



(51) International Patent Classification:

G06F 13/00 (2006.01) G06F 3/153 (2006.01)
G06F 3/12 (2006.01) H04N 1/00 (2006.01)

(21) International Application Number:

PCT/JP2011/062875

(22) International Filing Date:

31 May 2011 (31.05.2011)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

2010-130448 7 June 2010 (07.06.2010) JP
2011-112898 19 May 2011 (19.05.2011) JP

(71) Applicant (for all designated States except US): **CANON KABUSHIKI KAISHA** [JP/JP]; 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, 1468501 (JP).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **ITO, Fumitoshi** [JP/JP]; c/o CANON KABUSHIKI KAISHA, 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, 1468501 (JP).

(74) Agents: **OHTSUKA, Yasunori** et al.; 7th Fl., Kioicho Park Bldg., 3-6, Kioicho, Chiyoda-ku, Tokyo 1020094 (JP).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

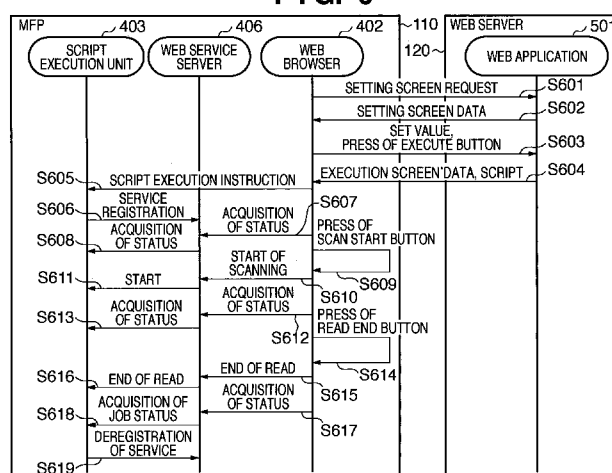
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: INFORMATION PROCESSING APPARATUS, INFORMATION PROCESSING APPARATUS CONTROL METHOD, AND PROGRAM

FIG. 6



(57) Abstract: There is provided a technique for enabling, as needed, a Web service for controlling execution of each function of an MFP when executing the function in response to a request from a Web server in an information processing apparatus. An information processing apparatus of the present invention includes a Web browser which transmits input information input through an displayed operation screen based on screen information provided from a Web server. Upon receiving, from the Web server, an execution request of processing corresponding to the input information, the information processing apparatus registers in a Web service server a service (Web service) defined in definition information contained in the execution request when starting execution of the processing according to the execution request. The web service server provides, among a plurality of providable services, the registered service which is used for controlling the execution of the processing.

- 1 -

DESCRIPTION

TITLE OF INVENTION

INFORMATION PROCESSING APPARATUS, INFORMATION
PROCESSING APPARATUS CONTROL METHOD, AND PROGRAM

TECHNICAL FIELD

[0001] The present invention relates to an information processing apparatus which can connect to a Web server and displays an operation screen based on screen information provided by the Web server, an information processing apparatus control method, and a program.

BACKGROUND ART

[0002] It is known to connect an information processing apparatus such as a PC to a Web server on a network, and to display, by the Web browser of the information processing apparatus, an operation screen provided by the Web server. In this case, the Web browser of the information processing apparatus first requests, of the Web server, screen information about an operation screen. Upon receiving the request from the information processing apparatus, a Web application in the Web server responds to the request by sending back, to the information processing apparatus, an HTML file corresponding to the screen information for causing the Web browser to display an operation screen.

The Web browser of the information processing apparatus parses the received HTML file, and displays the operation screen based on the description of the HTML file on a display unit such as a display.

[0003] If the user inputs an instruction via the operation screen displayed by the Web browser in the information processing apparatus, the Web browser notifies the Web server of the input instruction. Upon receiving the notification, the Web application in the Web server executes processing according to the received instruction.

[0004] Some recent Multi Function Peripherals (MFPs) having a scanner and a printer include the above-mentioned Web browser. The MFP displays an operation screen based on screen information provided by the Web server using the above procedures on the display unit through the Web browser, and accepts various instructions input through the displayed operation screen by the user.

[0005] According to Japanese Patent Laid-Open No. 2006-127503, for example, the Web server provides an MFP with screen information about an operation screen for inputting an instruction to use each function of the MFP. The user of the MFP inputs an instruction to the MFP using the displayed operation screen. The Web browser of the MFP notifies the Web server of the input instruction as a request. Upon receiving the

notification, the Web server generates a script describing various processes for the MFP based on the input instruction. The Web server notifies the MFP of the generated script as a response to the above-mentioned request from the Web browser. The MFP passes, from the Web browser to a script processing unit, the script sent from the Web server to execute the script, thereby using a function of the device of the MFP.

[0006] According to Japanese Patent Laid-Open No. 2008-003833, unlike Japanese Patent Laid-Open No. 2006-127503, the Web browser of an MFP executes a screen display script. The MFP calls a Web service for processing a function of the MFP within the script to use a function of the device of the MFP.

[0007] In the MFP described in Japanese Patent Laid-Open No. 2006-127503, software of the script processing unit displays, during execution of the script, an operation screen for inputting an instruction associated with the execution process. The MFP, therefore, cannot display a screen provided by the Web application of the Web server during execution of the script. Furthermore, it is difficult to customize the operation screen.

[0008] In the MFP described in Japanese Patent Laid-Open No. 2008-003833, in order for the Web application of the Web server to control the user interface (UI) of the MFP, the MFP must always enable

the Web server to use a Web service for causing the Web server to control a function of the MFP regardless of whether the script is in progress. Therefore, MFP resources such as a memory are needed even when the script is not in progress.

SUMMARY OF INVENTION

[0009] The present invention has been made in consideration of the above problems, and provides a technique which allows, as needed, to use a service for controlling execution of a function of an MFP when executing the function in the MFP in response to a request from a Web server.

[0010] According to one aspect of the present invention, there is provided a information processing apparatus connectable to a Web server, the information processing apparatus comprising: display control means including transmission means for displaying an operation screen based on screen information provided by the Web server, and transmitting input information input through the operation screen to the Web server, and reception means for receiving, from the Web server, an execution request of processing corresponding to the input information transmitted by the transmission means, the execution request containing definition information of a service associated with execution of the processing; execution control means for executing, upon

receiving the execution request from the Web server via the display control means, processing according to the execution request; and service providing means for providing, among a plurality of providable services for controlling the execution control means, a service which is registered by the execution control means, the execution control means including registration means for registering in the service providing means the service defined in the definition information contained in the execution request when starting execution of the processing according to the execution request.

[0011] According to another aspect of the present invention, there is provided a control method for controlling an information processing apparatus connectable to a Web server, the control method comprising: causing display control means of the information processing apparatus to display an operation screen based on screen information provided by the Web server, and transmit input information input through the operation screen to the Web server; causing the display control means to receive, from the Web server, an execution request of processing corresponding to the input information transmitted in the causing the display control means to display, the execution request containing definition information of a service associated with execution of the processing; causing execution control means of the information

processing apparatus to execute, upon receiving the execution request from the Web server via the display control means, processing according to the execution request; and causing service providing means of the information processing apparatus to provide, among a plurality of providable services for controlling the execution control means, a service which is registered by the execution control means, the causing the execution control means to execute including registering in the service providing means the service defined in the definition information contained in the execution request when starting execution of the processing according to the execution request.

[0012] According to the present invention, it is possible to provide a technique which allows, as needed, to use a service for controlling execution of a function of an MFP when executing the function in the MFP in response to a request from a Web server.

[0013] Further features of the present invention will become apparent from the following description of exemplary embodiments (with reference to the attached drawings).

BRIEF DESCRIPTION OF DRAWINGS

[0014] Fig. 1 is a view showing the overall configuration of an information processing system according to the first embodiment;

[0015] Fig. 2 is a block diagram showing the hardware configuration of an MFP according to the first embodiment;

[0016] Fig. 3 is a block diagram showing the hardware configuration of a Web server according to the first embodiment;

[0017] Fig. 4 is a block diagram showing the software configuration of the MFP according to the first embodiment;

[0018] Fig. 5 is a block diagram showing the software configuration of the Web server according to the first embodiment;

[0019] Fig. 6 is a chart showing a communication processing sequence according to the first embodiment;

[0020] Fig. 7 is a view showing an example of the user interface of a Web browser according to the first embodiment;

[0021] Fig. 8 is a view showing an example of a script file according to the first embodiment;

[0022] Figs. 9A and 9B are views showing an example of HTML data according to the first embodiment;

[0023] Fig. 10 is a flowchart illustrating a script execution procedure in the MFP according to the first embodiment;

[0024] Figs. 11A, 11B, and 11C are views each showing an example of the user interface of the Web browser in a scan execution state, standby state, or

end state according to the first embodiment;

[0025] Fig. 12 is a chart showing a communication processing sequence according to the second embodiment; and

[0026] Fig. 13 is a view showing an example of HTML data according to the second embodiment.

DESCRIPTION OF EMBODIMENTS

[0027]<First Embodiment>

In the first embodiment, a case in which an MFP as an example of an information processing apparatus according to the present invention executes image forming processing in response to an instruction from a Web server will be explained.

[0028] (Overall Configuration of Information Processing System)

Fig. 1 is a view showing the overall configuration of an information processing system according to the first embodiment. As shown in Fig. 1, an MFP 110 and Web server 120 are connected to a LAN 100 such as Ethernet®. The MFP 110 has a copy function and a data transmission function. The data transmission function reads a document image, and transmits the acquired image data (to be also referred to as document data) to a specified device on the LAN 100 using the File Transfer Protocol (FTP), the Server Message Block (SMB) protocol, and the like. The MFP

110 may also have a function of creating an e-mail message attached with image data as an attached file, and transmitting it using an e-mail server (not shown).

[0029] The server 120 serves as a Web server. In response to a request from the MFP 110, a Web application serving as software operating on the Web server generates screen information (HTML data) for displaying an operation screen on (the liquid crystal display unit of) the operation unit of the MFP 110, and transmits it to the MFP 110 as a response to the request.

[0030] The number of components of the information processing system in this embodiment is not limited to that shown in Fig. 1. Although a LAN is applied as a connection method in this embodiment, the present invention is not limited to this. For example, any network such as a WAN (public network), a serial transmission method such as USB, and a parallel transmission method such as the Centronics and SCSI are also applicable.

[0031] (Hardware Configuration of MFP)

The MFP 110 includes the following components, as shown in Fig. 2. The MFP 110 has a scanner unit 112 serving as an image input device, a printer unit 113 serving as an image output device, a controller 200 for controlling the MFP 110, and an operation unit 111.

[0032] The scanner unit 112 serves as a device for

reading an image on a document to generate image data. When documents are set on a document feeder and a document read instruction is given from the controller 200 to the scanner unit 112, the scanner unit 112 feeds the documents one by one from the document feeder, and reads respective documents. Note that a method of scanning a document by placing the document on a glass surface (not shown) and moving an exposure unit may be used as a document read method instead of an automatic feed method using the document feeder.

[0033] The printer unit 113 serves as an image forming device which forms an image on a paper sheet based on image data received from the controller 200. Note that in this embodiment, an electrophotographic method using a photosensitive drum or photosensitive belt, or an ink-jet method may be adopted as an image forming method used by the printer unit 113.

[0034] The controller 200 is electrically connected to the operation unit 111, scanner unit 112, and printer unit 113, while connecting to the LAN 100 via a network interface 206. That is, the controller unit 200 is also connected to other devices via the LAN 100. This enables the controller 200 to communicate with other devices (for example, the server 120) connected via the LAN 100 according to the HTTP protocol.

[0035] A CPU 201 centrally controls access to

various devices in connection with the controller 200 and access from the devices based on a control program stored in a ROM 202. In addition, the CPU 201 centrally controls various processes executed within the controller 200. This control processing includes execution of programs for implementing processing shown in a flowchart of Fig. 10 (to be described later).

[0036] The ROM 202 stores the boot program of the apparatus in addition to the above-described control program. A RAM 203 serves as a system work memory for operation of the CPU 201, and also functions as a memory for temporarily storing image data. The RAM 203 includes an area which holds contents stored as a backup or the like even after powering off the apparatus main body, and a volatile area which erases stored contents after power off.

[0037] An HDD 204 serves as a hard disk drive, which can store system software and image data. An operation unit I/F 205 serves as an interface for connecting a system bus 211 and the operation unit 111. The operation unit I/F 205 receives image data for displaying a screen on (the liquid crystal display unit of) the operation unit 111 from the system bus 211, and outputs it to the operation unit 111. The operation unit I/F 205 also outputs, to the system bus 211, information input through the operation unit 111.

[0038] The network I/F 206 is connected to the LAN

100 and the system bus 211 to input/output information. A scanner I/F 207 corrects, processes, and edits image data received from the scanner unit 112. The scanner I/F 207 has a function of determining the type of received image data, that is, a color document or monochrome document, or a text document or photographic document.

[0039] An image processing unit 208, for example, changes the orientation of image data and compresses/decompresses an image. The image processing unit 208 can also composite a plurality of images saved in the HDD 204 to generate image data corresponding to one image. A printer I/F 209 receives image data sent from the image processing unit 208, and performs image processing on the image data with reference to attribute data added to the image data. The printer I/F 209 then outputs the image data which has undergone the image processing to the printer unit 113.

[0040] Note that in this embodiment, the MFP is a network MFP which displays a user interface (UI). The MFP, however, may be an image processing apparatus such as a general-purpose computer connected with a general-purpose scanner, a general-purpose printer, and the like.

[0041] (Hardware Configuration of Web Server)

Fig. 3 is a block diagram showing the hardware configuration of the Web server 120 shown in Fig. 1.

As shown in Fig. 3, a CPU 301, RAM 302, ROM 303, network I/F 305, and hard disk drive (HDD) 306 are communicably connected with each other via a system bus 309. A display device 307 such as CRT, an input device 304 such as a keyboard, and a pointing device 308 such as a mouse are also communicably connected with each other via the system bus 309.

[0042] The ROM 303 or HDD 306 stores a control program such as an operating system or Web application. The CPU 301 reads out the control program from the ROM 303 or HDD 306 into the RAM 302, and then executes it, thereby causing the computer to function as the Web server 120.

[0043] The CPU 301 displays various pieces of information using the display device 307, and also accepts a user instruction or the like from the input device 304 or pointing device 308. The CPU 301 communicates with other devices (for example, the MFP 110) on the LAN 100 via the network I/F 305.

[0044] (Software Configuration of MFP)

Fig. 4 is a block diagram showing the configuration of the software modules of the MFP 110 associated with communication processing in this embodiment. These software modules are stored in the HDD 204 of the MFP 110. The CPU 201 reads out the software modules from the HDD 204 into the RAM 203, and executes them, thereby implementing functions of the

software modules in the MFP 110. An HTTP communication unit 401 is a software module which makes communication via the network interface 206 according to the HTTP protocol.

[0045] A Web browser 402 is a browser which functions as a display control means, and communicates with the Web server 120 via the HTTP communication unit 401. The Web browser 402 requests, of the Web server 120, HTML data (screen information) of an operation screen described in HTML. The Web browser 402 renders the HTML data of the operation screen received from the Web server 120 via the HTTP communication unit 401, and displays the operation screen on the operation unit 111. The Web browser 402 then notifies the Web server 120 of information input through the operation screen displayed on the operation unit 111.

[0046] A script execution unit 403 receives a script generated by the Web server 120 from the Web server 120 via the Web browser 402, and parses the received script. With this operation, the script execution unit 403 extracts one or more pieces of job information contained in the received script. The script execution unit 403 also instructs a job management unit 404 to execute jobs based on the extracted job information in an order specified in the received script. The script execution unit 403 causes a Web service server 406, which provides a service (Web

service) in the MFP 110, to provide as a Web service a processing interface for causing the script execution unit 403 to execute processing. Examples of the processing interface (Web service) of the script execution unit 403 include an interface for acquiring information indicating an execution status of a script including a status of a job in progress, and an interface for controlling a job managed by the job management unit 404. In the case of a scan job, control of the job includes scan cancellation, and control of replacing a document to restart scanning.

[0047] The job management unit 404 manages job information in response to an instruction from the script execution unit 403, and instructs a job execution unit 405 to execute a job based on the managed job information. The job execution unit 405 is a module for executing processing (a job) based on job information in response to an instruction from the job management unit 404. If, for example, the job to be executed is a scan job, the job execution unit 405 executes a job for reading a paper document by operating the scanner unit 112, and saving image data in the HDD 204. If the job to be executed is a transmission job, the job execution unit 405 executes a job for transmitting image data stored in the HDD 204 by operating the network I/F 206. In this embodiment, the script execution unit 403, job management unit 404,

and job execution unit 405 function as an execution control means.

[0048] The Web service server 406 is a software module, which functions as a service providing means, has a server function of providing a Web service. The Web service server 406 provides a Web service for controlling the script execution unit 403 when it is called by the script execution unit 403. The Web service server 406 can call the script execution unit 403 in accordance with the called Web service.

[0049] (Software Configuration of Web Server)

Fig. 5 is a block diagram showing the configuration of the software modules of the Web server 120 in this embodiment. The Web server 120 includes a Web application 501 and HTTP communication unit 504 as software modules. The Web application 501 includes a presentation unit 502 and logic unit 503. These software modules are stored in the HDD 306 of the Web server 120. The CPU 301 reads out the software modules from the HDD 306 into the RAM 302, and executes them, thereby implementing functions of the software modules in the Web server 120.

[0050] In response to a request from the MFP 110, the presentation unit 502 transmits, as a response, HTML data of an operation screen to be displayed on the Web browser 402 of the MFP 110 to the MFP 110 via the HTTP communication unit 504. The presentation unit 502

receives, from the MFP 110 via the HTTP communication unit 504, input information which has been input by the user through the operation screen displayed by the Web browser 402 of the MFP 110.

[0051] The logic unit 503 processes the input information from the MFP 110, which has been passed from the presentation unit 502. For example, the logic unit 503, based on the user input information, changes a screen to be displayed in the MFP 110, and generates job definition information (a script) for causing the MFP 110 to execute a function of the device of the MFP 110. The HTTP communication unit 504 serves as a software module for operating the network I/F 305 to make communication according to the HTTP protocol.

[0052] (Execution Sequence of Job)

Fig. 6 is a sequence chart showing a series of processes executed in the information processing system in this embodiment. As an example of processing executed in the information processing system, "scan and transmission processing" executed according to the sequence chart shown in Fig. 6 will be described. In this embodiment, "scan and transmission processing" corresponds to processing of interactively scanning and reading a paper sheet placed on a document table to generate image data using the scanner unit 112 of the MFP 110, and transmitting the generated image data to an FTP server (not shown) on the LAN 100.

[0053] When the user presses a hardware button (not shown) for executing a transmission function, which is provided on the operation unit 111 of the MFP 110, the MFP 110 starts the Web browser 402. In order to display, on the operation unit 111, a setting screen for executing processing in the MFP 110, the Web browser 402 requests, of the Web application 501 in the Web server 120, HTML data of the setting screen. In this example, assume that the MFP 110 has been configured, in advance, to request the HTML data of the setting screen based on a URL indicating the location of the contents of the setting screen at startup.

[0054] In step S601, the Web browser 402 of the MFP 110 sends a request (setting screen request) to the Web application 501 holding the contents of the setting screen specified by the above URL using the GET command of HTTP. More specifically, when the Web browser 402 transmits the URL of the setting screen to the HTTP communication unit 401, the HTTP communication unit 401 communicates with the HTTP communication unit 504 of the Web server 120 according to the HTTP protocol. Upon receiving the URL from the HTTP communication unit 401, the HTTP communication unit 504 notifies the Web application 501 that an appropriate request has been sent. With this processing, the Web browser 402 sends the request to the Web application 501. Note that a description of an operation of the HTTP communication

units 401 and 504 which is inevitably executed when the MFP 110 and the Web server 120 communicate with each other will be omitted for the sake of simplicity.

[0055] In step S602, in response to the request in step S601, the presentation unit 502 of the Web application 501 generates HTML data (setting screen data) of the setting screen, and transmits the generated HTML data to the Web browser 402 as a response to the request. The Web browser 402 parses the received HTML data, and displays the setting screen (operation screen) on the operation unit 111.

[0056] Fig. 7 shows an example of the setting screen displayed on the operation unit 111 by the Web browser 402 as a result of step S602. A setting screen (GUI panel) 701 includes a text box 702 for setting a file name, and a radio button 703 for selecting an image format. In this example, it has been set in the setting screen 701 that a file name "test.pdf" is to be given to image data, and image data is to be generated in a PDF format. A GUI button 704 is used to instruct execution of a function based on the setting contents. When the user presses the GUI button 704, execution of processing (start of scanning) is instructed.

[0057] In step S603, the Web browser 402 notifies the Web application 501 that execution of the processing has been instructed. When the user presses the GUI button 704, the Web browser 402 notifies the

Web application 501 of parameters which have been set using the text box 702 and radio button 703 as setting information (set values). Assume that this communication processing uses the POST command of the HTTP protocol. With this operation, the Web browser 402 transmits, to the Web application 501, the input information which has been input through the operation screen based on the screen information provided from the Web application 501 (Web server 120).

[0058] In step S604, in response to the request in step S603, the presentation unit 502 of the Web application 501 generates a processing execution request containing HTML data (execution screen data) of a screen to be displayed during execution of the processing in the MFP 110. Based on setting information contained in the POST command in step S603, the logic unit 503 generates a script for causing the MFP 110 to execute processing corresponding to the setting information. In step S604, in response to the notification in step S603, the Web application 501 transmits, as response data, the execution request containing the generated execution screen data and the script to the Web browser 402. The Web browser 402 receives the response data, and extracts the script (Fig. 8) and the HTML data (Figs. 9A and 9B) of the execution screen from the received response data. As will be described in detail below, the script

corresponds to definition information of a service (Web service) associated with the input information transmitted from the Web browser 402, and contains a URL for specifying the service. The Web browser 402 uses a Web service provided by the Web service server 406 by specifying the Web service by the URL contained in the script.

[0059] With reference to Fig. 8, the script which is received by the Web browser 402 from the Web application 501 in step S604 will be explained. The script shown in Fig. 8 is an example of a script for sequentially executing, in the MFP 110, a scan job interactively executed based on a user input and a transmission job. This script assigns an interface for acquiring a state of the script execution unit 403 and that for controlling the scan job to the Web services provided by the Web service server 406. The script shown in Fig. 8 is described in XML but may have any format as long as the MFP 110 can process the format.

[0060] A Binding tag 801 describes which processing interface of the script execution unit 403 is provided as a Web service. In this example, processing (indicated by a URI "http://www.kanon.com/jobScript/status/getStatus") for acquiring information representing an execution status of the script of the script execution unit 403 is defined as "getStatus". The whole processing is

defined as "status Binding".

[0061] A Binding tag 802 is similar to the tag 801. In this example, processing (indicated by a URI "http://www.kanon.com/jobScript/Scan/scanStart") for starting scanning of the scan job is defined as "scanStart". Processing (http://www.kanon.com/jobScript/Scan/scanEnd) for terminating scanning of the scan job is defined as "scanEnd". A set of the two processes is defined as "scanBinding".

[0062] A Service tag 803 defines association of the processing indicated by the tag 801 with an actual Web service. In this example, it is defined that the processing defined as "statusBinding" is assigned to a Web service with a URL represented by "/JobScript/ID_012345/status/".

[0063] A Service tag 804 defines association of the processing indicated by the tag 802 with an actual Web service. In this example, it is defined that the processing defined as "scanBinding" is assigned to a Web service with a URL represented by "/JobScript/ID_012345/scan/".

[0064] As described above, in this example, definition for associating processing interfaces of software modules within the MFP 110 with Web services is formed by a script. The definition information of the Web services is used to disclose, among the

processing interfaces (Web services) provided by the script execution unit 403, processing interfaces which are needed for displaying a screen based on HTML data (to be described later).

[0065] In this example, the URLs specified by the tags 801 and 802 are different for each script, and correspond to the HTML data in Figs. 9A and 9B (to be described later). Even when a plurality of scripts are simultaneously executed in the MFP 110, therefore, it is possible to handle processes executed by the scripts as different Web services.

[0066] A Scan tag 805 represents setting information of the scan job executed in the script processing. In this example, the tag 805 indicates that the scanner unit 112 should automatically determine (Auto) a scan document size (DocumentSize). The tag 805 also indicates that a color mode (ColorMode) is gray scale (GRAY_SCALE).

[0067] A Send tag 806 represents setting information of the transmission job executed in the script processing. In this example, the tag 806 indicates that transmission to a server "ftpserver.xxx.com" should be done according to the FTP protocol. The tag 806 also indicates that the format (FileFormat) of a file to be transmitted is PDF and a file name (FileName) is "test".

[0068] Referring back to Fig. 6, in step S604, the

logic unit 503 of the Web application 501 generates and transmits the above-described script, and then the Web browser 402 receives this script.

[0069] With reference to Figs. 9A and 9B, the HTML data (execution screen data) received by the Web browser 402 from the Web application 501 will now be described. A JavaScript code is embedded in the HTML data shown in Figs. 9A and 9B. In the MFP 110, the JavaScript code embedded in the HTML data calls the Web service server 406, thereby executing display update processing and interaction processing with the user by the script execution unit 403 in accordance with an execution status of the script. This shows an example of the HTML data used for displaying a screen on the operation unit 111 up to when the interaction processing associated with the scan job executed according to the script shown in Fig. 8 is performed and the transmission job is executed.

[0070] Blocks 910, 920, 930, and 940 include JavaScript codes for updating screen display on the operation unit 111 based on an execution status of the script.

[0071] The block 910 defines that a timer is set for a predetermined time period (for example, two seconds), and when a time-out occurs, a `getScriptStatus` function is called. When a screen based on the HTML data is displayed, the Web browser 402 executes the

JavaScript code in the block 910.

[0072] The getScriptStatus function in the block 920 includes blocks 921, 922, 923, and 924. This function is described in JavaScript for performing HTTP communication with a Web service of the Web service server 406 in the background, and acquiring information indicating an execution status of the script through the communication process.

[0073] The block 921 defines that an object (XMLHttpRequest) for performing background communication is generated, and processing to be executed in communication in the blocks 922 and 923 is registered.

[0074] The block 922 is a portion called when background communication succeeds, and defines that whether a response acquired in the communication process indicates a scan standby state (a character string "SCAN_SUSPEND") is determined. According to the block 922, if the response indicates a scan standby state, a scan end button (a button with an ID "buttonScanEnd") and a scan start button (a button with an ID "buttonScanStart") are enabled; otherwise, the display of these buttons is disabled.

[0075] The block 923 is a portion called when background communication succeeds, and defines that whether a response acquired in the communication process indicates a transmission state (a character

string "SEND_START") or a transmission end state (a character string "SEND_END") is determined. In this example, these two states mean that the scan job defined in the script has been already terminated, and the process has advanced to the transmission job.

[0076] If it is determined that the scan job is still in progress, a timer is set for two seconds again. When a time-out occurs, the getScriptStatus function in the block 920 is called. If it is determined that the transmission job has started, a block 951 (a display region indicated by message_scan) is caused not to be displayed and a block 952 (a region indicated by message_send) is caused to be displayed.

[0077] In the block 924, communication is executed by specifying an URL in the object for background communication generated in the block 921. The URL is defined in the script passed from the Web server 120, as explained in Fig. 8.

[0078] As described above, the JavaScript code in the block 920 defines that the URL of the Web service enabled by the script received from the Web application 501 together with the HTML data is used to acquire the information indicating an execution status of the script execution unit 403, thereby updating the display.

[0079] A scanEnd function in the block 930 is described in JavaScript for performing HTTP communication with the Web service server 406 in the

background, and instructing to terminate scanning. Similarly to the block 920, the URL of the Web service for controlling the scan job defined in the script received from the Web server 120 is accessed. Thus, this function instructs, via the Web service server 406, the script execution unit 403 to terminate scanning.

[0080] A scanStart function in the block 940 is described in JavaScript for performing HTTP communication with the Web service server 406 in the background, and instructing to start scanning. Similarly to the block 930, the URL of the Web service for controlling the scan job defined in the script received from the Web server 120 is accessed. Thus, this function instructs, via the Web service server 406, the script execution unit 403 to start scanning.

[0081] A block 950 is a portion describing screen information in HTML. The block 951 represents a description for displaying, on a screen, a message indicating that scanning is in progress. The block 952 represents a description for displaying, on a screen, a message indicating that transmission has started. As for screen display based on these descriptions, the JavaScript code in the block 923 controls display/non-display on the screen, as described above.

[0082] A block 953 represents a description for displaying an execution status of the script on the screen. A block 954 represents a description about

display of a button for instructing termination of the scan job. When the button is pressed, the JavaScript code in the block 930 is called. The JavaScript code in the block 922 switches between enabling and disabling of the button.

[0083] A block 955 represents a description about display of a button for instructing start of scanning in the scan job. When the button is pressed, the JavaScript code in the block 940 is called. The JavaScript code in the block 922 switches between enabling and disabling of the button. The above-described HTML data controls the screen using the Web service defined based on the script in Fig. 8.

[0084] Referring back to Fig. 6, in step S605, among the execution screen data (screen information) and the script (Web service definition information) contained in the execution request received from the Web server 120 in step S604, the Web browser 402 transmits the script to the script execution unit 403. With this operation, the Web browser 402 sends an execution instruction of the script to the script execution unit 403. Upon receiving the script, the script execution unit 403 executes the received script, as shown in Fig. 10. In this way, the script execution unit 403 executes processing according to the execution request from the Web server 120 (Web application 501).

[0085] Fig. 10 is a flowchart illustrating a

processing procedure of executing, by the script execution unit 403, the script received from the Web application 501. In step S1001, the script execution unit 403 parses the script received in step S605, and registers in the Web service server 406 URLs described in the script. The script execution unit 403, in turn, registers in the Web service server 406 Web services defined by the definition information described in the script. With this operation, the Web service server 406 provides, among a plurality of providable Web services, the Web services registered by the script execution unit 403. The Web browser 402 can use the Web services registered by the script execution unit 403 among the Web services providable by the Web service server 406.

[0086] As detailed registration processing, the script execution unit 403 configures the Web service server 406 to call the script execution unit 403 when the Web browser 402 sends a request for the registered URLs. In the example of Fig. 8, the script execution unit 403 enables two Web services via the Web service server 406. Note that step S1001 corresponds to step S606.

[0087] In step S1002, the script execution unit 403 extracts one or more pieces of job information from the script (divides the script into jobs), and registers the extracted job information in the job

management unit 404. In the example of Fig. 8, the script execution unit 403 registers, as Web services, information about two jobs including a scan job and a transmission job in the job management unit 404. In step S1003, in response to a request from the script execution unit 403 via the job management unit 404, the job execution unit 405 sequentially executes the registered jobs.

[0088] Upon accepting the request for the registered Web services (from the Web browser 402), the Web service server 406 instructs the script execution unit 403 to execute processing corresponding to the request. If the script execution unit 403 receives the instruction from the Web service server 406 in step S1004, the process advances to step S1005; otherwise, the process advances to step S1006. In step S1005, the script execution unit 403 executes the processing corresponding to the request accepted by the Web service server 406 by performing the processing instructed from the Web service server 406.

[0089] In the example of Fig. 8, if a Web service for acquiring information indicating an execution status of the script is called (that is, execution of the Web service is requested), the script execution unit 403 determines the execution status of the script, and transmits, as a response, a determination result to a client (that is, the Web browser 402) as a request

source via the Web service server 406. If a Web service for controlling execution of the scan job is called, the script execution unit 403 controls, via the job management unit 404, execution of the scan job by the job execution unit 405.

[0090] In step S1006, based on a monitoring result of the job execution unit 405 by the job management unit 404, the script execution unit 403 determines whether execution of the job is complete. If the job is complete, the process advances to step S1007; otherwise, the process advances to step S1004. In step S1007, based on the monitoring result of the job execution unit 405 by the job management unit 404, the script execution unit 403 determines whether all the jobs described in the script are complete. If not all of the jobs are complete, the process advances to step S1003 to execute the next job; otherwise, the process advances to step S1008.

[0091] In step S1008, upon accepting a request for the registered Web services, the Web service server 406 instructs the script execution unit 403 to execute processing corresponding to the request, as in step S1004. If the Web service server 406 accepts a request for the Web services, the process advances to step S1009; otherwise, the process advances to step S1010. In step S1009, the script execution unit 403 executes the processing corresponding to the request accepted by

the Web service server 406 by performing the processing instructed from the Web service server 406, as in step S1005.

[0092] In step S1010, the script execution unit 403 determines whether to disable the Web services. If the unit 403 determines to disable the Web services, the process advances to step S1011; otherwise, the process advances to step S1008. In this embodiment, as an example, when a predetermined time period elapses after execution of the jobs ends, the script execution unit 403 determines to disable the Web services. This time period may be set in the script execution unit 403 in advance, or may be described within the script received from the Web server 120. That is, the Web service server 406 provides the Web services until the Web server 120 instructs, using the script, to disable the services or until a predetermined time period elapses after execution of the processing (jobs) by the script execution unit 403 ends.

[0093] If the script execution unit 403 determines in step S1010 to disable the Web services, it causes the Web service server 406 to delete the URLs registered in the Web service server 406 in step S1001, thereby disabling the Web services corresponding to the URLs in step S1011.

[0094] Referring back to Fig. 6, according to the processing sequence shown in Fig. 10, display control

based on the HTML data (Figs. 9A and 9B) by the Web browser 402 will be described. In step S605 described above, the Web browser 402 transmits the script to the script execution unit 403 to instruct execution of the script, and parses the HTML data (execution screen data) received from the Web application 501 together with the script to display a screen based on the HTML data on the operation unit 111.

[0095] Fig. 11A shows an example of the screen displayed on the operation unit 111 by the Web browser 402. As shown in Fig. 11A, the Web browser 402 displays information 1102 based on the block 951 of Fig. 9B and information 1103 based on the block 953 of Fig. 9B within a display region 1101. The information 1103 represents, as initial display about an execution status of the script, "SCAN_SCANNING" indicating that scanning processing is in progress.

[0096] Step S606 corresponds to step S1001 in Fig. 10, in which the script execution unit 403 registers in the Web service server 406 the Web services based on the script received in step S605. In this example, the unit 403 registers in the Web service server 406 two Web services including a Web service for acquiring information indicating an execution status of the script in the Web service server 406, and that for controlling execution of the scan job. This registration processing is implemented by registering,

in the Web service server 406, URLs corresponding to the Web services described in the script. This allows the Web browser 402 to use the Web services by specifying the registered URLs.

[0097] In step S607, the Web browser 402 uses the JavaScript code in the block 920 contained in the HTML data (Figs. 9A and 9B) to call the Web service for acquiring information indicating an execution status of the script among the Web services enabled in step S606.

[0098] Step S608 corresponds to step S1004 in Fig. 10, in which the Web service server 406 calls the script execution unit 403 to execute processing corresponding to the Web service requested from the Web browser 402. In response to this, the script execution unit 403 transmits an execution status of the script to the Web browser 402 via the Web service server 406. In this example, assume that the scan job is in a standby state at this time, and the unit 403 transmits a character string "SCAN_SUSPEND" indicating a state of the scan job to the Web browser 402. The Web browser 402 changes the display according to the JavaScript code in the block 922. As described above, the Web browser 402 acquires information indicating an execution status of the script based on the timer of the JavaScript code via the Web service server 406, and changes the screen display based on the acquired information.

[0099] Fig. 11B shows an example of a screen displayed on the operation unit 111 by the Web browser 402 as a result of step S608. As shown in Fig. 11B, the Web browser 402 displays information 1202 based on the block 951 of Fig. 9B and information 1203 based on the block 953 of Fig. 9B within a display region 1201. The Web browser 402 processes the JavaScript code in the block 922 to display GUI buttons 1204 and 1205 based on the blocks 954 and 955 of Fig. 9B, respectively.

[0100] If the user presses the GUI button 1205 of Fig. 11B in step S609, processing described in the script as processing executed upon press of the GUI button 1205 is performed. More specifically, as described in Figs. 9A and 9B, when the user presses the GUI button 1205, the Web browser 402 executes the JavaScript code in the block 940.

[0101] Upon executing the JavaScript code in the block 940, in step S610, the Web browser 402 calls the Web service for controlling execution of the scan job, which has been registered in step S605.

[0102] When the Web service is called in step S610, the Web service server 406 calls the script execution unit 403 in step S611. As explained in steps S1004 and S1005 of Fig. 10, the script execution unit 403 controls, via the job management unit 404, the job execution unit 405 to start scanning of the scan job.

As described above, the Web service used in steps S610 and S611 corresponds to a service for controlling job execution by the script execution unit 403 based on information input through the operation screen displayed by the Web browser 402 during execution of the scan job.

[0103] As in steps S607 and S608, in steps S612 and S613, the JavaScript code in the block 920 called based on the timer calls the Web service for acquiring an execution status of the script, and causes the script execution unit 403 to execute processing corresponding to the Web service.

[0104] Assume that the scan processing is in a standby state again and the screen shown in Fig. 11B is displayed. At this time, in step S614, when the user presses the GUI button 1204 shown in Fig. 11B, processing described in the script as processing executed upon press of the GUI button 1204 is performed. More particularly, as described in Figs. 9A and 9B, when the user presses the GUI button 1205, the Web browser 402 executes the JavaScript code in the block 930.

[0105] In steps S615 and S616, the Web browser 402 and the Web service server 406 execute processing, similarly to steps S610 and S611. Then, the script execution unit 403 controls, via the job management unit 404, the job execution unit 405 to terminate the

scan job. As described above, the Web service used in steps S615 and S616 corresponds to a service for controlling job execution by the script execution unit 403 based on information input through the operation screen displayed by the Web browser 402 during the scan job.

[0106] In steps S617 and S618, the JavaScript code in the block 920 called based on the timer calls the Web service server 406 for acquiring information indicating an execution status of the script, similarly to steps S607 and S608. At this time, the scan job is complete and the transmission job is in progress. The Web browser 402, therefore, receives a character string "SEND_START" from the Web service. The Web browser 402 uses the JavaScript code in the block 923 to set non-display of information based on the block 951 and display of information based on the block 952.

[0107] Fig. 11C shows an example of a screen displayed by the Web browser 402 as a result of steps S617 and S618. As shown in Fig. 11C, the Web browser 402 displays the block 952 shown in Fig. 9B as display 1302 and a link 1303, and information 1304 based on the block 953 shown in Fig. 9B within a display region 1301. In this example, the user ends the interaction processing associated with the scan job at this time. If the user selects the link 1303, the Web browser 402 returns the display of the operation unit 111 to an

initial screen.

[0108] Finally, as in steps S1010 and S1011 of Fig. 10, the script execution unit 403 executes processing for disabling the Web services in the Web service server 406 in step S619. As described above, when a predetermined time period elapses after execution of the script (jobs) ends, the unit 403 executes the processing.

[0109] As explained above, in this embodiment, the Web browser 402 can control screen display and function execution in the MFP 110 by instructions from the Web application 501 (Web server 120). More specifically, the Web browser 402 displays an operation screen based on screen information provided by the Web server 120, and transmits input information input through the operation screen to the Web server 120. The Web browser 402 receives from the Web server 120 an execution request of processing corresponding to the input information, and passes a script contained in the received execution request to the script execution unit 403. The script contains definition information of a Web service for controlling the script execution unit 403. The script execution unit 403 registers in the Web service server 406 the Web service defined in the definition information contained in the script. The Web service server 406 provides the Web browser 402 with the registered Web service of Web services

providable during execution of the processing by the script execution unit 403.

[0110] In this way, when executing a function of the MFP 110 according to a request from the Web server 120 in the MFP 110, it is possible to enable a service for controlling the execution processing as needed. Consequently, as compared with a case in which all the providable services are always available in the MFP 110, it is possible to avoid wasting the resources of the MFP 110 such as a memory.

[0111] In addition to the above-described script, the execution request from the Web server 120 contains screen information about an operation screen to be displayed by the Web browser 402 while the script execution unit 403 executes the processing according to the execution request. The Web browser 402 uses a service provided by the Web service server 406 to control display based on the screen information received from the Web server 120. According to this embodiment, in display control by the Web browser 402 during execution of the processing by the script execution unit 403, a service provided by the Web service server 406 is enabled as needed, thereby avoiding wasting the resources of the MFP 110, as described above.

[0112] It is possible to change a URL for specifying a Web service by an instruction from the Web

server 120. As compared with a case in which the MFP 110 has a fixed Web service, therefore, it is possible to form a function necessary for display control as a necessary Web service, thereby allowing more flexible control in the MFP 110.

[0113]<Second Embodiment>

In the first embodiment, the MFP 110 executes processing based on the HTML data and script received from the Web application 501. In the second embodiment, a case in which display control of an MFP 110 is done when a Web application 501 transmits appropriate HTML data to the MFP 110 in accordance with an execution status of a script will be explained. In the second embodiment, for the sake of simplicity, a description of the same parts as in the first embodiment will be omitted, and different parts will be mainly explained. More specifically, the second embodiment is different from the first embodiment in respective processing sequences executed by software modules in the MFP 110 and those in a Web server 120.

[0114] (Execution Sequence of Job)

Fig. 12 is a sequence chart showing a series of processes executed in the information processing system in the second embodiment. With reference to Fig. 12, as in Fig. 6, a case in which scan and transmission processing is executed in the MFP 110 will be explained. After start of a Web browser 402, processing from step

S1401 to step S1406 is the same as that from step S601 to step S606 in the first embodiment. In the second embodiment, however, in step S1404, the Web browser 402 receives, from the Web application 501, HTML data different from that (Figs. 9A and 9B) in the first embodiment.

[0115] Fig. 13 shows an example of the HTML data transmitted from the Web application 501 to the Web browser 402 in step S1404. As shown in Fig. 13, unlike the first embodiment, the HTML data does not describe the JavaScript code for calling a Web service by the Web browser 402. Instead, in the second embodiment, based on a JavaScript code in a block 1510, the Web browser 402 sends a request to the Web application 501 when a predetermined time period elapses. In step S1404, the Web application 501 transmits, to the Web browser 402, response data containing the HTML data shown in Fig. 13 and the script explained in the first embodiment.

[0116] In step S1407, the Web browser 402 sends a request (screen request) of screen information to the Web application 501 based on the HTML data (Fig. 13). In steps S1408 and S1409, the Web application 501 communicates with a Web service server 406 in response to the screen request in step S1407, and calls a Web service for acquiring an execution status of the script. At this time, the Web application 501 specifies a URL

for the Web service contained in the script transmitted in step S1404. In this example, as a result of step S1408, the Web application 501 receives a character string indicating that a scan job is in a standby state.

[0117] In step S1410, the Web application 501 acquires information indicating that the scan job is in a standby state as a result of step S1408. As in Fig. 11B, therefore, the Web application 501 transmits the HTML data for interaction processing with the user to the Web browser 402. In the first embodiment, when the user presses the GUI button 1204 or 1205, the Web browser 402 uses the JavaScript code to call the Web service. On the contrary, in this embodiment, the Web application 501 notifies the Web service server 406 of it.

[0118] In step S1411, the user instructs to terminate the scan job through the same screen as in Fig. 11B. Upon receiving a notification indicating end of scanning from the Web browser 402 in step S1411, the Web application 501 communicates with the Web service server 406 to call the Web service for controlling the scan job in step S1412. At this time, the Web application 501 transmits a request to the URL specified in step S1404 to call the Web service.

[0119] In step S1413, the Web service server 406 calls a script execution unit 403 as in step S1409. Consequently, the script execution unit 403 controls

the scan job as in the first embodiment. After that, in step S1414, the Web application 501 transmits, to the Web browser 402, the HTML data for displaying the same screen as in Fig. 11C as a response to the request from the Web browser 402 in step S1411.

[0120] In step S1415, the script execution unit 403 disables the Web service of the Web service server 406, similarly to the first embodiment.

[0121] As described above, in the second embodiment, while the script execution unit 403 executes processing according to a processing execution request from the Web application 501 (Web server 120), the Web browser 402 requests screen information of the Web server 120. Upon receiving the request, the Web server 120 uses the provided Web service which has been registered in the Web service server 406 of the MFP 110 to generate screen information to be displayed by the Web browser 402, and transmits the generated information to the Web browser 402. The Web browser 402 displays an operation screen based on the received screen information. According to the second embodiment, the Web server 120 can use, as needed, the Web service registered in the Web service server 406 to perform display control of the MFP 110 even if the Web server 120 has a leading role in performing the display control processing instead of the Web browser 402 while the script execution unit 403 executes the processing.

[0122]<Other Embodiments>

In the above-described first and second embodiments, a Web service of the Web service server 406 is set to be disabled when a predetermined time period elapses after the script is executed. The present invention, however, is not limited to this. For example, a Web service server 406 may provide a Web service for disabling a Web service, or an instruction for disabling a Web service may be accepted according to any protocol. These conditions may be described in a script received from a Web application 501. In this way, it is possible to flexibly perform control for disabling a Web service of an MFP 110 from the Web application 501.

[0123] Aspects of the present invention can also be realized by a computer of a system or apparatus (or devices such as a CPU or MPU) that reads out and executes a program recorded on a memory device to perform the functions of the above-described embodiment(s), and by a method, the steps of which are performed by a computer of a system or apparatus by, for example, reading out and executing a program recorded on a memory device to perform the functions of the above-described embodiment(s). For this purpose, the program is provided to the computer for example via a network or from a recording medium of various types serving as the memory device (e.g., computer-readable

medium).

[0124] While the present invention has been described with reference to exemplary embodiments, it is to be understood that the invention is not limited to the disclosed exemplary embodiments. The scope of the following claims is to be accorded the broadest interpretation so as to encompass all such modifications and equivalent structures and functions.

[0125] This application claims the benefit of Japanese Patent Application Nos. 2010-130448, filed June 7, 2010 and 2011-112898, filed May 19, 2011, which are hereby incorporated by reference herein in their entirety.

CLAIMS

1. An information processing apparatus connectable to a Web server, the information processing apparatus comprising:

display control means including

transmission means for displaying an operation screen based on screen information provided by the Web server, and transmitting input information input through the operation screen to the Web server, and

reception means for receiving, from the Web server, an execution request of processing corresponding to the input information transmitted by said transmission means, the execution request containing definition information of a service associated with execution of the processing;

execution control means for executing, upon receiving the execution request from the Web server via said display control means, processing according to the execution request; and

service providing means for providing, among a plurality of providable services for controlling said execution control means, a service which is registered by said execution control means,

said execution control means including registration means for registering in said service providing means the service defined in the definition

information contained in the execution request when starting execution of the processing according to the execution request.

2. The apparatus according to claim 1, wherein said service providing means provides the service registered by said execution control means until the Web server instructs to disable the service or until a predetermined time period elapses after the processing according to the execution request by said execution control means ends.

3. The apparatus according to claim 1 or 2, wherein the execution request further contains screen information about an operation screen to be displayed during execution of the processing according to the execution request by said execution control means, and said display control means uses the service provided by said service providing means to control display based on the screen information received from the Web server.

4. The apparatus according to claim 3, wherein the execution request contains definition information of a service for acquiring, from said execution control means, information indicating an execution status of the processing by said execution control means, and

said display control means acquires the information indicating the execution status of the

processing according to the execution request from said execution control means via the service provided by said service providing means, and updates the display based on the screen information in accordance with the acquired information.

5. The apparatus according to claim 3 or 4, wherein the execution request contains definition information of a service for controlling execution of the processing by said execution control means based on information input through the operation screen displayed by said display control means during execution of the processing by said execution control means, and

said display control means further includes means for instructing, via the service provided by said service providing means, said execution control means to control execution of the processing based on the information input through the operation screen being displayed.

6. The apparatus according to claim 3 or 4, wherein said display control means further comprises request means for requesting the screen information of the Web server during execution of the processing according to the execution request by said execution control means,

upon receiving the request from said request means, the Web server generates screen information to

be displayed by said display control means using the service provided by said service providing means, and transmits the generated screen information to said display control means, and

said display control means displays an operation screen based on the screen information received from the Web server.

7. The apparatus according to any one of claims 3 to 5, wherein

the execution request includes a script containing, as definition information of a service, a URL for specifying the service,

said service providing means provides, as a service to be registered by said execution control means, a Web service specified by the URL, and

said display control means serves as a Web browser which uses a Web service provided by said service providing means by specifying the Web service by the URL contained in the script corresponding to the execution request.

8. The apparatus according to any one of claims 1 to 7, wherein the processing according to the execution request is executed based on a scan job for reading an image on a document and generating image data.

9. A control method for controlling an information processing apparatus connectable to a Web server, the control method comprising:

causing display control means of the information processing apparatus to display an operation screen based on screen information provided by the Web server, and transmit input information input through the operation screen to the Web server;

causing the display control means to receive, from the Web server, an execution request of processing corresponding to the input information transmitted in the causing the display control means to display, the execution request containing definition information of a service associated with execution of the processing;

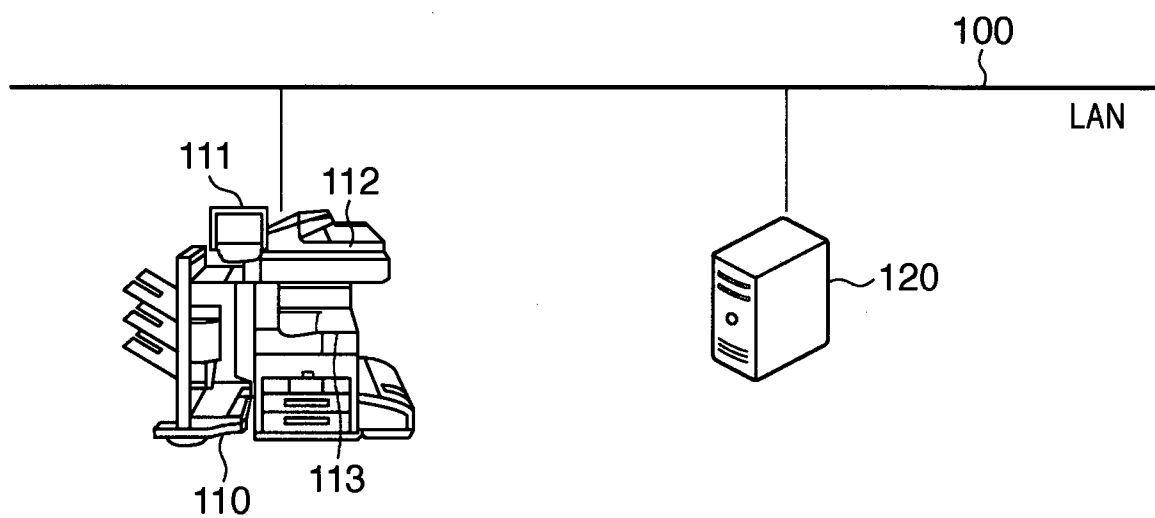
causing execution control means of the information processing apparatus to execute, upon receiving the execution request from the Web server via the display control means, processing according to the execution request; and

causing service providing means of the information processing apparatus to provide, among a plurality of providable services for controlling the execution control means, a service which is registered by the execution control means,

the causing the execution control means to execute including registering in the service providing means the service defined in the definition information contained in the execution request when starting execution of the processing according to the execution request.

10. A program for causing a computer to execute a control method for an information processing apparatus according to claim 9.

FIG. 1



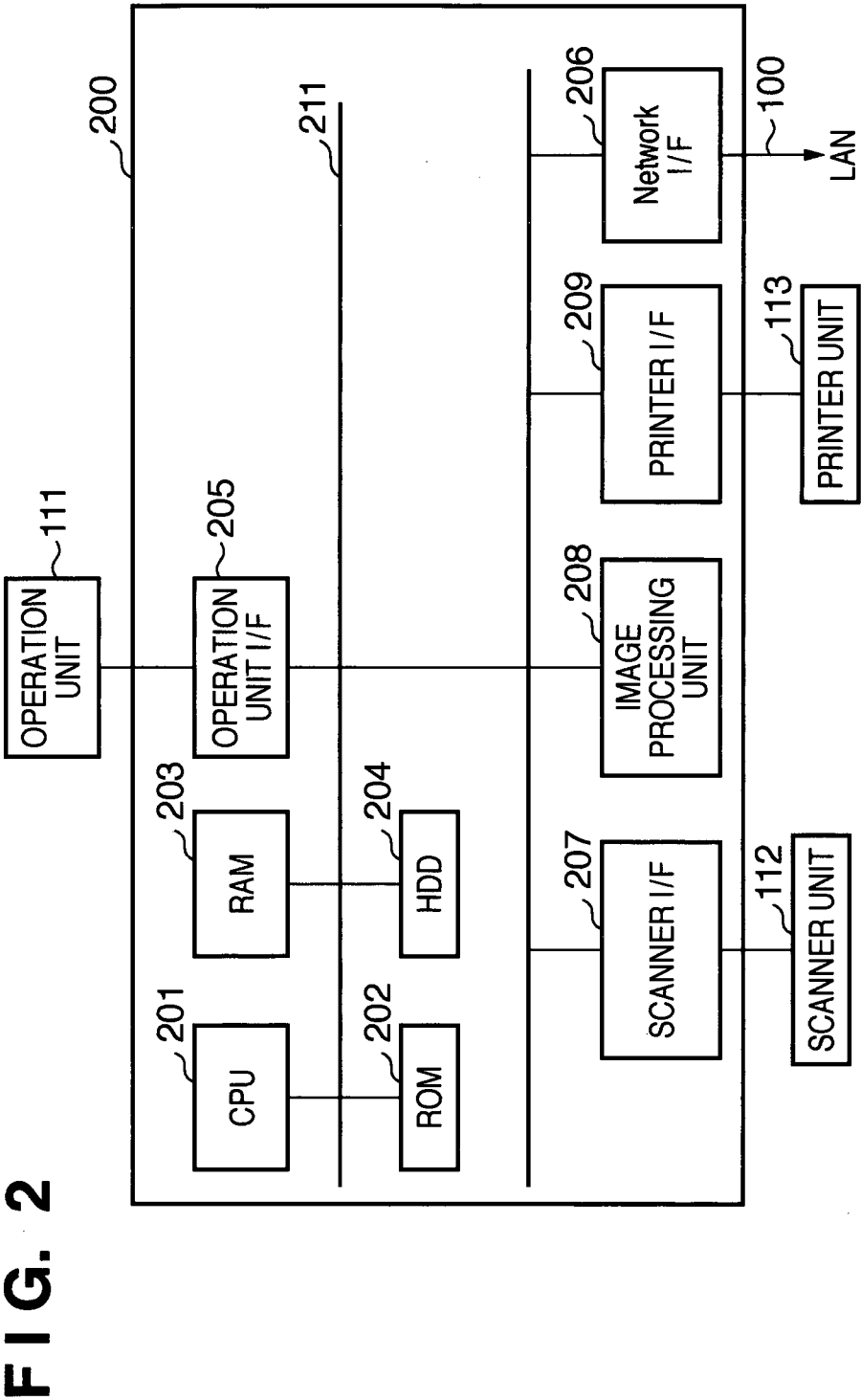
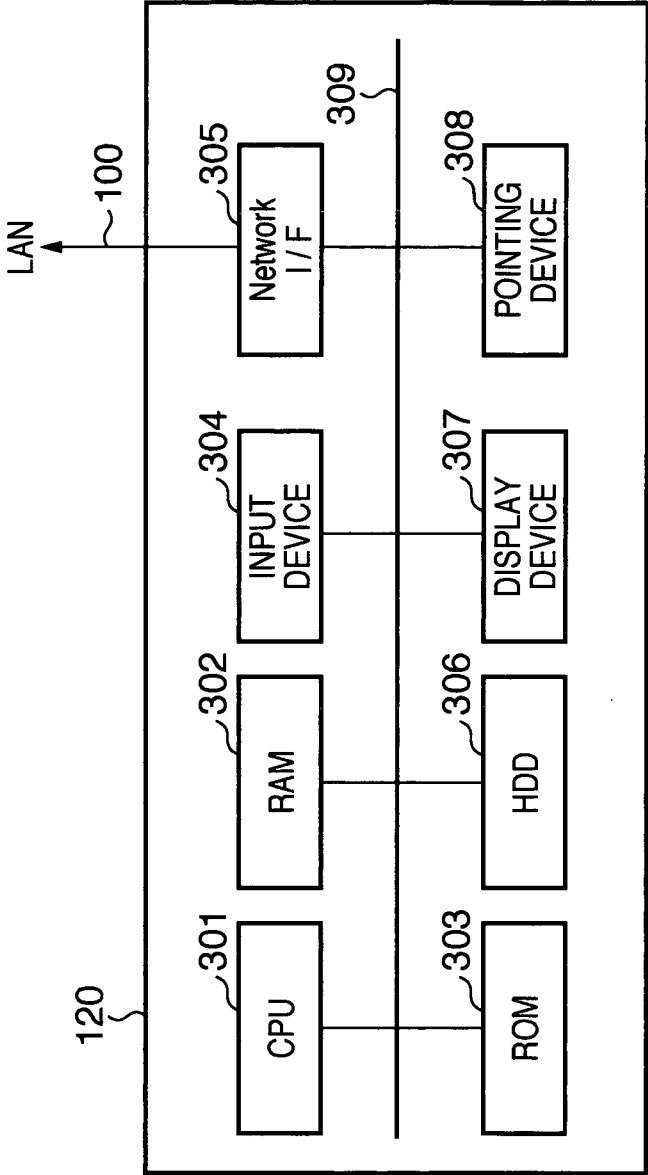


FIG. 3



4/13

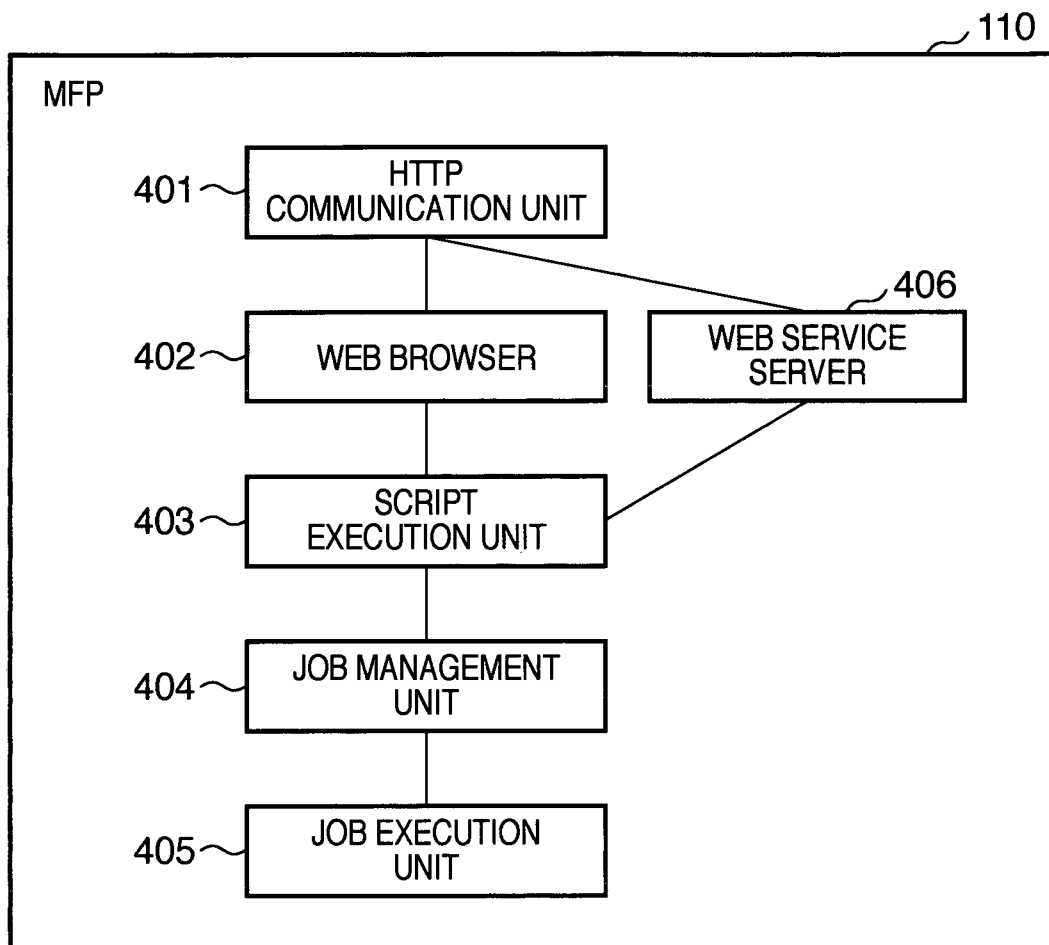
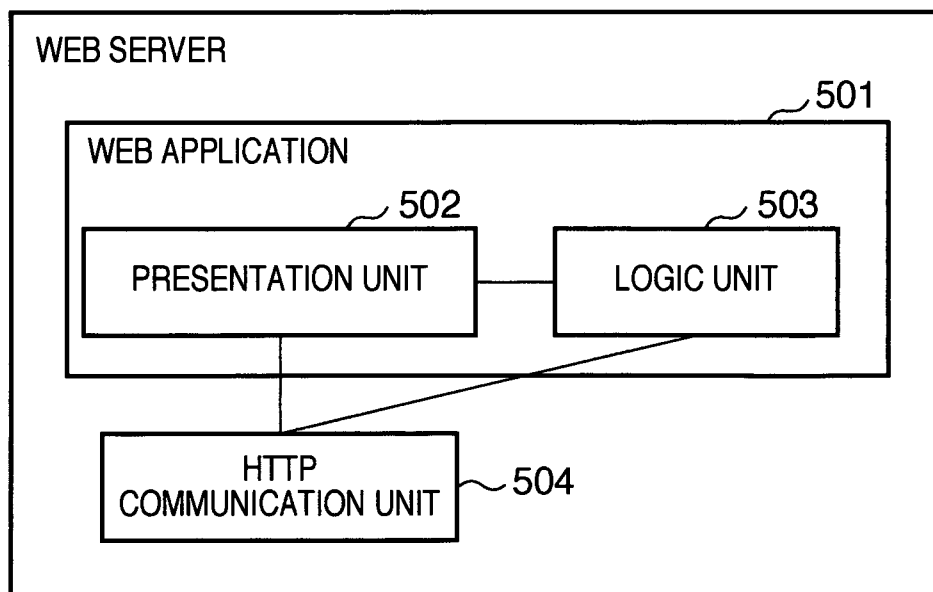
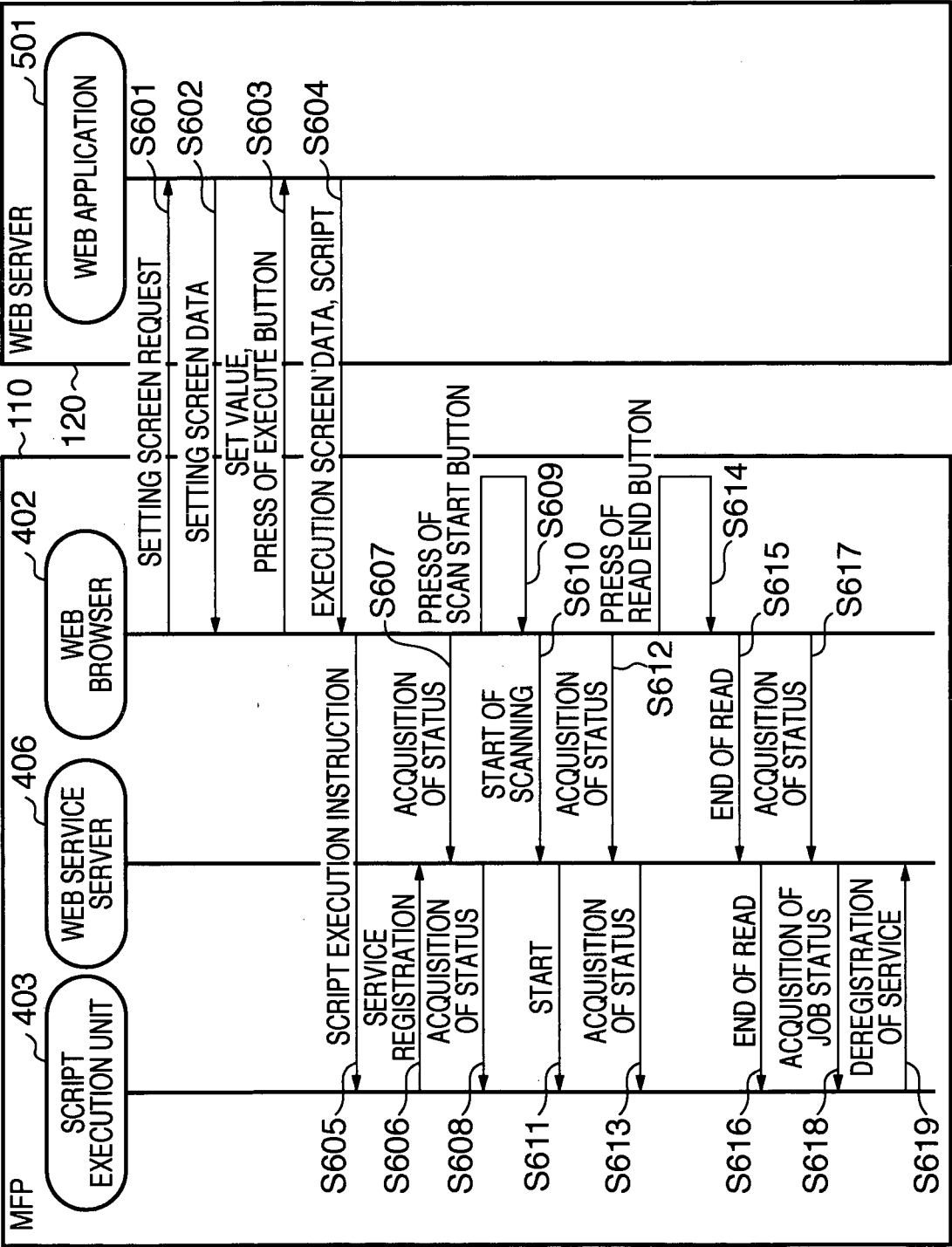
FIG. 4**FIG. 5**

FIG. 6



6/13

FIG. 7

The figure shows a web browser window titled "WEB BROWSER" with a sub-header "[TRANSMISSION SETTING]". The main content area contains a section titled "■ SCAN AND TRANSMIT TO FTP SERVER" followed by the instruction "PLEASE SET DOCUMENTS AND PRESS EXECUTE BUTTON". Below this is a form with two fields: "FILE NAME" with a text input containing "test.pdf", and "IMAGE FORMAT" with two radio button options, "PDF" (which is selected) and "JPEG". An "EXECUTE" button is located at the bottom right of the form area.

701

WEB BROWSER [TRANSMISSION SETTING]

■ SCAN AND TRANSMIT TO FTP SERVER

PLEASE SET DOCUMENTS AND PRESS EXECUTE BUTTON

FILE NAME test.pdf

IMAGE FORMAT ☒ PDF ☐ JPEG

EXECUTE

702

703

704

7/13

FIG. 8

```

<?xml version="1.0" encoding="UTF-8"?>
<JobScript xmlns="http://www.kanon.com/jobScript">
  <Binding name="statusBinding" type="status"> 801
    <Operation name="getStatus">
      <scpitOperation action="http://www.kanon.com/jobScript/status/getStatus"/>
    </Operation>
  </Binding>
  <Binding name="scanBinding" type="scan"> 802
    <Operation name="scanStart">
      <scpitOperation action="http://www.kanon.com/jobScript/scan/scanStart"/>
    </Operation>
    <Operation name="scanEnd">
      <scpitOperation action="http://www.kanon.com/jobScript/scan/scanEnd"/>
    </Operation>
  </Binding>
  <Service name="statusControl" binding="statusBinding"> 803
    <Port>
      <Address location="/JobScript/ID_012345/status"/>
    </Port>
  </Service>
  <Service name="scanControl" binding="scanBinding"> 804
    <Port>
      <Address location="/JobScript/ID_012345/scan"/>
    </Port>
  </Service>
  <Scan> 805
    <Settings>
      <DocumentSize>AUTO</DocumentSize>
      <ColorMode>GRAY_SCALE</ColorMode>
    </Settings>
  </Scan>
  <Send> 806
    <Settings>
      <FTPAddress>
        <HostName>ftpserver.xxx.com</HostName>
        <Path>ScanDocument</Path>
        <UserName>user</UserName>
        <Password>pass</Password>
      </FTPAddress>
      <FileFormat>PDF</FileFormat>
      <FileName>test</FileName>
    </Settings>
  </Send>
</JobScript>

```


9/13

FIG. 9B

```

function scanEnd() {
  document.getElementById("buttonScanEnd").disabled = true;
  document.getElementById("buttonScanStart").disabled = true;
  xmlhttp = new XMLHttpRequest();
  xmlhttp.open("POST", "http://localhost/jobScript/ID_012345/scan", true);
  xmlhttp.send("Operation=scanEnd");
}
function scanStart() {
  document.getElementById("buttonScanEnd").disabled = true;
  document.getElementById("buttonScanStart").disabled = true;
  xmlhttp = new XMLHttpRequest();
  xmlhttp.open("POST", "http://localhost/jobScript/ID_012345/scan", true);
  xmlhttp.send("Operation=scanStart");
}
-->
</script>
</head>
<body>
  <form id="form1" action=" " method="post">
    <div id="message_scan" style="visibility: visible"> 951
      <h1>SCANNING IS IN PROGRESS</h1>
    </div>
    <div id="message_send" style="visibility: hidden"> 952
      <h1>TRANSMISSION HAS STARTED</h1>
      <a href="menu.html">TO MENU</a>
    </div>
    SCRIPT STATUS : 953
    <input type="text" name="status" size="10" maxlength="8" />
    <input id="buttonScanEnd" Type="button" value="READ END" disabled="disabled" onclick="scanEnd();" /> 954
    <input id="buttonScanStart" Type="button" value="SCAN START" disabled="disabled" onclick="scanStart();" /> 955
  </form>
</body>
</html>

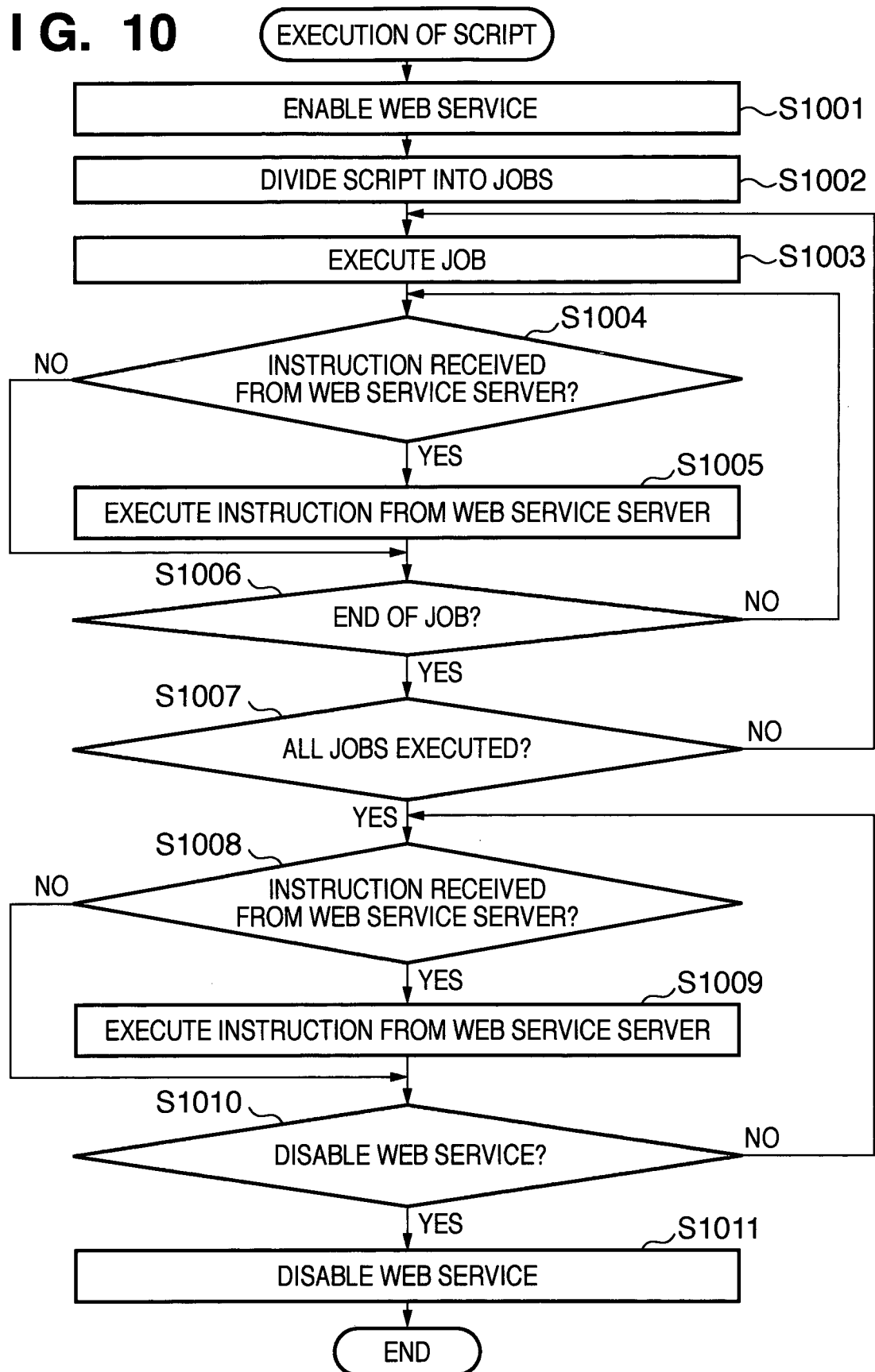
```

930

940

950

10/13

FIG. 10

11/13

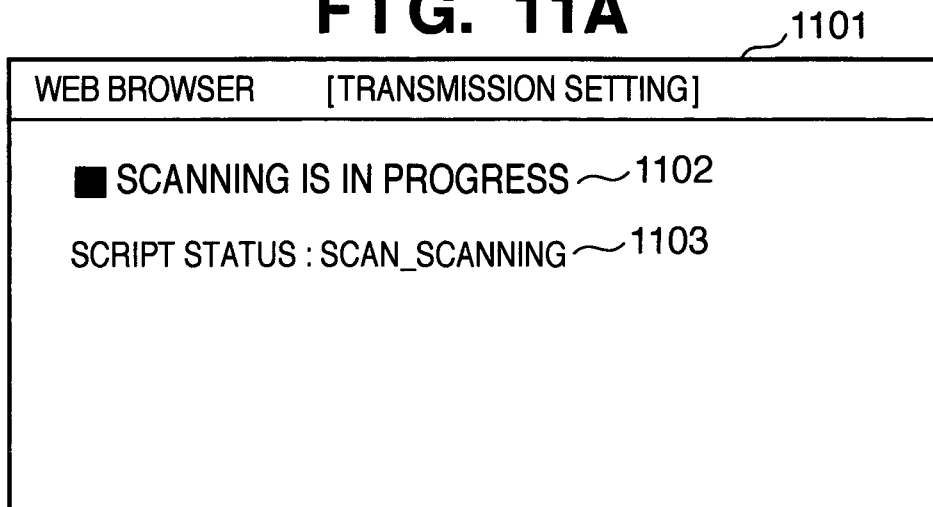
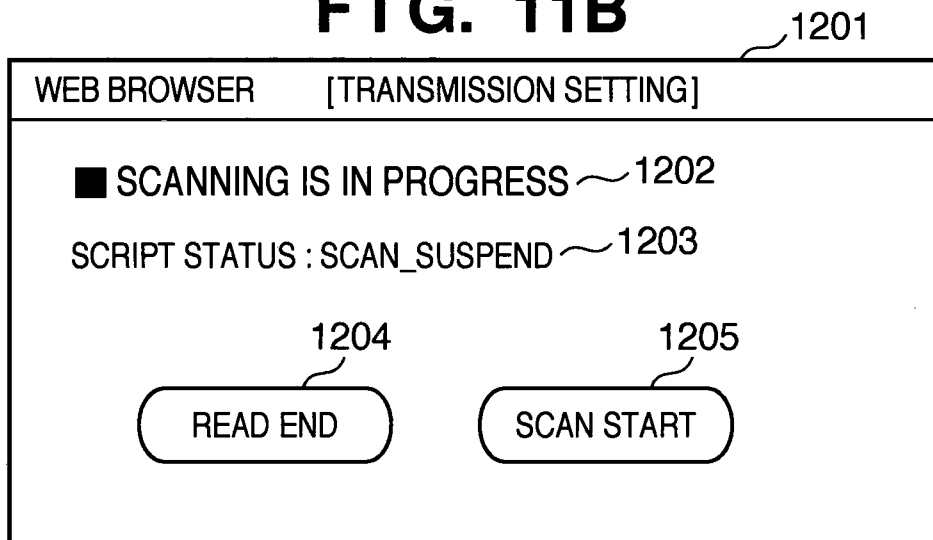
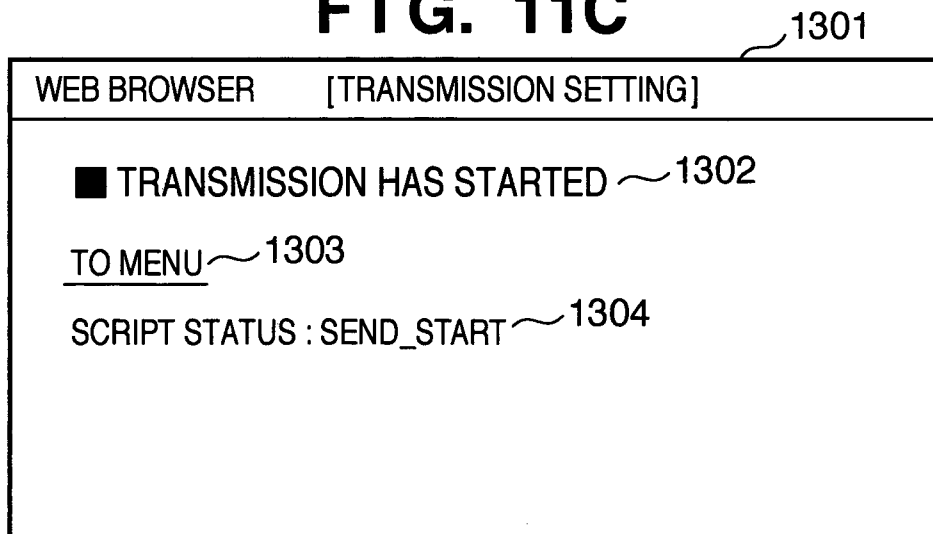
FIG. 11A**FIG. 11B****FIG. 11C**

FIG. 12

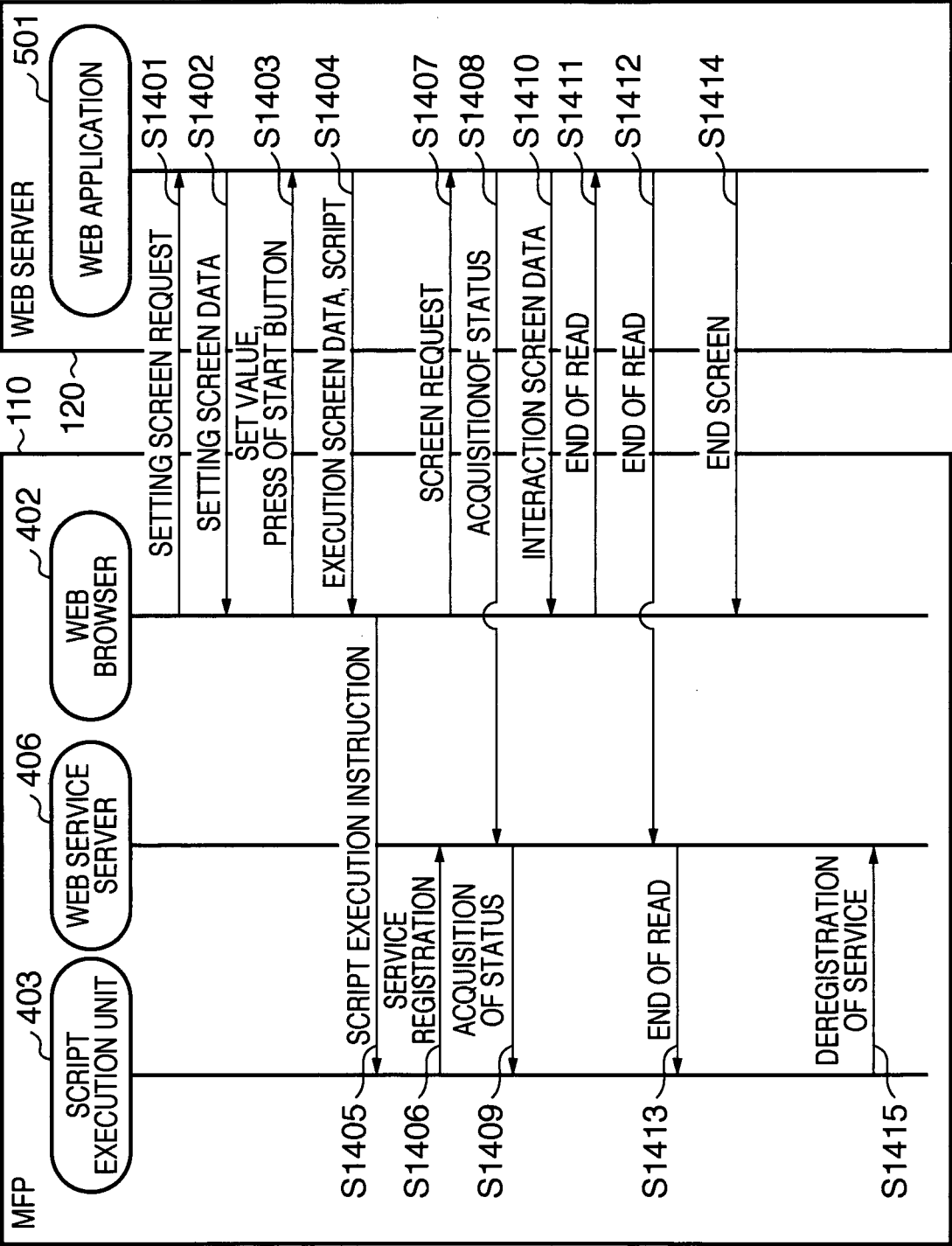


FIG. 13

```
<!DOCTYPE html ...>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>SCAN AND TRANSMISSION</title>
  <script type="text/JavaScript">
    <!--
    window.onload = function( ) {
      setTimeout(" location.reload( )", 2000);
    }
    -->
  </script>
</head>
<body>
  <form id="form1" action=" "method="post">
    <div id="message_scan" >
      <h1>SCANNING IS IN PROGRESS</h1>
    </div>
    SCRIPT STATUS : SCAN_SCANNING
  </form>
</body>
</html>
```

} 1510

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/062875

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. G06F13/00 (2006.01) i, G06F3/12 (2006.01) i, G06F3/153 (2006.01) i,
H04N1/00 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl. G06F13/00, G06F3/12, G06F3/153, H04N1/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan 1922-1996
Published unexamined utility model applications of Japan 1971-2011
Registered utility model specifications of Japan 1996-2011
Published registered utility model applications of Japan 1994-2011

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 2008-003833 A (R i c o h C o m p a n y) 2008.01.10, 【0030】 - 【0079】 , Fig4-7 & CN 101094281 A	1-10



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

28.06.2011

Date of mailing of the international search report

05.07.2011

Name and mailing address of the ISA/JP

Japan Patent Office

3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan

Authorized officer

Ryuichi TANOUE

Telephone No. +81-3-3581-1101 Ext. 3565

51

4176