4-2

## **(2)Server Machine2**

- CPU: Celeron M Processor340 (1.5GHz)
- OS: CentOS 6
- Apache: httpd-2.2.15-5
- Database: postgresql8.4.4-2
- CGI Program on Apache: bulletin board
- Language: php-5.3.2-6

## **(3)Client Machine**

- CPU: Celeron M Processor340 (1.5GHz)
- OS: WindowsXP Professional
- DACS Client:
- Language: VisualC++ 7.1,WinsockSPI [18],Putty [19]

## **(4)Others**

- AuthenticationServer: openldap-2.1.22-8(FedoraCore1)
- DHCP Server: dhcp-4.1.1-12
- DNS Server: bind-9.2.2.P3-9(FedoraCore1)

In this experimental network system, same kinds of two CGI programs were set in the directories of “wforum1” and “wforum2” on the Server Machine 2, which had same program files and data files and log files. To be concrete, after setting for programs in the directory of “wforum1” was performed, the directory “wforum1” was copied and pasted as the directory “wforum2”. That is, same identical files were stored. After these setting process, setting for the Web Server was performed and the following setting were added into the configuration file (httpd/conf/httpd.conf).

### **(Additional setting)**

Listen 5001

Listen 5002

<VirtualHost 192.168.1.5:5001>

DocumentRoot “/var/www/cgi-bin/wforum1

</VirtualHost>

<VirtualHost 192.168.1.5:5002

DocumentRoot “/var/www/cgi-bin/wforum1

</VirtualHost>

Next, setting information as follows was set to DACS Server in Fig. 23. In this figure, “from\_ip” shows the IP address before destination IP address change, and “from\_port” shows the port number before destination port number change. Then, “to\_ip” shows the IP address after destination IP address change, and “to\_port” shows the port number after destination port number change.

username	from_ip	from_port	to_ip	to_port
user1	192.168.1.10	80	192.168.1.5	5001
user2	192.168.1.10	80	192.168.1.5	5002

Fig. 23 Setting information

After these settings were added, experiments were performed by using the experimental network system.

First, “user1” logged in to the client which was shown as “Client Machine” in Fig. 22, and input the URL (<http://dacsweb.ac.jp/wforum.cgi>) into a Web Browser.

At that time, the web page in Fig. 24 was displayed on the Web Browser. The system movements for displaying it were as follows.

**(Step1)** Inputting URL into a Web Browser

**(Step2)** Hostname was converted to IP Address (192. 168.1.10) by a name solution service of Domain Name Service (DNS).

**(Step3)** The Web Browser was trying to be communicated with the destination (192.168.1.10:80).

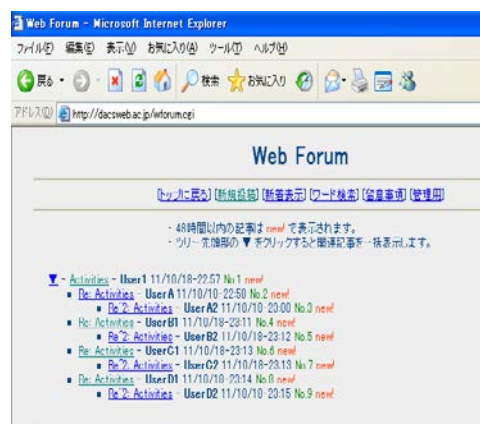


Fig. 24 Web page (1)

**(Step4)** The DACS Client changed the destination to 192.168.1.5:5001 form 192.168.1.10:80.

**(Step5)** The Web Server received the communication from the Web Browser, and the CGI program stored in /var/www/cgi-bin/wforum1 was accessed by the function of Virtual host function.



Next, “user2” logged in to the same client, and input the same URL into the Web Browser. The web page in Fig. 25 was displayed on the Web Browser. The system movements for displaying it were same in case of “user1”, and the CGI program stored in /var/www/cgi-bin/wforum2 was accessed by the function of Virtual host function with the same URL.

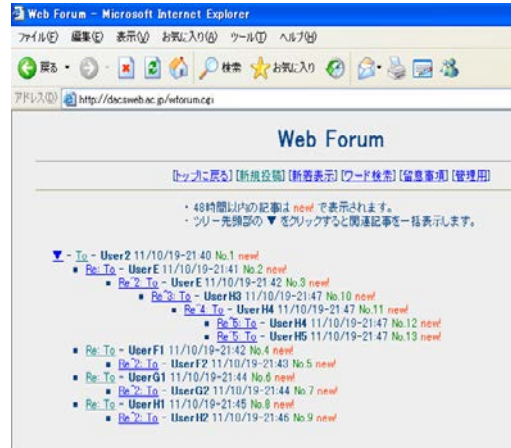


Fig. 25 Web page (2)

In the former experiments, by using the same URL, different users could access the same type of CGI programs that were set indifferent directories.

First, the additional function of user authentication mechanism is explained. Normally, when CGI programs with no user authentication mechanism are set on the Web Server, each CGI program is accessed from every user. However, on the experimental network system, only when each user could log in the client, the CGI program is accessed. Though every user can know the URL for accessing the CGI programs, the IP address which acquired from host name in the URL (IP address A) by DNS is not the IP address set as a virtual host(IP address B). By the function of the DACS Client, the destination IP address was changed from IP address A to IP address B. The rules for this destination change were acquired from the DACS Server at the timing of user authentication. In other words, the user authentication mechanism was used as the authentication mechanism for the CGI programs with no user authentication mechanism.

Next, the additional function of access control mechanism is explained. Normally, when CGI programs with no user authentication mechanism are set on the Web Server, each CGI program is accessed through each different URL. That is, each URL and each CGI program have one-to-one relationship. However, on the experimental network system, when each user logged in the client and input same URL, the different CGI program was accessed. In other words, each URL and each CGI program had one-to-many relationship. To be concrete, though “user1” could access the CGI programs in Fig. 24, “user2” could not the same CGI program by same URL. It is clear by “user2” accessing the CGI program in Fig. 25. In addition, though “user2” could access the CGI programs in Fig. 25, “user1” could not the same the CGI program by same URL. It is

clear by “user1” accessing the CGI program in Fig. 24. Thus, the access control mechanism was added to the CGI programs with no access control mechanism.

## 6. CONCLUSION

In this paper, contents of the virtual use method of the CGI program were shown, and evaluations of the virtual use method of the CGI program were performed by experiments with the real CGI program opened to Internet. As the result, it became possible for the CGI program to be used in the Web page as the personal portal of the real network. In the future, we are going to study about the new implementation method of CGI on the network introducing the DACS Scheme.

## REFERENCES

- [1]. J.Hartmann, Y.Sure, “An infrastructure for scalable, reliable semantic portals,” IEEE Intelligent Systems, Vol.19, no.3, pp.58-65, 2004.
- [2]. N.Lowe, A.Datta, “A New Technique for Rendering Complex Portals,” IEEE Tran. on Visualization and Computer Graphics, Vol.11, no.1, 2005.
- [3]. D.Robinson, “The WWW Common Gateway Interface Version 1.1, Internet Draft,” 1995.
- [4]. <http://www.netvibes.com/ja-jp>
- [5]. <http://www.google.co.jp/ig?hl=ja>
- [6]. K.Odagiri, R.Yaegashi, M.Tadauchi, N.Ishii, “Efficient Network Management System with DACS Scheme : Management with communication control,” International Journal of Computer Science and Network Security, Vol.6, no.1, pp.30-36, 2006.
- [7]. K.Odagiri, R.Yaegashi, M.Tadauchi, N.Ishii, “Efficient Network Management System with DACS Scheme,” in Proc. of International Conference on Networking and Services (ICNS), 16-19 July, Silicon Valley, USA, IEEE Computer Society, 2006.
- [8]. K.Odagiri, R.Yaegashi, M.Tadauchi, N.Ishii, “New Network Management Scheme with Client 's Communication Control” Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, Springer, Vol.4252, pp.379-386, 2006.
- [9]. K.Odagiri, R.Yaegashi, M. Tadauchi, N. Ishii, “Secure DACS Scheme,” Journal of Network and Computer Applications, Elsevier, Vol.31, Issue 4, pp.851-861, November, 2008
- [10]. K.Odagiri, R.Yaegashi, M.Tadauchi, N.Ishii, “New Web Service Based on Extended DACS Scheme,” International Journal of Computer Science and Network Security, Vol.6, no.3, pp.8-13, March, 2006.

- [11]. K.Odagiri, R.Yaegashi, M.Tadauchi, N.Ishii,"Free Information Usage System on the Network Introducing DACS Scheme" in Proc of International Conference on Internet and Web Applications and Services (ICIW), Mauritius, IEEE Computer Society, May, 2007.
- [12]. K.Odagiri, R.Yaegashi, M.Tadauchi, N.Ishii,"Practical DACS Web Service for User's Free Portal Creation," in Proc. of International Conference on Web Services (ICWS), Salt Lake City, UT, USA, pp.952-959, July, 2007.
- [13]. C.Bouras, V.Kapoulas, I.Misedakis,"Web Page Fragmentation for Personalized Portal Construction," in Proc. of the International Conference on Information Technology: Coding and Computing(ITCC'04), Las Vegas, Nevada, USA, IEEE Computer Society,2004.
- [14]. Ge Kaikai, Zhao Wenbin, Zhang Xintong,"Service-oriented personalized portal platform with QoS guarantee," International Conference on Advanced Computer Theory and Engineering, Vol.6, pp.205-209, August,2010.
- [15]. M.Vering, et al., "The E-Business Workplace: Discovering the Power of Enterprise Portals", John Wiley & Sons, February,2001.
- [16]. N.Hanakawa, Y.Akazawa, A.Mori, T.Maeda, S Inoue, S.Tsutsui,,"A Web-Based Integrated Education System for a Seamless Environment among Teachers, Students, and Administrators, "IEICE Transactions on Information and Systems, Pt.1 Vol.J88-D, no.2, pp.498-507,2005.
- [17]. <http://www.blackboard.com/>
- [18]. M. P. Papazoglou, D. Georgakopoulos, "Service Oriented Computing, " In Communications of the ACM, Vol.46, No.10, pp.25-28, October 2003.
- [19]. M. Endrei: Service-Oriented Architecture and Web Service, IBM, SG24-6303-00(2004)
- [20]. The World Wide Web Consortium (W3C), "Simple Object Access Protocol (SOAP) (version 1.1)", W3C Note 08, May, 2000.
- [21]. S.Woodman, G.Morgan, S.Parkin, "Portal Replication for Web Application Availability Via SOAP," in Proc. of the 8th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems, pp. 122-130, January, 2003.
- [22]. SOAP Specifications, W3C Recommendation. <http://www.w3.org/TR/soap/>
- [23]. Roy T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," University of California, 2000.
- [24]. R.Fielding, R.Taylor,"Principled Design of the Modern Web Architecture" in ACM Transaction on Internet Technology, Vol.2, No.2, pp.115-150, May, 2002.
- [25]. Roy T. Fielding and Richard N, "Taylor, Principled design of the modern Web architecture," ACM Transactions on Internet Technology, Vol.2, No.2, pp.115-150, May, 2002.
- [26]. W3C Recommendation 16, "Extensible Markup Language (XML) 1.1 (Second Edition)", <http://www.w3.org/TR/xml11/>, August, 2006.

- [27]. Lawrence A. Cunningham. "Language, Deals and Standards: The Future of XML Contracts," Boston College Law School Faculty papers, Paper 138,2006.
- [28]. D.F.Ferraiolo, D.R.Kuhn,"Role Based Access Control," 15th National Computer Security Conference, pp.554-563, October, 1992.
- [29]. R.S.Sandhu, E.J.Coyne, H.L.Feinstein and C.E.Youman,"Role-Based Access Control Models," IEEE Computer, Vol.29, No.2, pp.38-47. IEEE Press, February, 1996.
- [30]. R.Sandhu, D.F.Ferraiolo and D.R.Kuhn,"The NIST Model for Role Based Access Control: Toward a Unified Standard," 5th ACM Workshop Role-Based Access Control, pp.47-63, July, 2000.
- [31]. C.Metz, "The latest in virtual private networks: part I," IEEE Internet Computing, vol.7, No.1, pp.87-91, 2003.
- [32]. C.Metz, "The latest in VPNs: part II," IEEE Internet Computing, Vol.8, No.3, pp.60-65, 2004.
- [33]. D.Wood, V.Stoss, L.Chan-Lizardo, G.S.Papacostas, M.E.Stinson, "Virtual private networks," International Conference on Private Switching Systems and Networks, pp.132-136, Jun, 1998.
- [34]. J.Mizusawa, N.Shigematsu, H.Itoh,"Virtual private network control system concept," International Conference on Private Switching Systems and Networks, pp.137-141, Jun, 1988.
- [35]. SSH : The Secure Shell Connection Protocol, RFC 4254, 2006.
- [36]. Winsock SPI: <http://www.microsoft.com/msj/0599/layeredservice/layeredservice.aspx>
- [37]. Putty: <http://www.chiark.greenend.org.uk/~sgtatham/putty/>