Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2375676 A1 2000/12/21

(21) 2 375 676

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2000/06/14
(87) Date publication PCT/PCT Publication Date: 2000/12/21
(85) Entrée phase nationale/National Entry: 2001/11/28
(86) N° demande PCT/PCT Application No.: GB 2000/002311
(87) N° publication PCT/PCT Publication No.: 2000/077661
(30) Priorité/Priority: 1999/06/16 (11/170197) JP

(51) Cl.Int.$^{7}$/Int.Cl.$^{7}$ G06F 17/00

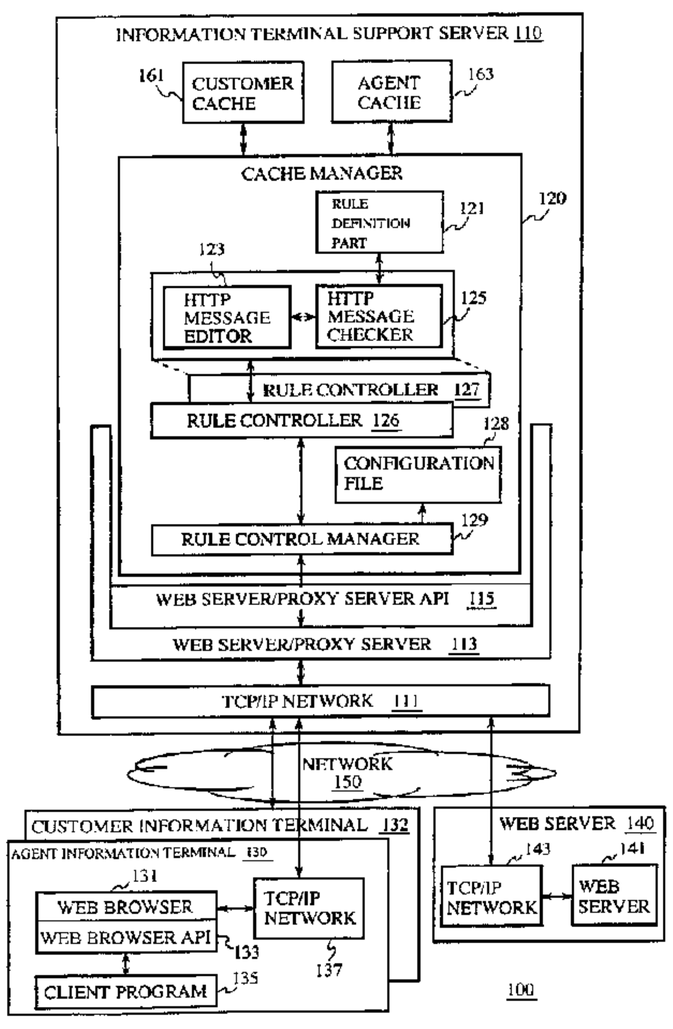
(71) Demandeur/Applicant:
INTERNATIONAL BUSINESS MACHINES CORPORATION, US

(72) Inventeurs/Inventors:
YOSHIDA, YOICHI, JP;
OHKADO, AKIRA, JP;
KAWASE, SATOSHI, JP

(74) Agent: ROSEN, ARNOLD

(54) Titre : PROCEDE DE TRAITEMENT D'INFORMATIONS, SERVEUR SUPPORT DE TERMINAL, SYSTEME DE COLLABORATION ET SUPPORT DE STOCKAGE D'UN PROGRAMME DE TRAITEMENT D'INFORMATIONS

(54) Title: INFORMATION PROCESSING METHOD, INFORMATION TERMINAL SUPPORT SERVER, COLLABORATION SYSTEM, AND STORAGE MEDIUM STORING AN INFORMATION PROCESSING PROGRAM




(57) Abrégé/Abstract:
Provided are methods, software and systems to change the behavior of a collaboration client according to context while collaborating on an internet. An HTTP request from a customer-side browser is sent to a web server (140) via an information terminal support server (110). The web server (140) then sends context in response to the request to the customer-side web browser (131) via the information terminal support server (110). At this time, the information terminal support server (110) checks the contents of the context, and when it is determined that there is agreement with a predetermined condition, a command is inserted into the context, context without the command inserted is sent to the customer-side web browser (131), and context with the command inserted is sent to the agent-side web browser (131). A client program (135) of the agent-side information terminal (130) then reads the embedded command using an API possessed by a web browser and carries out processing in order to prohibit form submission and form alteration and conceal specific forms or fields in accordance with the contents of this command.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 December 2000 (21.12.2000)

PCT

(10) International Publication Number
WO 00/77661 A2

(51) International Patent Classification[7]: G06F 17/00

(21) International Application Number: PCT/GB00/02311

(22) International Filing Date: 14 June 2000 (14.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/170197 16 June 1999 (16.06.1999) JP

(71) Applicants: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York, NY 10504 (US). **IBM UNITED KINGDOM LIMITED** [GB/GB]; P.O. Box 41, North Harbour, Portsmouth, Hampshire PO6 3AU (GB).

(72) Inventors: **YOSHIDA, Yoichi**; 1433-43, Honmachida, Machida-shi, Tokyo 194-0032 (JP). **OHKADO, Akira**; 4-3-1, 8-1402, Ryokuen, Izumi-ku, Yokohama-shi, Kanagawa (JP). **KAWASE, Satoshi**; 311-1-206, Kanamori, Machida-shi, Tokyo (JP).

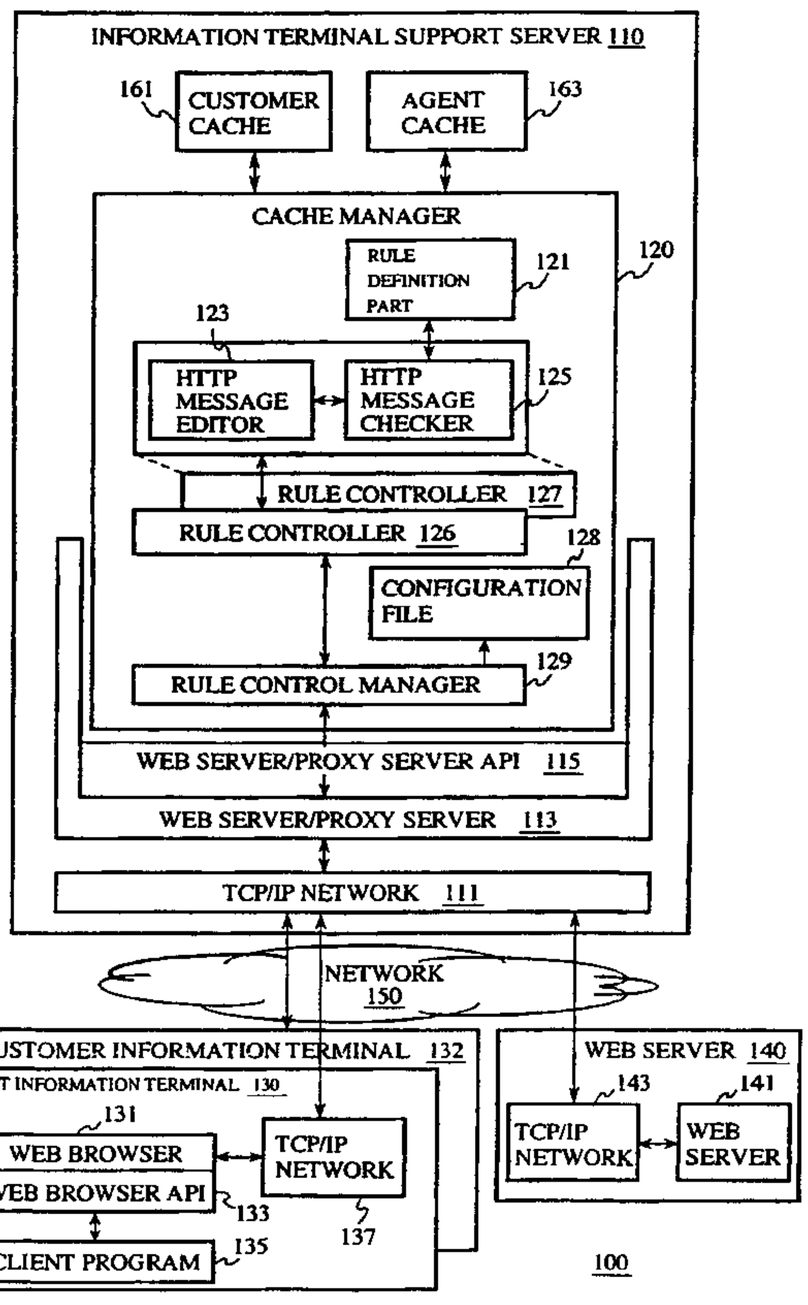
(74) Agent: **JENNINGS, Michael, John**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester, Hampshire SO21 2JN (GB).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,

*[Continued on next page]*

(54) Title: INFORMATION PROCESSING METHOD, INFORMATION TERMINAL SUPPORT SERVER, COLLABORATION SYSTEM, AND STORAGE MEDIUM STORING AN INFORMATION PROCESSING PROGRAM




(57) Abstract: Provided are methods, software and systems to change the behavior of a collaboration client according to context while collaborating on an internet. An HTTP request from a customer-side browser is sent to a web server (140) via an information terminal support server (110). The web server (140) then sends context in response to the request to the customer-side web browser (131) via the information terminal support server (110). At this time, the information terminal support server (110) checks the contents of the context, and when it is determined that there is agreement with a predetermined condition, a command is inserted into the context, context without the command inserted is sent to the customer-side web browser (131), and context with the command inserted is sent to the agent-side web browser (131). A client program (135) of the agent-side information terminal (130) then reads the embedded command using an API possessed by a web browser and carries out processing in order to prohibit form submission and form alteration and conceal specific forms or fields in accordance with the contents of this command.

WO 00/77661 A2

# INFORMATION PROCESSING METHOD, INFORMATION TERMINAL SUPPORT SERVER, COLLABORATION SYSTEM, AND STORAGE MEDIUM STORING AN INFORMATION PROCESSING PROGRAM

## Field of the invention

The present invention relates to an information processing method, and more particularly relates to a method for editing content sent from a server in response to a request from a client according to the intentions of an administrator.

## Background

At the time of applying for the present invention, there is provided, as in Japanese Patent Laid-open Publication Hei. 10-124461, collaboration technology where cooperative work (reading, moving, and changing) of an HTML page can be carried out simultaneously by a number of users. FIG. 5 shows an example of this HTML page cooperative work applied to an internet banking system. A bank agent and an internet customer consultation center agent etc. can then refer to and operate on the same page as the customer by employing this technology.

With specific HTML documents such as, for example, a screen for ordering a product, or a screen for designating a destination for a money transfer, there are situations where an agent should not be able to submit an HTML form or change data inputted to an HTML form. Collaboration servers were, however, originally provided in order to distribute the same HTML content between collaborating browsers. There is therefore no way of collaborating clients knowing when and how such behavior has changed.

Summary of Invention

In a first aspect, the present invention provides a collaboration system capable of changing collaborating client behavior according to context during internet based collaboration.

The present invention preferably provides a system where an administrator can intentionally change the content requested by a client.

The present invention preferably provides a low-cost collaboration system where resources required while supporting an information terminal are reduced.

The present invention preferably provides a collaboration system where centralized control is possible during information terminal support.

The present invention preferably provides a collaboration system that is not dependent on the platform of the information terminal operated.

The present invention preferably provides a collaboration control system capable of controlling the contents of a service provided to an information terminal without changing a server responding to a request from the information terminal.

According to one embodiment of the present invention, an HTTP request from a customer-side browser is sent to a web server via an information terminal support server. The web server then sends context

in response to the request to the customer-side web browser via the information terminal support server. At this time, the information terminal support server checks the contents of the context, and when it is determined that there is agreement with a predetermined condition, a command is inserted into the context. Context without the command inserted is sent to the customer-side web browser, and context with the command inserted is sent to the agent-side web browser. The client program of the agent-side information terminal then reads the embedded command via an API possessed by a web browser and carries out processing in order to prohibit form submission and form alteration and conceal specific forms or fields in accordance with the contents of this command.

In one aspect of the present invention, there is provided an information processing method in an information processing system having an information terminal support server which supports collaboration of a browser loaded on a customer-side information terminal and a browser loaded on an agent-side information terminal, said method comprising the steps of:

(a) receiving, at said information terminal support server, an HTTP message sent in response to a customer-side browser request;

(b) determining whether or not said HTTP message agrees with a predetermined condition;

(c) editing contents of said HTTP message when said HTTP message agrees with said predetermined condition;

(d) sending the unedited HTTP message to said customer-side browser; and

(e) sending the edited HTTP message to said agent-side browser.

In the context of this application, "information terminal support server" is a concept that includes collaboration servers and proxy servers.

In a further aspect of the present invention there is provided an information processing method wherein, in said step (c), editing is carried out to insert a command for said client program loaded on said agent-side information terminal. In the context of this application, "command" is a concept that includes commands instructing processing that can be carried out by browser-specific APIs.

In another aspect of the present invention there is provided an information processing method wherein, said command is one of "form submit prohibition", "form alteration prohibition", "concealment of specific form" or "concealment of specific field".

In a further aspect of the present invention there is provided an information processing method in an information processing system having an information terminal support server which supports an information terminal connected to a content server and loaded with a client program, said method comprising the steps of:

(a) receiving, at said information terminal support server, a message sent from said content server responding to a request from said information terminal;

(b) determining whether or not said message agrees with a predetermined condition;

(c) editing said message to insert a command for said client program when said message agrees with said predetermined condition; and

(d) sending said edited message to said information terminal.

In the context of this application, "content server" refers to servers such as web servers etc. that send content to the client-side in response to requests from a client.

In a still further aspect of the present invention there is provided an information terminal support server which supports collaboration of a browser loaded on a customer-side information terminal and a browser loaded on an agent-side information terminal, said information support server comprising:

(a) a rule definition part including a condition setting part and a command setting part;

(b) a rule control manager for monitoring, at said information support server, HTTP messages sent in response to customer-side browser requests;

(c) an HTTP checker, determining whether or not said HTTP message agrees with a condition of said condition setting part of said rule definition part;

(d) an HTTP editor, editing contents of said HTTP message according to contents of said command setting part of said rule definition part, when said HTTP message agrees with a predetermined condition;

(e) customer cache storing the unedited HTTP message sent to said customer-side browser; and

(f) agent cache storing the edited HTTP message sent to said agent-side browser.

In another aspect of the present invention there is provided an information terminal support server, wherein said HTTP editor carries out editing to insert a command for said client program loaded on said agent-side information terminal.

In a further aspect of the present invention there is provided an information terminal support server, wherein said command is one of "form submit prohibition", "form alteration prohibition", "concealment of specific form" or "concealment of specific field".

In another aspect of the present invention there is provided an information terminal support server which supports an information terminal connected to a content server and loaded with a client program, said information terminal support server comprising:

(a) a message checker for determining whether or not a message sent from a content server in response to a request from said information terminal agrees with a predetermined condition; and

(b) a message editor, editing said message to insert a command instructing said client program when said message agrees with said predetermined condition.

In a still further aspect of the present invention there is provided a collaboration system supporting collaboration of a browser loaded on a customer-side information terminal and a browser loaded on an agent-side information terminal, said collaboration system comprising:

(a) an agent-side information terminal collaborating with said customer-side information terminal;

(b1) a rule definition part including a condition setting part and a command setting part;

(b2) a rule control manager for monitoring an HTTP message sent in response to customer-side browser requests;

(b3) an HTTP checker determining whether or not said HTTP message agrees with a condition of said condition setting part of said rule definition part;

(b4) an HTTP editor, editing contents of said HTTP message according to contents of said command setting part of said rule definition part, when said HTTP message agrees with a predetermined condition;

(b5) customer cache storing the unedited HTTP message sent to said customer-side browser; and

(b6) an information terminal support server having agent cache storing the edited HTTP message sent to said agent-side browser.

In another aspect of the present invention there is provided a recording medium storing an information processing program executed within a system having an information terminal support server supporting collaboration of a browser loaded on a customer-side information terminal and a browser loaded on an agent-side information terminal, said program comprising:

(a) program code instructing said information terminal support server to receive an HTTP message sent in response to a request of said customer-side browser;

(b) program code instructing said information terminal support server to determine whether or not said HTTP message agrees with a predetermined condition;

(c) program code instructing said information terminal support server to edit said HTTP message contents when said HTTP message agrees with said predetermined condition;

(d) program code instructing said information terminal support server to send the unedited HTTP message to said customer-side browser; and

(e) program code instructing said information terminal support server to send the edited HTTP message to said agent-side browser.

In a further aspect of the present invention there is provided a recording medium, wherein editing is carried out to insert a command into said client program loaded on said agent-side information terminal.

In a still further aspect of the present invention there is provided a recording medium, wherein said command is one of "form submit prohibition", "form alteration prohibition", "concealment of specific form" or "concealment of specific field".

In another aspect of the present invention there is provided a recording medium storing an information processing program executed within a system having an information terminal support server which supports an information terminal connected to a content server and loaded with a client program, said program comprising:

(a) program code instructing said information terminal support server to receive a message sent from said content server in response to a request from said information terminal;

(b) program code instructing said information terminal support server to determine whether or not said message agrees with a predetermined condition;

(c) program code instructing said information terminal support server to edit said message to insert a command providing instructions to said client program when said message agrees with said predetermined condition; and

(d) program code instructing said information terminal server of sending of said edited message to said information terminal.

## Brief Description of the Drawings

FIG. 1 is a block diagram showing an embodiment of an information terminal support server or information terminal hardware configuration of the present invention;

FIG. 2 is a block diagram of the processing elements of the preferred embodiment of the present invention;

FIG. 3 is a flowchart showing the operating procedure for the information terminal support system of the preferred embodiment of the present invention;

FIG. 4 is a conceptual view of the configuration file of the preferred embodiment of the present invention; and

FIG. 5 is a view illustrating related collaboration technology.

## Detailed Description of Preferred Embodiments

### A. Hardware Configuration

FIG. 1 is an outline view showing an embodiment of a hardware configuration for an information terminal support server 110 used in the present invention. The information terminal support server 110 includes a Central Processing Unit (CPU) 1 and a memory 4. The CPU 1 and the memory 4 are connected to an auxiliary storage device that is a hard disc 13 via a bus 2. A floppy disc device (or a recording medium driver 26, 28 or 29 such as an MO or CD-ROM, etc.) 20 is connected to the bus 2 via a floppy disc controller 19 (or IDE controller 25 or SCSI controller 27 etc.).

A floppy disc (or recording medium such as an MO or CD-ROM) is inserted into the floppy disc device (or recording medium driver such as an MO or CD-ROM, etc.) 20. This floppy disc etc., hard disc device 13 or ROM 14 can then be loaded with code for a computer program, that provides instructions to the CPU etc. in unison with the operating system so as to execute the present invention. This program is loaded into the memory 4 for execution. The code for this computer program can be compressed or can be recorded distributed between a plurality of mediums.

The information terminal support server 110 is equipped with user interface hardware that can comprise a pointing device (mouse, joystick, etc.) 7 or a keyboard 6 for inputting data, and a display 12 for presenting the user with visual data. The input means can also be a touch panel. It is also possible to connect a printer via a parallel port 16 or connect a modem via a serial port 15. The information terminal support server 110 is connected to a network via the serial port 15 and the modem or a communications adapter card 18 (Ethernet or token ring card) etc. and is capable of communicating with other computers.

A speaker 23 receives an audio signal, D/A converted (Digital to Analog-converted) by an audio controller 21, via an amplifier 22 and outputs this signal as audio. An audio controller 21 A/D (Analog to Digital) converts audio information received from a microphone 24 so that audio information from outside the system can be taken in by the system.

It can be easily understood that the information terminal support server 110 can be an information terminal having a

communications function including one or a combination of a typical personal computer (PC), workstation, notebook PC, palmtop PC, or network computer, etc. Such elements of the configuration are, however, only shown as an example, and all of these elements of the configuration are not elements of the configuration essential to the present invention.

In particular, in the hardware configuration described here, the audio controller 21, amplifier 22, speaker 23 and microphone 24 required for processing audio, the keyboard 6, mouse 7 and keyboard/mouse controller 5 enabling direct input from an operator, the CRT 12, display device 11, VRAM 9, and VGA 8 for presenting visual data to the user, and each of the recording medium processors 19, 25 and 27, etc. are not essential to the support of the information terminal and may be omitted.

Various modifications such as combining each element of the configuration of the information terminal support server 110 over a number of machines and then executing these functions in a distributed manner can easily be assumed by one skilled in the art and such concepts are considered to be included in the spirit and scope of the present invention.

An information terminal 130 used in the present invention can also be realized, as with the information terminal support server 110, by the hardware configuration shown in FIG. 1, i.e. the information terminal 130 can also enable input of information requests and can issue and receive such requests. It can therefore be easily understood that the information terminal 130 can be realized by one or a combination of a typical personal computer (PC), a notebook PC,

palmtop PC, various household products such as a television with a computer built-in, game machines having communications functions, or information terminals having communications functions including a telephone, FAX, portable telephone, Personal Handy System (PHS) or electronic notebook, etc.. However, these elements of the configuration are shown as an example, and all of these elements of the configuration are by no means essential to the present invention.

The operating system on the side of the information terminal support server 110 is by no means limited to a specific operating system environment, and can be an operating system supporting a GUI multi-window environment as standard, such as Windows NT (trademark of Microsoft), Windows 9x (trademark of Microsoft), Windows 3.x (trademark of Microsoft), OS/2 (trademark of IBM), MacOS (trademark of Apple), Linux (trademark of Linus Torvalds), or the X-WINDOW system (trademark of MIT) on AIX (trademark of IBM), a character-based environment such as PC-DOS (trademark of IBM) or MS-DOS (trademark of Microsoft), a real time OS such as VxWorks (trademark of Wind River Systems, Inc.), or an operating system such as Java OS, etc., incorporated in a network computer.

The operating system on the side of the information terminal 130 is also by no means limited to a specific operating system environment, and can be an operating system supporting a GUI multi-window environment as standard, such as Windows NT (trademark of Microsoft), Windows 9x (trademark of Microsoft), Windows 3.x (trademark of Microsoft), OS/2 (trademark of IBM), MacOS (trademark of Apple), Linux (trademark of Linus Torvalds), or the X-WINDOW system (trademark of MIT) on AIX (trademark of IBM), a character-based environment such as PC-DOS (trademark of IBM) or MS-DOS (trademark of

Microsoft), a real time OS such as VxWorks (trademark of Wind River Systems, Inc.), or an operating system such as Java OS, etc., incorporated in a network computer.

B. System Configuration.

FIG. 2 is a functional block diagram showing a system configuration of a collaboration system including the information terminal support server 110 of the preferred embodiment of the present invention.

A collaboration system 100 of the preferred embodiment of the present invention includes information terminals 130 and 132, a information terminal support server 110 and a web server 140.

In the preferred embodiment of the present invention, a web browser 131 is installed on the information terminals 130 and 132. The web browser 131 designates a URL and sends a request to go to a predetermined web server. The web browser 131 then receives a response sent from the web server 140 for displaying on the display screen. A client program 135 is installed on the agent information terminal 130. The client program 135 reads HTML content loaded at the web browser 131 via a web browser API 133 and executes commands carried out by a cache manager on the proxy server via the web browser API 133 and an OS API.

The information terminal support server 110 of the preferred embodiment of the present invention is equipped with a web server/proxy server 113, a web server/proxy server API 115, and a cache manager 120.

The cache manager 120 caches HTML contents (HTTP responses) for HTTP requests sent by the web browsers 131. When the two web browsers are shared, the cache manager 120 controls returning of the same HTML data in accordance with HTTP requests sent in an interactive manner. In the preferred embodiment of the present invention, the cache manager 120 is written using the proxy server or web server/proxy server API 115 so that all HTTP messages can be monitored using the proxy server or web server 113. HTTP messages handled and monitored by a rule control manager 129 are sent to the rule control manager 129, which is entrusted with the processing of these HTTP messages.

The rule control manager 129 delivers HTTP messages sent from the web server/proxy server 113 to rule controllers 126 and 127 designated by a configuration file 128 and is entrusted with the processing of these HTTP messages. The rule control manager 129 can designate a plurality of configuration files 128 and can call the rule controllers 126 and 127 in a designated order.

The rule controllers 126 and 127 comprise a HTTP message checker 125 and a HTTP message editor 123. The HTTP message checker 125 refers to a rule definition part 121 and determines whether or not to edit an HTTP message. When it is determined to edit a message, this HTTP message is sent to the HTTP message editor 123. The HTTP message editor 123 then edits the HTTP message according to the contents of the rule definition part 121, stores the edited message in an agent cache 163, and the results of editing are returned to the rule control manager 129. When it is determined not to edit a message, this message is returned to the rule control manager 129 as is. The rule definition part 121 comprises a condition setting part and a command setting

part, and defines under what conditions and in what manner HTTP messages are to be edited.

Each functional block shown in FIG. 2 is described above but these functional blocks are logical functional blocks. This does not necessarily mean that these functional blocks each have to be individually realized by hardware and software, and the functional blocks may be realized by combinations of hardware and software or by shared hardware and software. It also goes without saying that all of the functional blocks shown in FIG. 2 are not essential elements of the configuration of the present invention.

C. Operating Procedure.

FIG. 3 is a flowchart showing the operation of the information processing terminal support system 100 of the preferred embodiment of the present invention. This procedure starts from step 400. At this time, the agent-side information terminal 130 and the web browser 131 of the client-side information terminal 132 are connected to the information terminal support server 110.

In the preferred embodiment of the present invention, when the web browser 131 of the customer information terminal 132 makes a request to the web server on the collaboration server for a specific URL, a customer-specific UAI (customer identification information) is generated on the information terminal support server 110 and this is set as a Cookie at the HTTP response header and sent to the customer-side web browser.

On the agent side, the collaboration software is started up, and the collaboration server can recognize that the web browser 131 having an agent-specific ID is connected from the registration of a user ID and password. A cookie command is then included in all HTTP request headers generated thereafter. As a result, UAI information uniquely identifying the customer-side browser and the presence of an agent can be generated. The customer and a corresponding agent are managed by an agent management table and are made to correspond to each other. This technology is well known and details thereof are therefore omitted.

In this initial condition, the web browser 131 of the customer information terminal 132 makes the URL designated by the user an HTTP message and sends this HTTP message to the information terminal support server 110 (step 401). The information terminal support server 110 receives this HTTP message (step 403), and transfers the HTTP message to the to a web server 140 designated by the HTTP message. The HTTP message is transferred to a web server designated in a configuration file in the case of a collaboration server.

The web server 140 then processes the received HTTP message (step 405), and a new HTTP message is generated and returned to the information terminal support server 110 (step 407).

The information terminal support server 110 stores the HTTP message received from the web server 140 in the customer cache 161, and delivers the HTTP message to the rule control manager 129, which is entrusted with processing the HTTP message (step 409). The rule control manager 129 is responsible for processing HTTP messages called and sent in the order requested by the rule controllers 126 and 127 designated by the configuration file 128 (step 411). A plurality of

rule controllers of the format shown in FIG. 4 can be designated in the configuration file.

The rule controllers 126 and 127 themselves are Shared Object (in Unix) or Dynamic Link Library (in Windows) data. In the preferred embodiment of the present invention, this programming interface is decided as follows.

[Table 1]

```
int CscPluginCInit (const char *pszConf)
This function is called when CacheManager is started via CscPlhdrInit
0.
argument
    const char                  : path of HTML plugin configuration file

return value
   0                            : OK
   4900-4999 : error
int CscPluginCConv ( const CsmConvParam param,
   const char   *pIn,
   const int    isize,
   char         **ppOut,
   int          *osize )
This function is called to convert an HTML content via CscPlhdrConv0.
argument
   const CsmConvParam           : information for conversion
   const char *pIn              : input content
   const int isize              : size of input content
   char    **ppOut              : output content
   int       *osize             : size of output content
```

return value

0 : OK

4900-4999 : error

void CsmPluginCFree ( char *p)

This function is called to free ppOut of CscPluginCConv() via CscPlhdrFree0.

argument

char * : ppOut of CscPluginCConv()

return value

None

int CscPluginCTerminate ()

This function is called when CacheManager is stopped via CscPlhdrdrTerminate ().

argument

None

return value

0 : OK

4900-4999 : error

The rule controllers 126 and 127 comprise the HTTP message checker 125 and the HTTP message editor 123, call the HTTP message check when a HTTP message is sent from the rule control manager 129, and determine whether or not to edit this HTTP message in accordance with the rule definition part 121 (step 413).

The rule definition part is designated using the following format. In this example, the rule definition part comprises a condition setting part and a command setting part. The rule definition part is set to URL = <url name> and the HTTP message checker 125 determines whether or not the HTTP message agrees with this condition.

In this example a command inhibiting a predetermined operation with respect to a predetermined URL is designated, but the HTTP message can by analyzed and a condition to determine at the HTTP checker 125 whether or not there is agreement with a predetermined condition (for example, the presence of a send button, an input field, or predetermined violent or obscene characters or images) can be set.

```
URL=<url name>;AgentActionProhibited=<command>[<command>...]
<command>=<submit |
   alterform |
   notshown:[formname=<formname>,
         fieldname=<fieldname>[,<fieldname>...]
>;
```

In this example, form submission, form alterations, and concealment of specific forms or fields are designated. When it is determined by the HTTP message checker 125 that an alteration is required, the data in the customer cache 161 is copied to the agent cache 163. The HTTP message editor 123 is then called, and editing is carried out according to the information contained in the rule definition part 121 (step 415).

In the preferred embodiment of the present invention, the HTTP message editor 123 embeds the following commands in the HTTP message in the agent cache 163.

```
<!--#CP_WebC_AgentActionProhibited=<command>[<command>...]-->
<command>=<submit |
     alterform |
notshown: [formname=<formname>;]
```

fieldname=<fieldname>[,<fieldname>...]

>;

The information terminal support server 110 then sends a reply to the HTTP message to the web browser 131 (step 417). At this time, the HTTP message in the customer cache 161 is sent to the web browser 131 of the customer information terminal 132 and the HTTP message in the agent cache 163 is sent to the web browser 131 of the agent information terminal 130.

The web browser 131 of the agent-side information terminal 130 displays the received HTTP message and notifies the client program 135 that this has been done (step 419). Specifically, the client program 135 acquires an end of displaying event for the web browser 131.

The client program 135 reads the embedded command using an API possessed by the web browser and then executes this command (step 421). In this example, the client program 135 instructs, using an API possessed by the web browser or the OS, prohibition of submission and form alteration, and concealment of specific forms and fields. In the preferred embodiment of the present invention, the following processing is carried out on the client side when the command is NOTSHOWN (non-displaying of predetermined information).

The field not displayed on the agent side is taken to be FieldA. When a customer inputs data in a certain field, the client program transfers the inputted data to the agent-side client as is, but when there is data inputted in FieldA, the same number of the symbols "*" are transferred rather than transferring the inputted data, to be stored at the agent side client with the symbols "*" in FieldA

remaining as they are. An extremely high standard of security can therefore be maintained with this processing.

D. Others

In the above, a description is given of an example of the present invention applied to collaboration technology. However, in the present invention an HTTP message sent from the web server 140 is analyzed, and when there is agreement with a predetermined condition, this message is edited and sent to the client side. This technology can therefore also be applied to fields other than collaboration, such as Internet education, where, for example, a correct answer field may be masked until there is agreement with a predetermined condition. Further, in the preferred embodiment of the present invention, an example of an internet/intranet is described but this technology can also be applied to communication systems employing protocols other than the Internet protocol. Still further, in the preferred embodiment of the present invention, context with a command inserted is sent to the agent-side web browser and this command is executed by the client program. However, it is also possible to send context with a command inserted to the customer-side web browser in order to inhibit the customer from carrying out predetermined operations and then have the client program execute this command, or have the command executed by both the customer and agent.

As described above, according to a preferred embodiment of the present invention, there is provided a collaboration system capable of changing the behavior of collaborating clients according to context during collaboration on an internet. In particular, there is provided a system

capable of changing content sent from a server in response to a request from a client according to the intentions of an administrator

**CLAIMS**

1. An information processing method in an information processing system having an information terminal support server which supports collaboration of a browser loaded on a customer-side information terminal and a browser loaded on an agent-side information terminal, said method comprising the steps of:

(a) receiving, at said information terminal support server, an HTTP message sent in response to a customer-side browser request;

(b) determining whether or not said HTTP message agrees with a predetermined condition;

(c) editing contents of said HTTP message when said HTTP message agrees with said predetermined condition;

(d) sending the unedited HTTP message to said customer-side browser; and

(e) sending the edited HTTP message to said agent-side browser.

2. The information processing method of claim 1, wherein, in said step (c), editing is carried out to insert a command for said client program loaded on said agent-side information terminal.

3. The information processing method of claim 2, wherein said command is one of "form submit prohibition", "form alteration prohibition", "concealment of specific form" or "concealment of specific field".

4. An information processing method in an information processing system having an information terminal support server which supports an information terminal connected to a content server and loaded with a client program, said method comprising the steps of:

(a) receiving, at said information terminal support server, a message sent from said content server responding to a request from said information terminal;

(b) determining whether or not said message agrees with a predetermined condition;

(c) editing said message to insert a command for said client program when said message agrees with said predetermined condition; and

(d) sending said edited message to said information terminal.

5. An information terminal support server which supports collaboration of a browser loaded on a customer-side information terminal and a browser loaded on an agent-side information terminal, said information support server comprising:

(a) a rule definition part including a condition setting part and a command setting part;

(b) a rule control manager for monitoring, at said information support server, HTTP messages sent in response to customer-side browser requests;

(c) an HTTP checker, determining whether or not said HTTP message agrees with a condition of said condition setting part of said rule definition part;

(d) an HTTP editor, editing contents of said HTTP message according to contents of said command setting part of said rule definition part, when said HTTP message agrees with a predetermined condition;

(e) customer cache storing the unedited HTTP message sent to said customer-side browser; and

(f) agent cache storing the edited HTTP message sent to said agent-side browser.

6. The information terminal support server of claim 5, wherein said HTTP editor carries out editing to insert a command for said client program loaded on said agent-side information terminal.

7. The information terminal support server of claim 6, wherein said command is one of "form submit prohibition", "form alteration prohibition", "concealment of specific form" or "concealment of specific field".

8. An information terminal support server which supports an information terminal connected to a content server and loaded with a client program, said information terminal support server comprising:

(a) a message checker for determining whether or not a message sent from a content server in response to a request from said information terminal agrees with a predetermined condition; and

(b) a message editor, editing said message to insert a command instructing said client program when said message agrees with said predetermined condition.

9. A collaboration system supporting collaboration of a browser loaded on a customer-side information terminal and a browser loaded on an agent-side information terminal, said collaboration system comprising:

(a) an agent-side information terminal collaborating with said customer-side information terminal;

(b1) a rule definition part including a condition setting part and a command setting part;

(b2) a rule control manager for monitoring an HTTP message sent in response to customer-side browser requests;

(b3) an HTTP checker determining whether or not said HTTP message agrees with a condition of said condition setting part of said rule definition part;

(b4) an HTTP editor, editing contents of said HTTP message according to contents of said command setting part of said rule definition part, when said HTTP message agrees with a predetermined condition;

(b5) customer cache storing the unedited HTTP message sent to said customer-side browser; and

(b6) an information terminal support server having agent cache storing the edited HTTP message sent to said agent-side browser.

10. An information processing program for execution within a system having an information terminal support server supporting collaboration of a browser loaded on a customer-side information terminal and a browser loaded on an agent-side information terminal, said program comprising:

(a) program code instructing said information terminal support server to receive an HTTP message sent in response to a request of said customer-side browser;

(b) program code instructing said information terminal support server to determine whether or not said HTTP message agrees with a predetermined condition;

(c) program code instructing said information terminal support server to edit said HTTP message contents when said HTTP message agrees with said predetermined condition;

(d) program code instructing said information terminal support server to send the unedited HTTP message to said customer-side browser; and

(e) program code instructing said information terminal support server to send the edited HTTP message to said agent-side browser.

11. A recording medium having recorded thereon a program according to claim 10.

12. The recording medium of claim 11, wherein said editing is carried out to insert a command into said client program loaded on said agent-side information terminal.

13. The recording medium of claim 12, wherein said command is one of "form submit prohibition", "form alteration prohibition", "concealment of specific form" or "concealment of specific field".

14. An information processing program for execution within a system having an information terminal support server which supports an information terminal connected to a content server and loaded with a client program, said program comprising:

(a) program code instructing said information terminal support server to receive a message sent from said content server in response to a request from said information terminal;

(b) program code instructing said information terminal support server to determine whether or not said message agrees with a predetermined condition;

(c) program code instructing said information terminal support server to edit said message to insert a command providing instructions to said client program when said message agrees with said predetermined condition; and

(d) program code instructing said information terminal server of sending of said edited message to said information terminal.

15. A recording medium having recorded thereon a program according to claim 14.

FIG.1

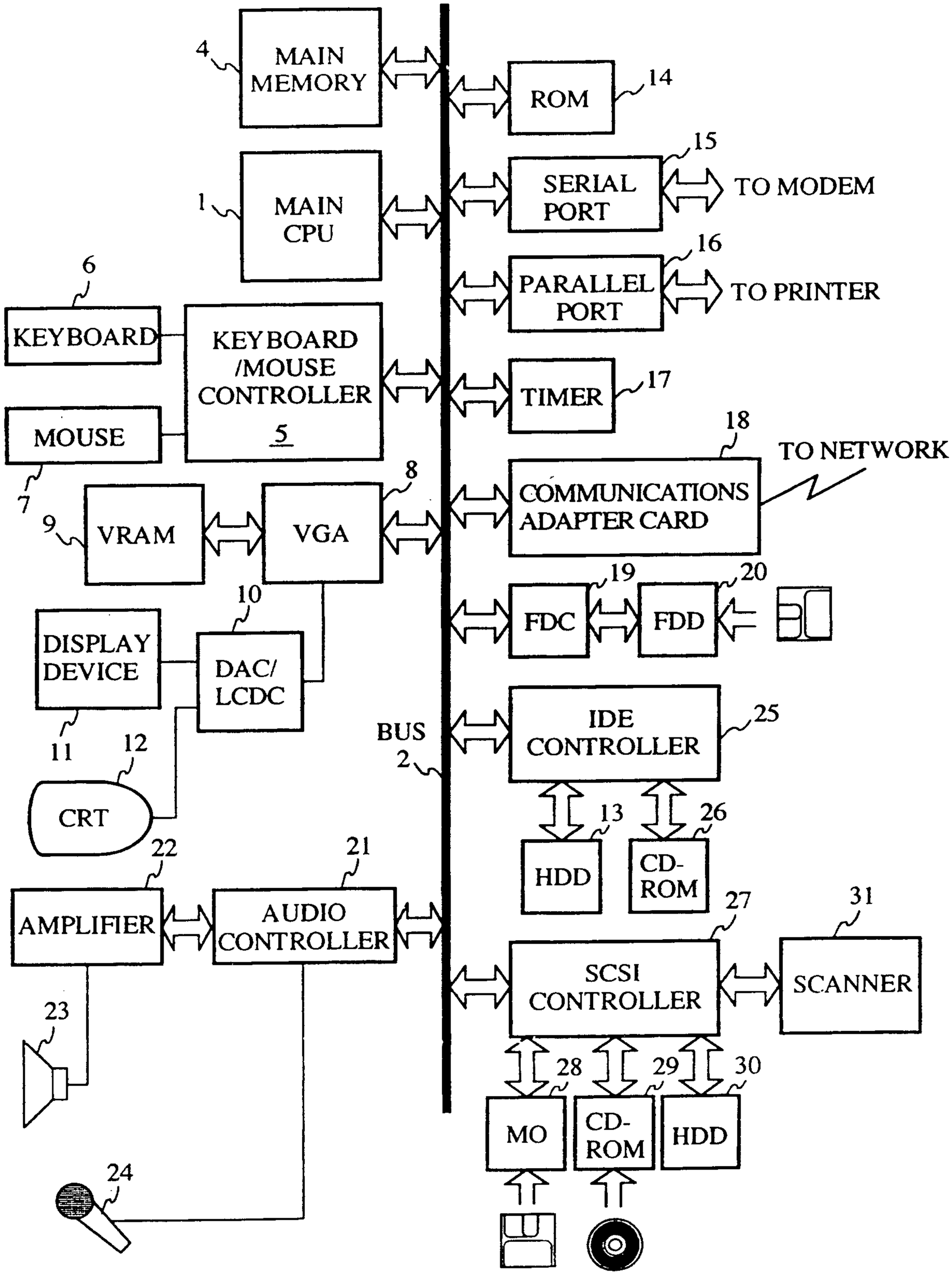

4
MAIN MEMORY
1
MAIN CPU
6
KEYBOARD
KEYBOARD /MOUSE CONTROLLER 5
MOUSE
7
9
VRAM
8
VGA
10
DISPLAY DEVICE
DAC/ LCDC
11
12
CRT
22
AMPLIFIER
21
AUDIO CONTROLLER
23
24
BUS 2
14
ROM
15
SERIAL PORT
TO MODEM
16
PARALLEL PORT
TO PRINTER
17
TIMER
18
COMMUNICATIONS ADAPTER CARD
TO NETWORK
19
FDC
20
FDD
25
IDE CONTROLLER
13
HDD
26
CD-ROM
27
SCSI CONTROLLER
31
SCANNER
28
MO
29
CD-ROM
30
HDD

FIG.2

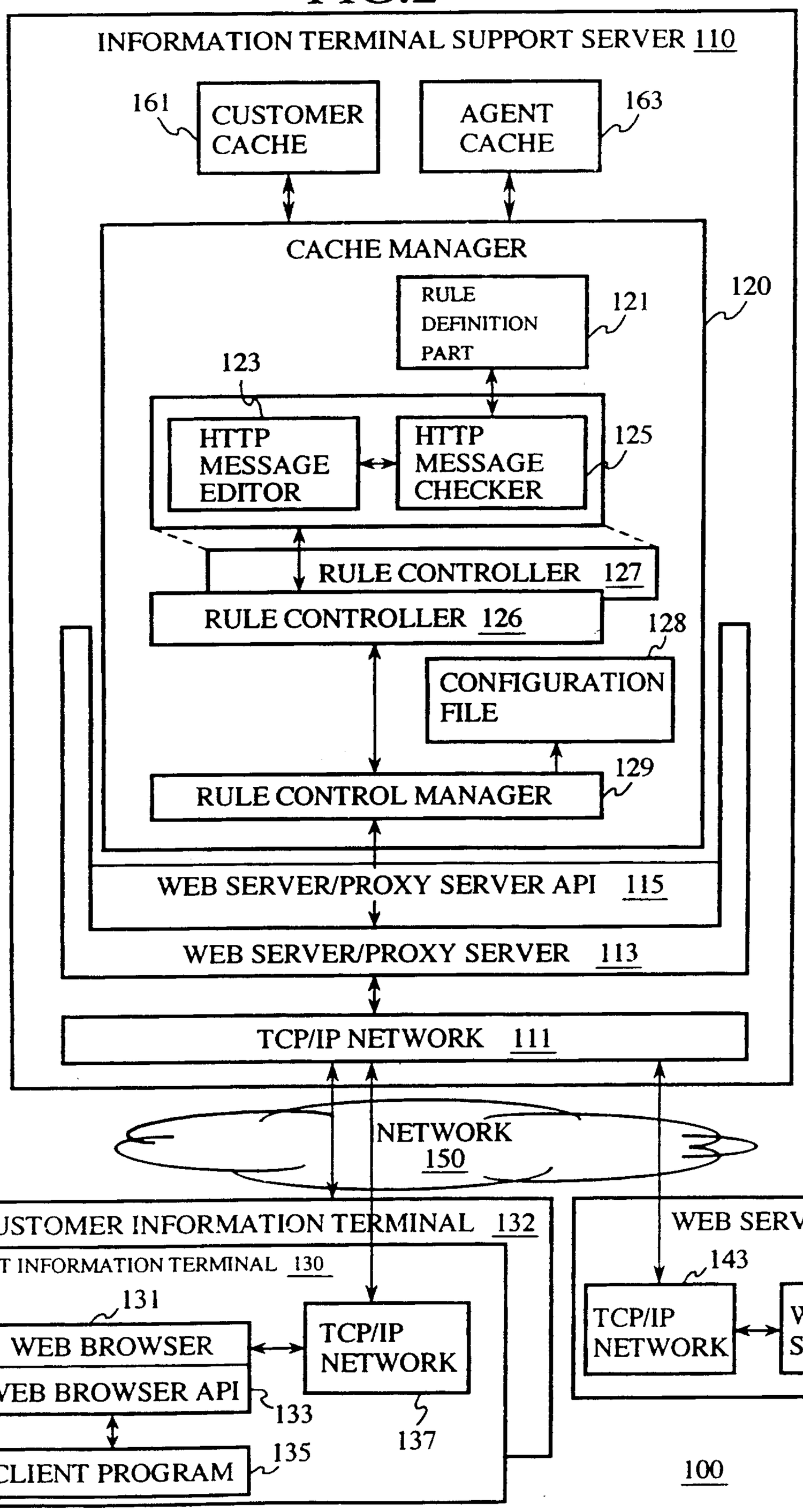

INFORMATION TERMINAL SUPPORT SERVER 110
161
CUSTOMER CACHE
AGENT CACHE
163
CACHE MANAGER
120
RULE DEFINITION PART
121
123
HTTP MESSAGE EDITOR
HTTP MESSAGE CHECKER
125
RULE CONTROLLER 127
RULE CONTROLLER 126
128
CONFIGURATION FILE
129
RULE CONTROL MANAGER
WEB SERVER/PROXY SERVER API 115
WEB SERVER/PROXY SERVER 113
TCP/IP NETWORK 111
NETWORK 150
CUSTOMER INFORMATION TERMINAL 132
AGENT INFORMATION TERMINAL 130
131
WEB BROWSER
WEB BROWSER API
133
TCP/IP NETWORK
137
CLIENT PROGRAM
135
WEB SERVER 140
143
141
TCP/IP NETWORK
WEB SERVER
100

FIG.3

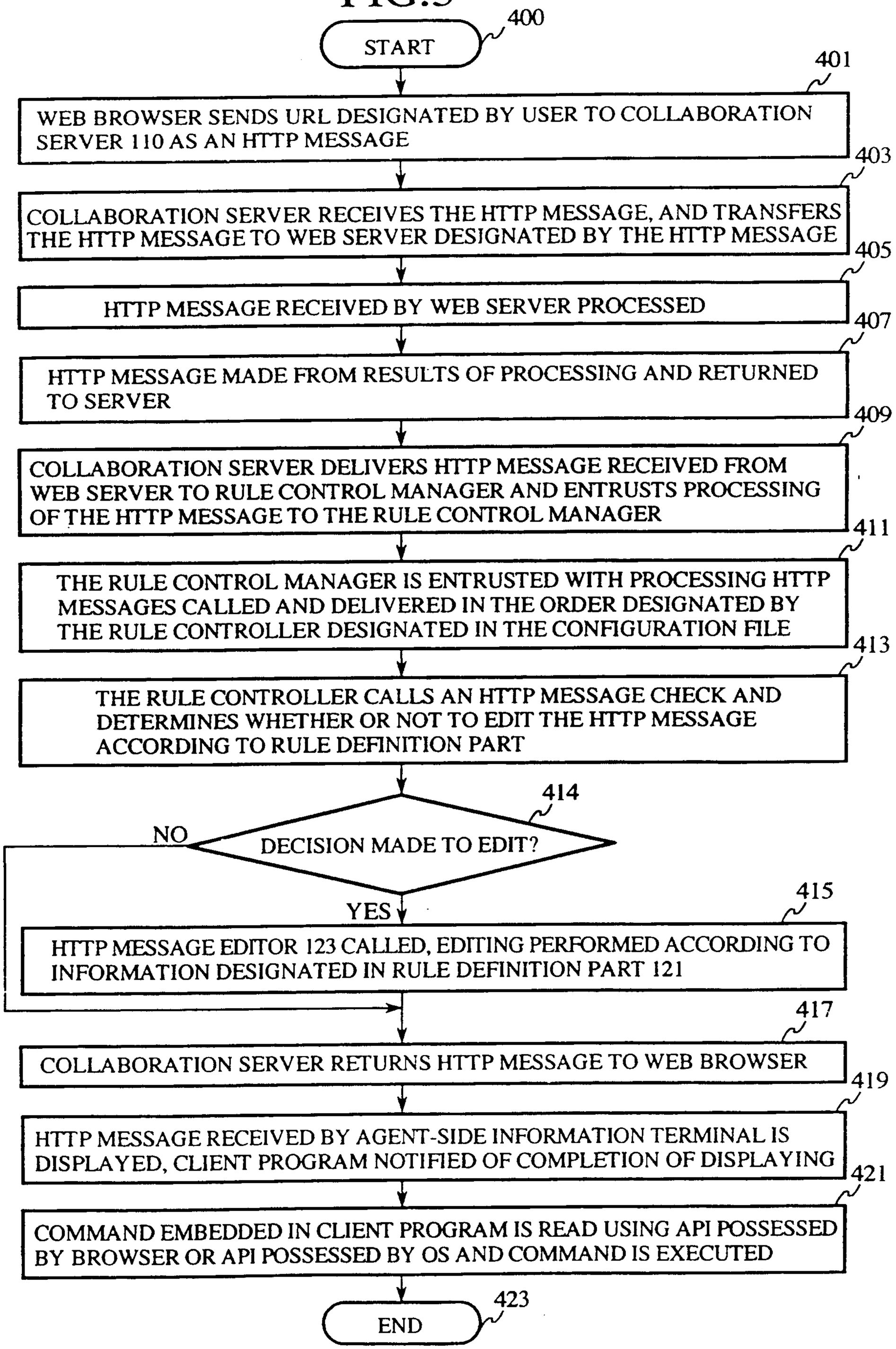
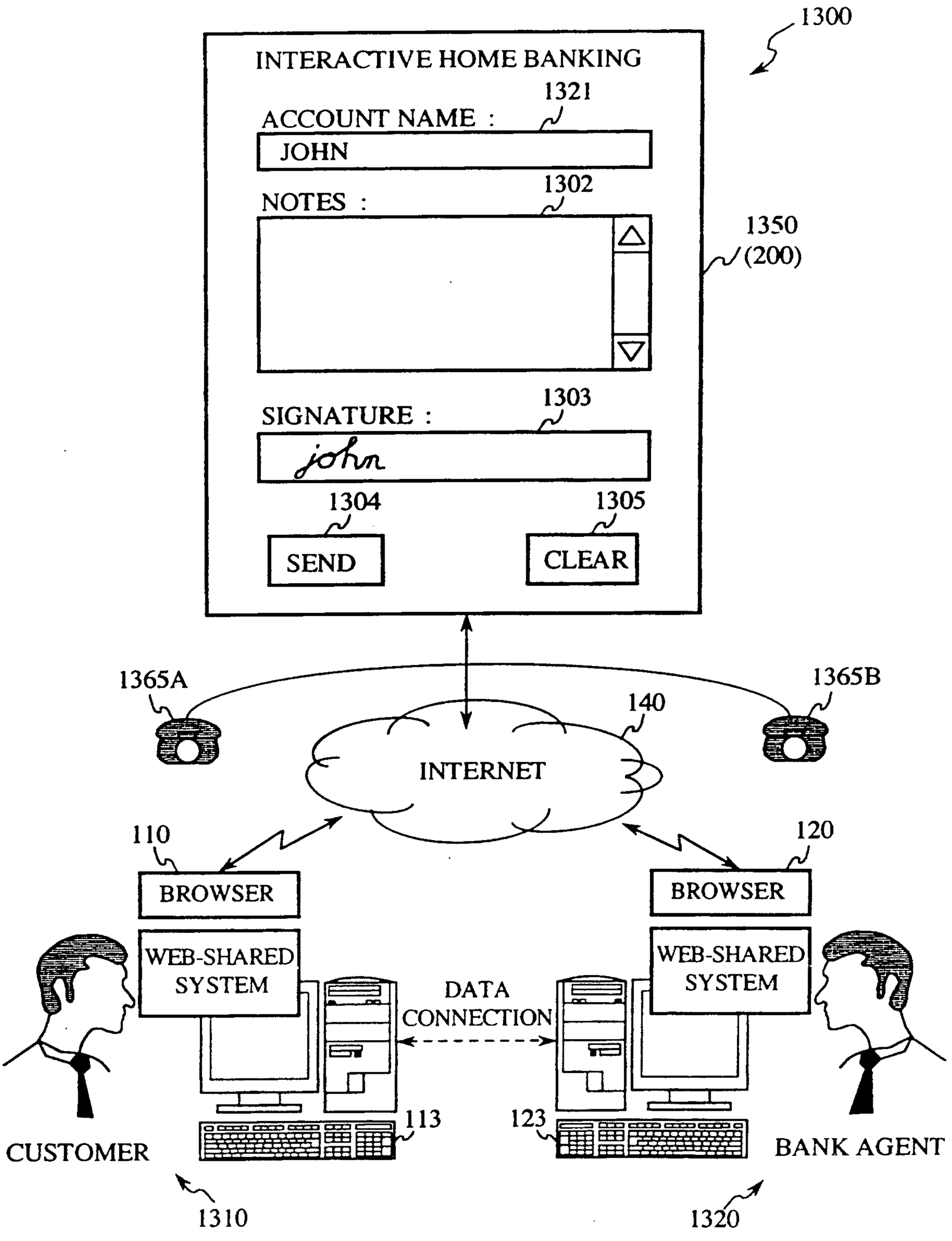
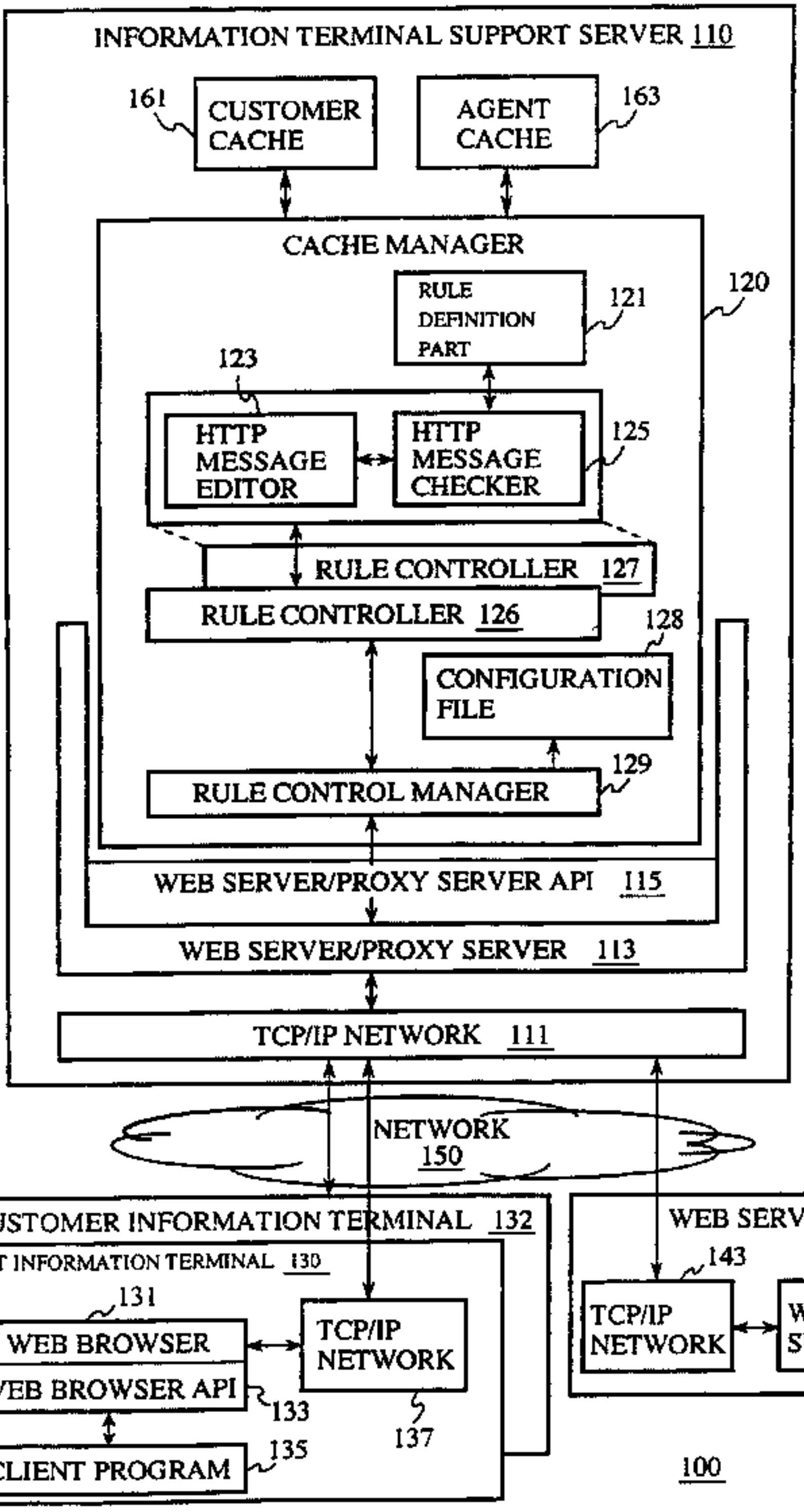

START
400
401
WEB BROWSER SENDS URL DESIGNATED BY USER TO COLLABORATION SERVER 110 AS AN HTTP MESSAGE
403
COLLABORATION SERVER RECEIVES THE HTTP MESSAGE, AND TRANSFERS THE HTTP MESSAGE TO WEB SERVER DESIGNATED BY THE HTTP MESSAGE
405
HTTP MESSAGE RECEIVED BY WEB SERVER PROCESSED
407
HTTP MESSAGE MADE FROM RESULTS OF PROCESSING AND RETURNED TO SERVER
409
COLLABORATION SERVER DELIVERS HTTP MESSAGE RECEIVED FROM WEB SERVER TO RULE CONTROL MANAGER AND ENTRUSTS PROCESSING OF THE HTTP MESSAGE TO THE RULE CONTROL MANAGER
411
THE RULE CONTROL MANAGER IS ENTRUSTED WITH PROCESSING HTTP MESSAGES CALLED AND DELIVERED IN THE ORDER DESIGNATED BY THE RULE CONTROLLER DESIGNATED IN THE CONFIGURATION FILE
413
THE RULE CONTROLLER CALLS AN HTTP MESSAGE CHECK AND DETERMINES WHETHER OR NOT TO EDIT THE HTTP MESSAGE ACCORDING TO RULE DEFINITION PART
414
DECISION MADE TO EDIT?
NO
YES
415
HTTP MESSAGE EDITOR 123 CALLED, EDITING PERFORMED ACCORDING TO INFORMATION DESIGNATED IN RULE DEFINITION PART 121
417
COLLABORATION SERVER RETURNS HTTP MESSAGE TO WEB BROWSER
419
HTTP MESSAGE RECEIVED BY AGENT-SIDE INFORMATION TERMINAL IS DISPLAYED, CLIENT PROGRAM NOTIFIED OF COMPLETION OF DISPLAYING
421
COMMAND EMBEDDED IN CLIENT PROGRAM IS READ USING API POSSESSED BY BROWSER OR API POSSESSED BY OS AND COMMAND IS EXECUTED
END
423

FIG.4

```
CscHTMLRuleHndlr=/usr/lpp/ibmcf/lib/csmplhdr.so
```

FIG.5




INFORMATION TERMINAL SUPPORT SERVER 110
161
CUSTOMER CACHE
AGENT CACHE
163
CACHE MANAGER
120
RULE DEFINITION PART
121
123
HTTP MESSAGE EDITOR
HTTP MESSAGE CHECKER
125
RULE CONTROLLER 127
RULE CONTROLLER 126
128
CONFIGURATION FILE
RULE CONTROL MANAGER
129
WEB SERVER/PROXY SERVER API 115
WEB SERVER/PROXY SERVER 113
TCP/IP NETWORK 111
NETWORK 150
CUSTOMER INFORMATION TERMINAL 132
AGENT INFORMATION TERMINAL 130
131
WEB BROWSER
WEB BROWSER API
133
CLIENT PROGRAM
135
TCP/IP NETWORK
137
WEB SERVER 140
143
141
TCP/IP NETWORK
WEB SERVER
100