US 20150212926A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2015/0212926 A1

(43) Pub. Date: Jul. 30, 2015

MAEDA et al.

(54) APPARATUS, SYSTEM, AND METHOD OF ACTIVATION CONTROL, AND MEDIUM STORING ACTIVATION CONTROL PROGRAM

(71) Applicants: **Kaoru MAEDA**, Kanagawa (JP); **Yoichiro MATSUNO**, Kanagawa (JP)

(72) Inventors: **Kaoru MAEDA**, Kanagawa (JP); **Yoichiro MATSUNO**, Kanagawa (JP)

(21) Appl. No.: **14/601,845**

(22) Filed: **Jan. 21, 2015**

(30) **Foreign Application Priority Data**

Jan. 28, 2014 (JP) 2014-013315

Publication Classification

(51) **Int. Cl.**
G06F 11/36 (2006.01)

(52) **U.S. Cl.**
CPC *G06F 11/3636* (2013.01)

(57) ABSTRACT

An activation control apparatus stores first association information that associates, for each application, application identification information for identifying an application with terminal identification information for identifying a communication terminal permitted to debug the application. When a first communication terminal is provided with a debugger, the activation control apparatus determines whether any candidate application available for use by the first communication terminal is associated with terminal identification information of the first communication terminal using the first association information, and excludes from the candidate applications one or more applications that are not associated with the terminal identification information of the first communication terminal to generate an application list of one or more applications that can be activated. When the first communication terminal is not provided with a debugger, the activation control apparatus generates the application list that includes all of the one or more candidate applications as applications that can be activated.

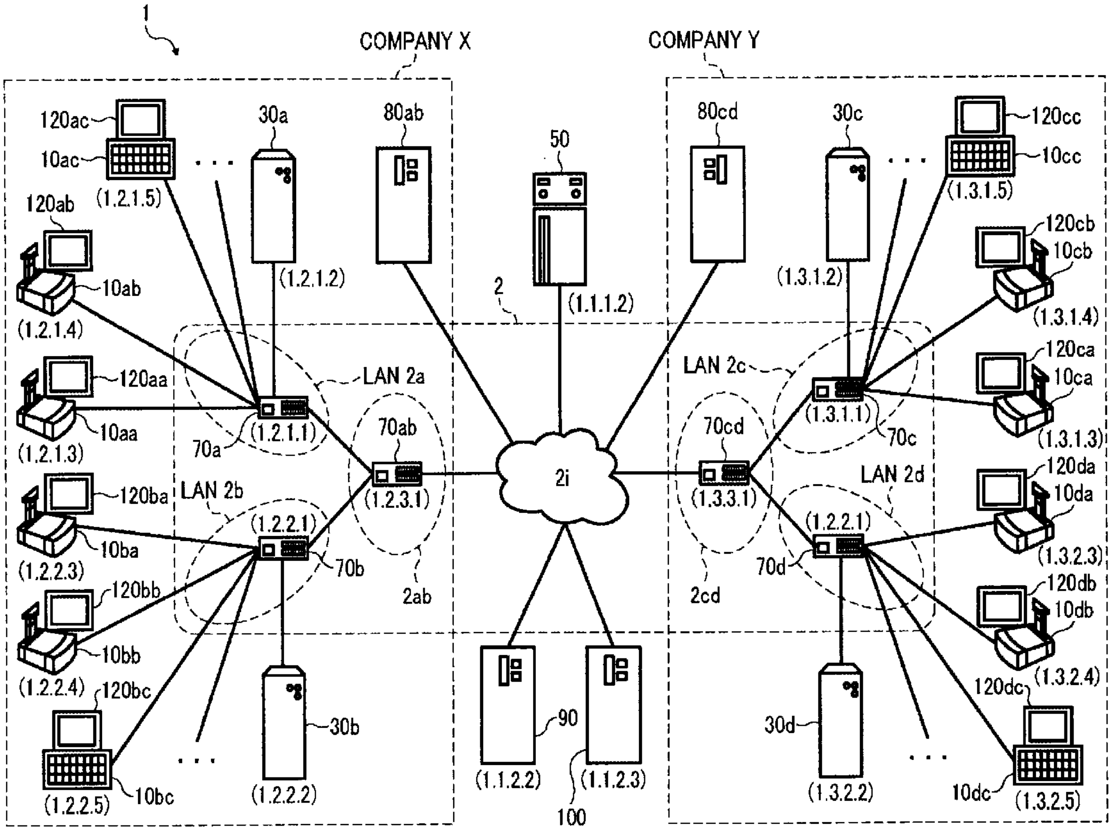


FIG. 1

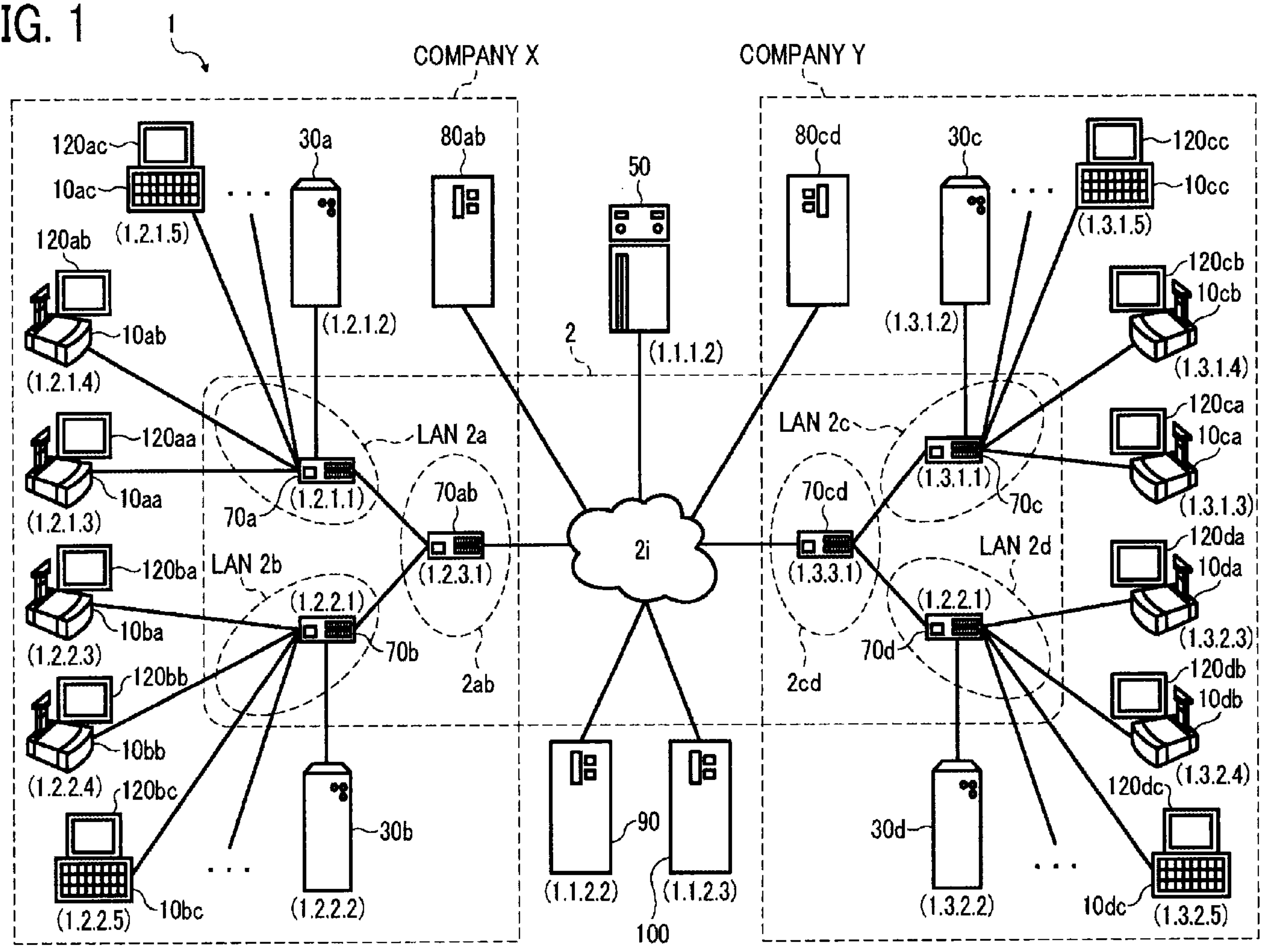

1
COMPANY X
COMPANY Y
120ac
10ac
(1.2.1.5)
30a
80ab
50
80cd
30c
120cc
10cc
(1.3.1.5)
120ab
10ab
(1.2.1.4)
(1.2.1.2)
2
(1.1.1.2)
(1.3.1.2)
120cb
10cb
(1.3.1.4)
120aa
10aa
(1.2.1.3)
LAN 2a
70a
(1.2.1.1)
70ab
(1.2.3.1)
2i
70cd
(1.3.3.1)
LAN 2c
(1.3.1.1)
70c
120ca
10ca
(1.3.1.3)
120ba
10ba
(1.2.2.3)
LAN 2b
(1.2.2.1)
70b
2ab
2cd
70d
(1.2.2.1)
LAN 2d
120da
10da
(1.3.2.3)
120bb
10bb
(1.2.2.4)
120db
10db
(1.3.2.4)
120bc
10bc
(1.2.2.5)
30b
(1.2.2.2)
90
(1.1.2.2)
(1.1.2.3)
100
30d
(1.3.2.2)
120dc
10dc
(1.3.2.5)

FIG. 2

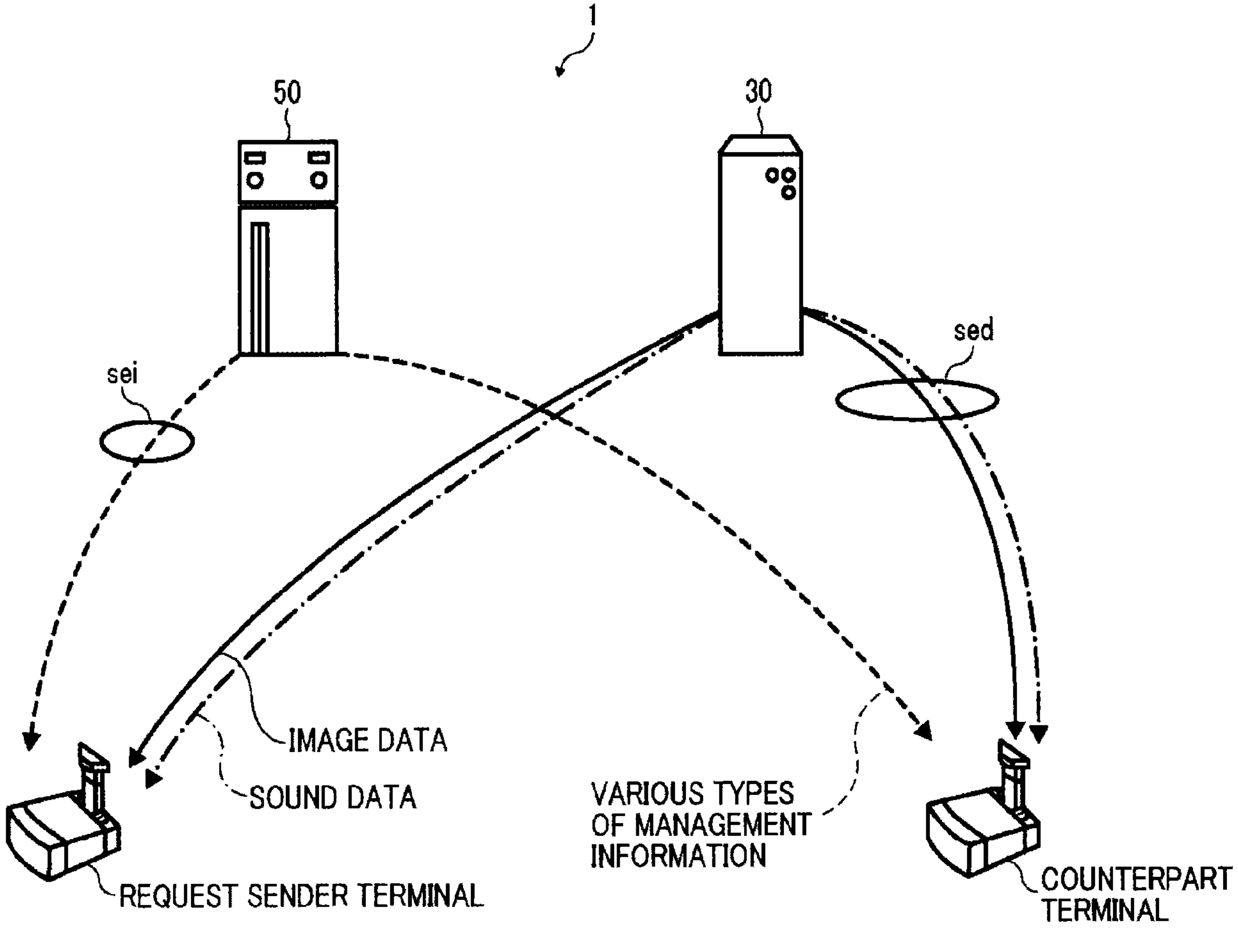

1
50
30
sei
sed
IMAGE DATA
SOUND DATA
VARIOUS TYPES OF MANAGEMENT INFORMATION
REQUEST SENDER TERMINAL
COUNTERPART TERMINAL

FIG. 3

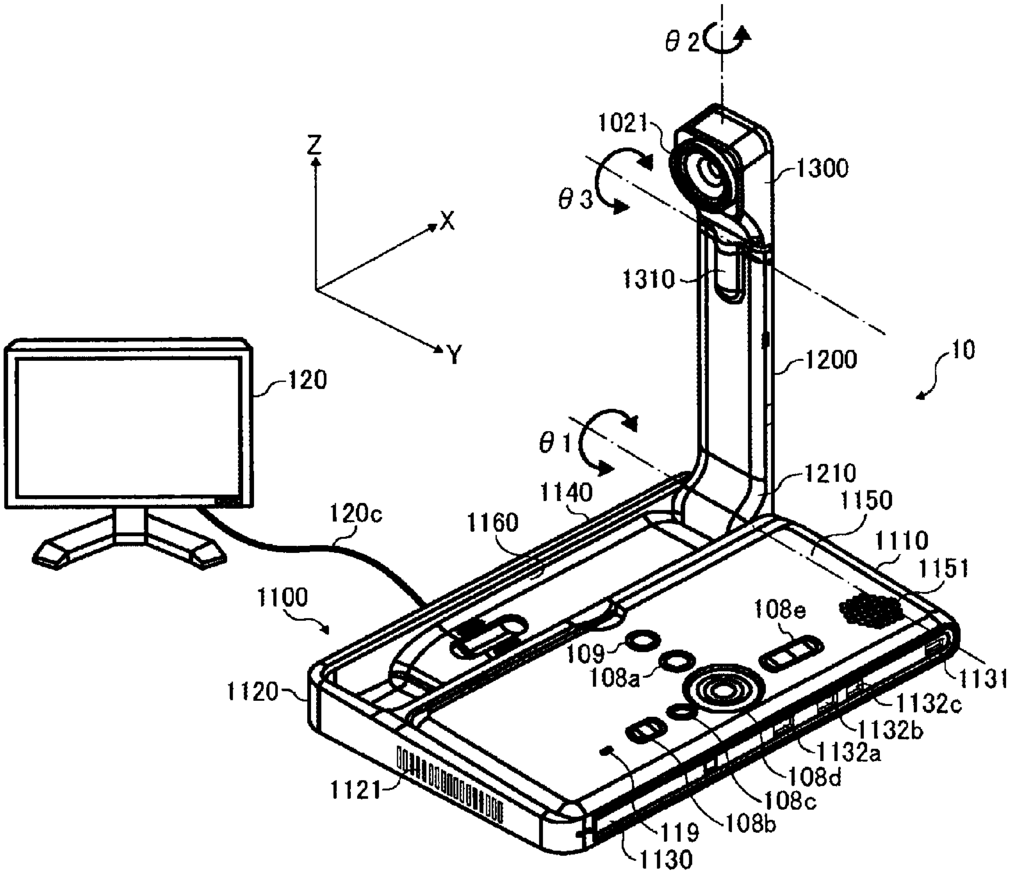

θ2
1021
1300
Z
X
θ3
1310
Y
120
1200
10
θ1
1210
120c
1140
1160
1150
1110
1151
1100
108e
109
108a
1131
1120
1132c
1132b
1132a
108d
108c
1121
119
108b
1130

FIG. 4

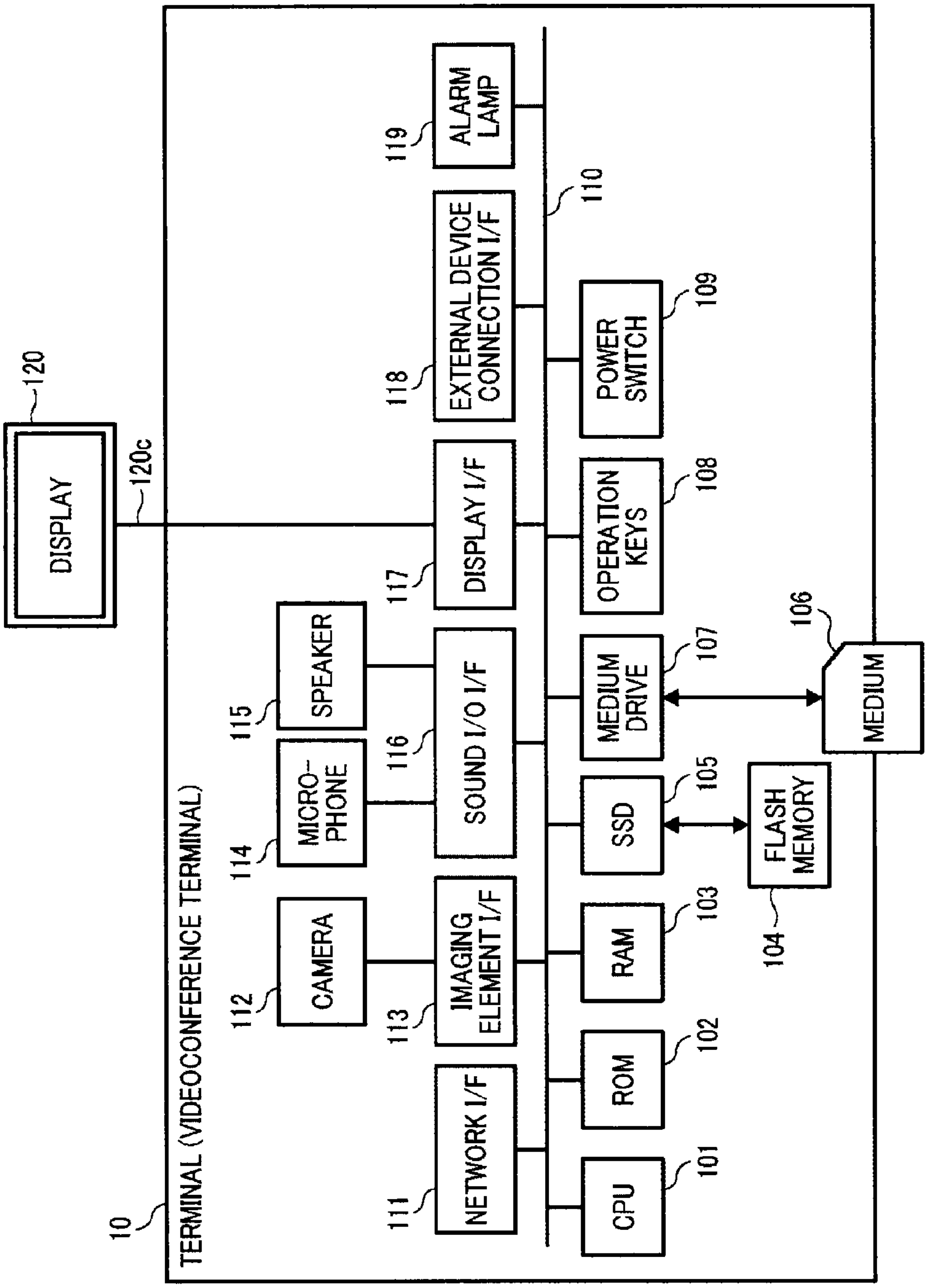

10
TERMINAL (VIDEOCONFERENCE TERMINAL)
120
DISPLAY
120c
111
NETWORK I/F
112
CAMERA
113
IMAGING ELEMENT I/F
114
MICRO-PHONE
115
SPEAKER
116
SOUND I/O I/F
117
DISPLAY I/F
118
EXTERNAL DEVICE CONNECTION I/F
119
ALARM LAMP
110
101
CPU
102
ROM
103
RAM
104
FLASH MEMORY
105
SSD
106
MEDIUM
107
MEDIUM DRIVE
108
OPERATION KEYS
109
POWER SWITCH

FIG. 5

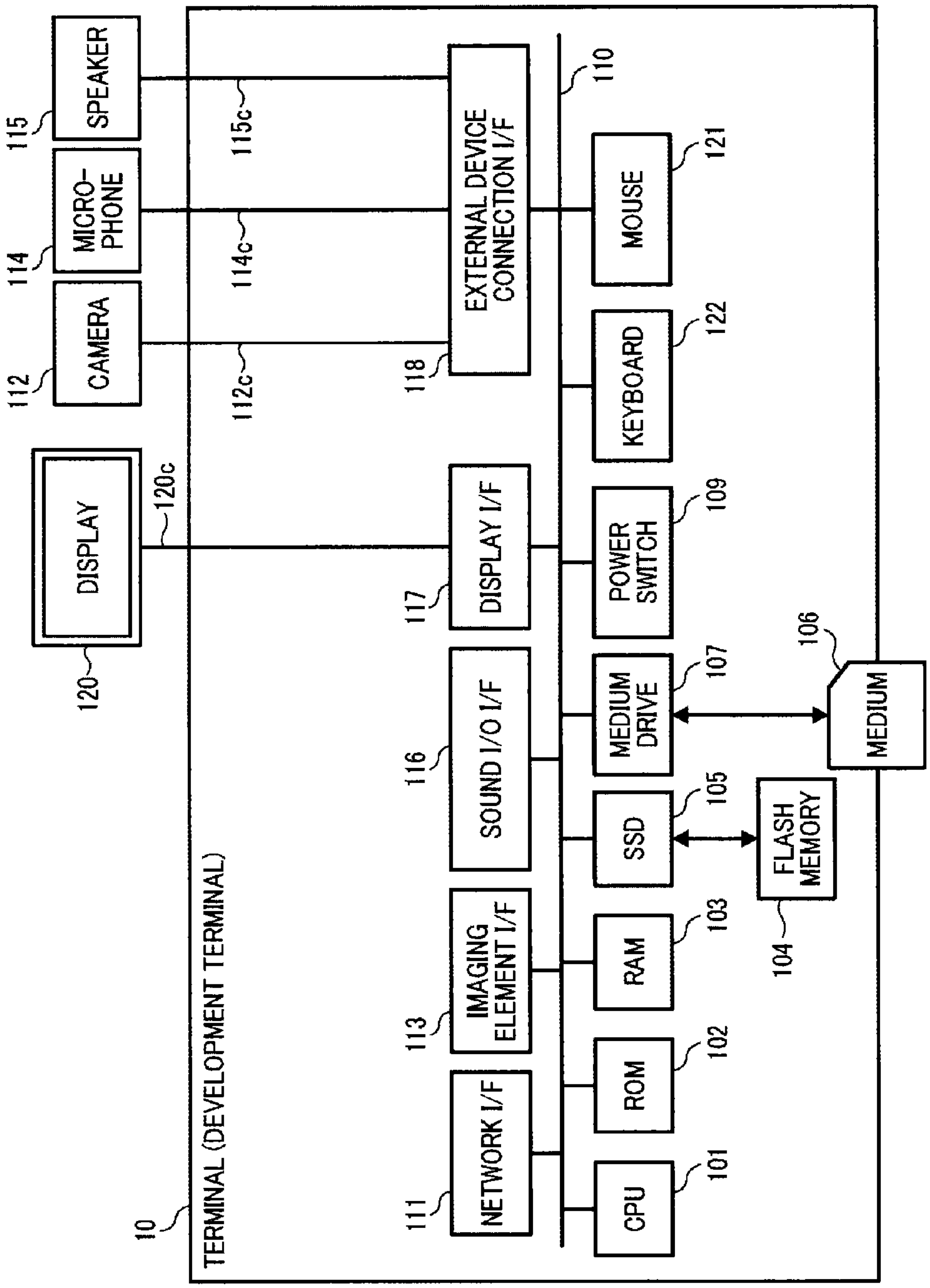

10
TERMINAL (DEVELOPMENT TERMINAL)
120
DISPLAY
120c
112
CAMERA
114
MICRO-PHONE
115
SPEAKER
112c
114c
115c
111
NETWORK I/F
113
IMAGING ELEMENT I/F
116
SOUND I/O I/F
117
DISPLAY I/F
118
EXTERNAL DEVICE CONNECTION I/F
110
CPU
101
ROM
102
RAM
103
SSD
105
FLASH MEMORY
104
MEDIUM DRIVE
107
MEDIUM
106
POWER SWITCH
109
KEYBOARD
122
MOUSE
121

FIG. 6

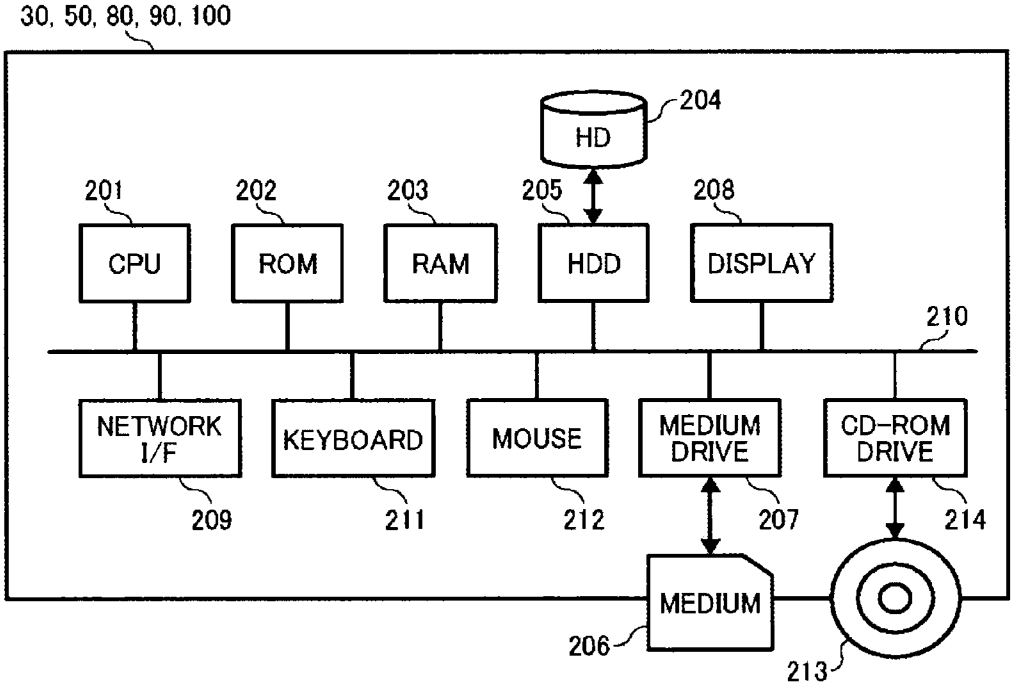

30, 50, 80, 90, 100
HD
204
201
202
203
205
208
CPU
ROM
RAM
HDD
DISPLAY
210
NETWORK I/F
KEYBOARD
MOUSE
MEDIUM DRIVE
CD-ROM DRIVE
209
211
212
207
214
MEDIUM
206
213

FIG. 7

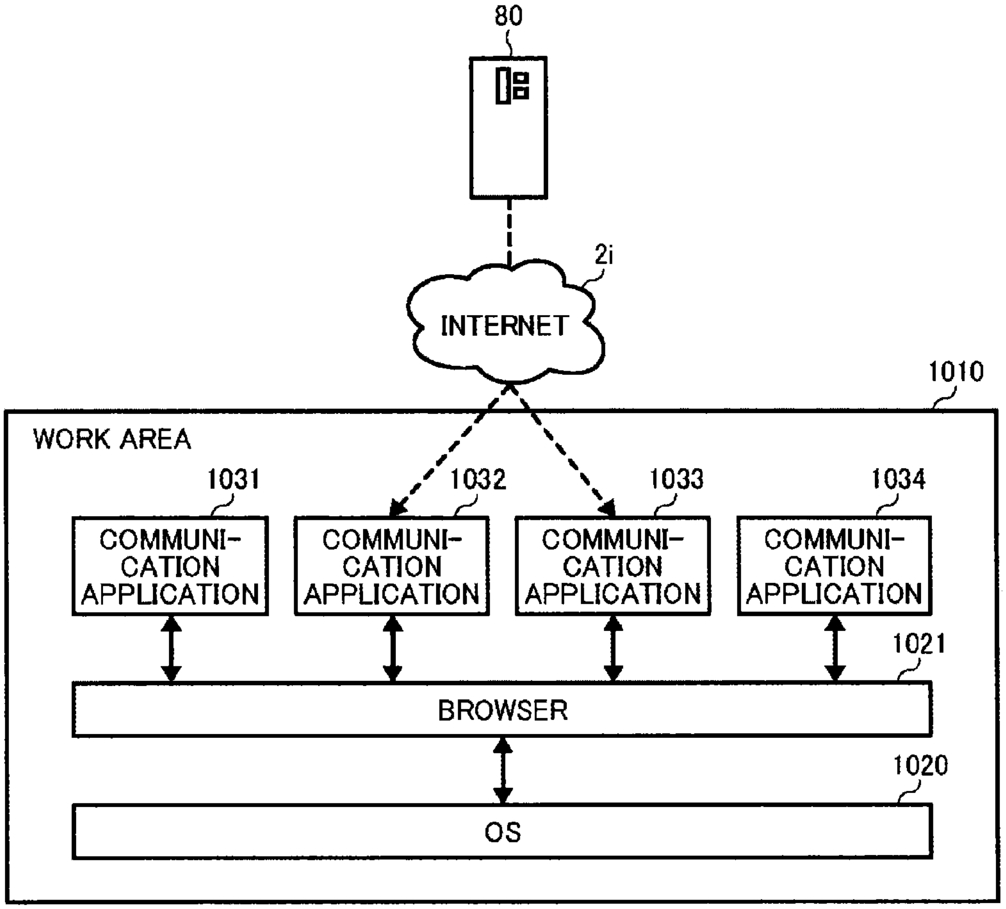

80
2i
INTERNET
1010
WORK AREA
1031
1032
1033
1034
COMMUNICATION APPLICATION
COMMUNICATION APPLICATION
COMMUNICATION APPLICATION
COMMUNICATION APPLICATION
1021
BROWSER
1020
OS

FIG. 8

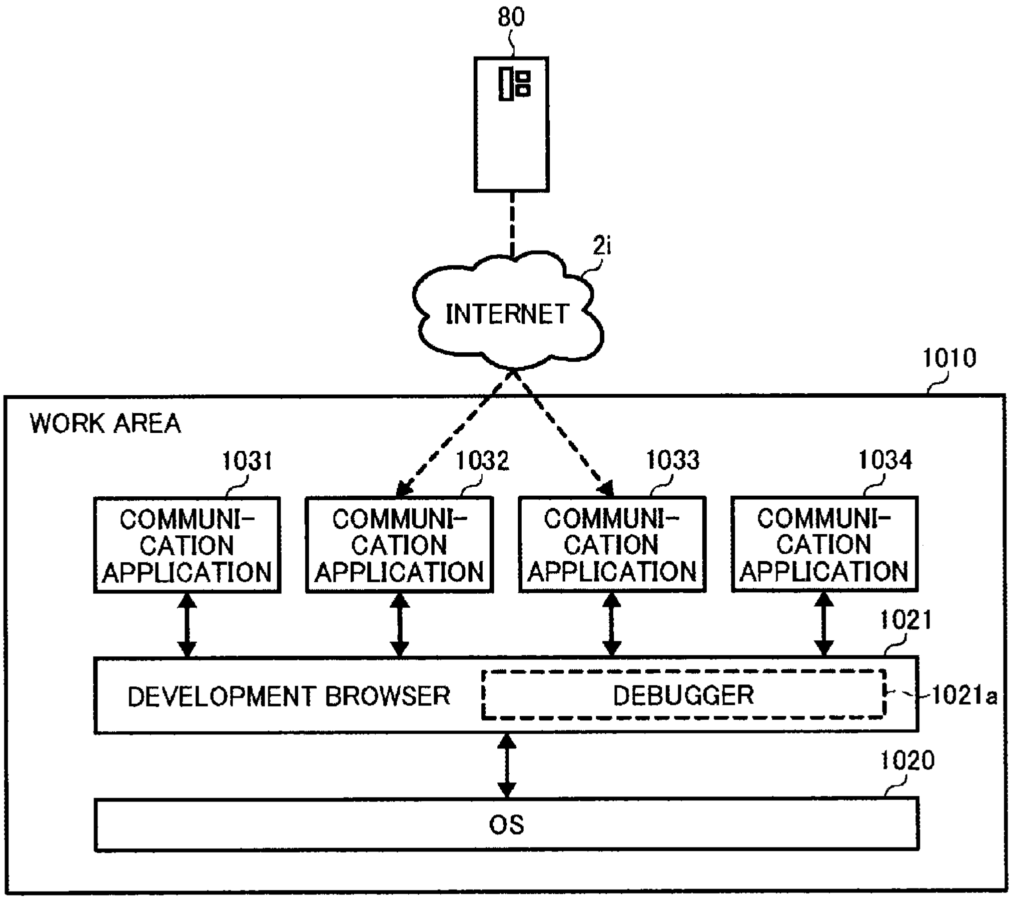

80
2i
INTERNET
1010
WORK AREA
1031
1032
1033
1034
COMMUNICATION APPLICATION
COMMUNICATION APPLICATION
COMMUNICATION APPLICATION
COMMUNICATION APPLICATION
1021
DEVELOPMENT BROWSER
DEBUGGER
1021a
1020
OS

FIG. 9

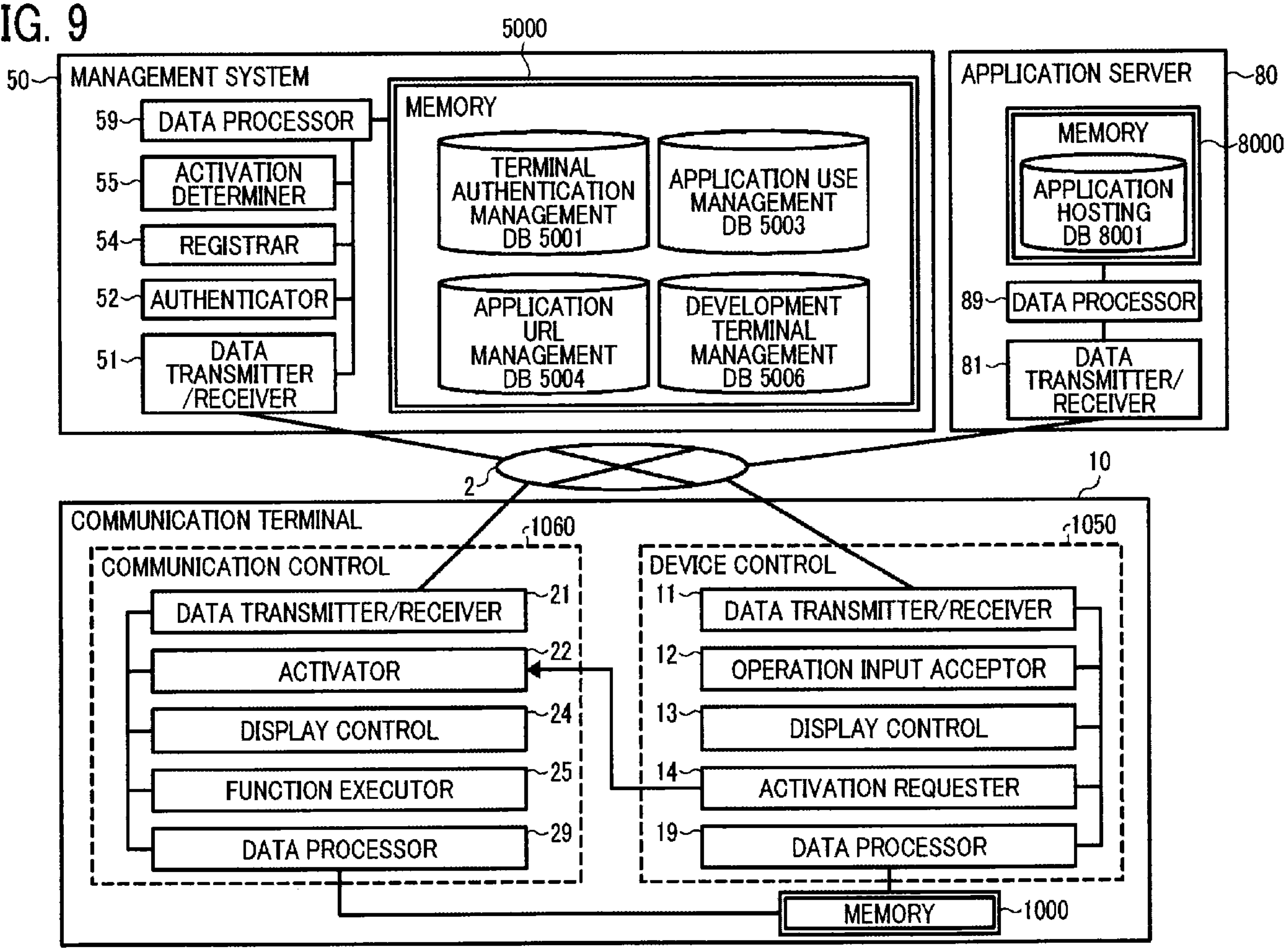

50
MANAGEMENT SYSTEM
59
DATA PROCESSOR
55
ACTIVATION DETERMINER
54
REGISTRAR
52
AUTHENTICATOR
51
DATA TRANSMITTER /RECEIVER
5000
MEMORY
TERMINAL AUTHENTICATION MANAGEMENT DB 5001
APPLICATION USE MANAGEMENT DB 5003
APPLICATION URL MANAGEMENT DB 5004
DEVELOPMENT TERMINAL MANAGEMENT DB 5006
APPLICATION SERVER
80
MEMORY
8000
APPLICATION HOSTING DB 8001
89
DATA PROCESSOR
81
DATA TRANSMITTER/ RECEIVER
2
10
COMMUNICATION TERMINAL
1060
COMMUNICATION CONTROL
DATA TRANSMITTER/RECEIVER
21
ACTIVATOR
22
DISPLAY CONTROL
24
FUNCTION EXECUTOR
25
DATA PROCESSOR
29
1050
DEVICE CONTROL
11
DATA TRANSMITTER/RECEIVER
12
OPERATION INPUT ACCEPTOR
13
DISPLAY CONTROL
14
ACTIVATION REQUESTER
19
DATA PROCESSOR
MEMORY
1000

FIG. 10

TERMINAL AUTHENTICATION MANAGEMENT TABLE

TERMINAL ID	PASSWORD
01aa	aaaa
01ab	abab
01ba	baba
...	...

FIG. 11

APPLICATION USE MANAGEMENT TABLE

TERMINAL ID	APPLICATION ID	EFFECTIVE PERIOD START DAY	EFFECTIVE PERIOD END DAY
01aa	a010	2013.01.01	2013.12.31
01ab	a001	2013.08.01	2014.07.31
01ac	a010	2013.01.01	2014.07.31
...	...	...	...

FIG. 12

APPLICATION URL MANAGEMENT TABLE

APPLICATION ID	URL OF ICON	URL OF APPLICATION
a001	https://ucs.Ulricoh.com/aaa/aicon.jpg	https://ucs.Ulricoh.com/aaa/aaa.html
a002	https://ucs.Ulricoh.com/bbb/bicon.jpg	https://ucs.Ulricoh.com/bbb/bbb.html
a003	https://ucs.Ulricoh.com/ccc/cicon.jpg	https://ucs.Ulricoh.com/ccc/ccc.html
a004	https://ucs.Ulricoh.com/ddd/dicon.jpg	https://ucs.Ulricoh.com/ddd/ddd.html
...	...	...

FIG. 13

DEVELOPER NAME MANAGEMENT TABLE

DEVELOPER ID	NAME
dv01	COMPANY A
dv02	COMPANY B
dv03	COMPANY C
...	...

FIG. 14

APPLICATION DEVELOPER MANAGEMENT TABLE

APPLICATION ID	DEVELOPER ID
a010	dv01
a011	dv01
a012	dv02
a013	dv03
a014	dv02
a015	dv01
...	...

FIG. 15

DEVELOPMENT TERMINAL MANAGEMENT TABLE

TERMINAL ID	DEVELOPER ID
01ac	dv01
01bc	dv01
01cc	dv02
01dc	dv02
...	...

FIG. 16

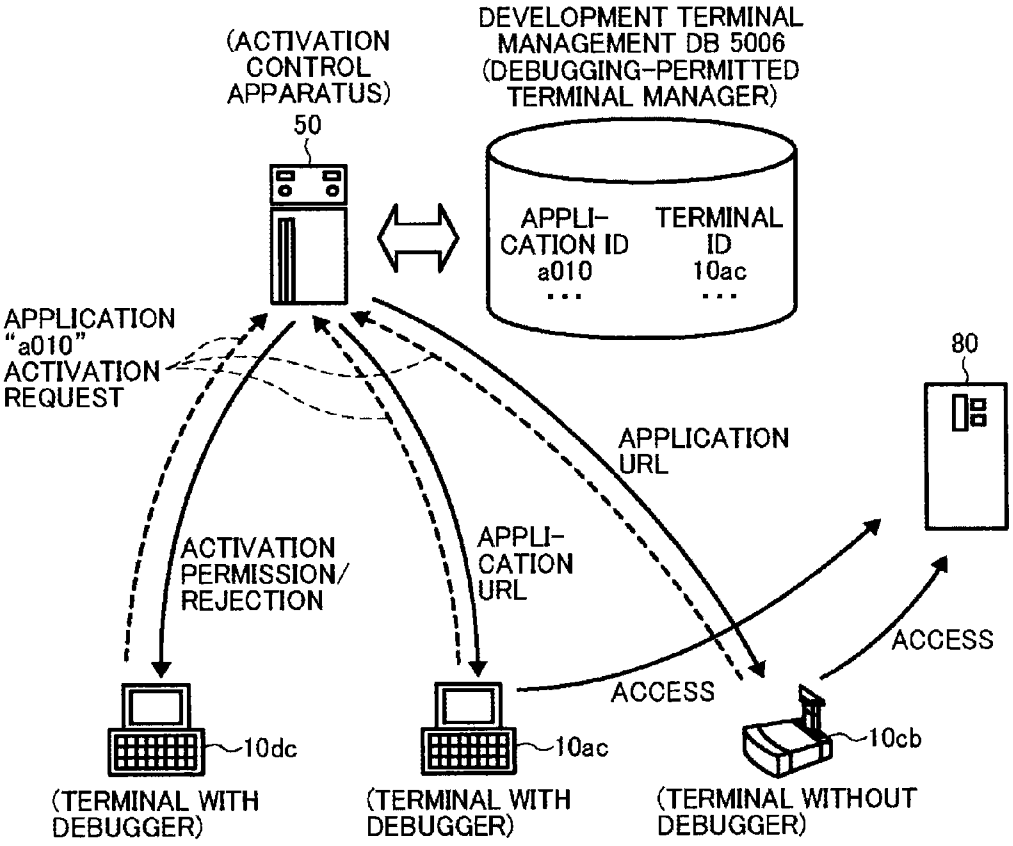

(ACTIVATION CONTROL APPARATUS)
50
DEVELOPMENT TERMINAL MANAGEMENT DB 5006 (DEBUGGING-PERMITTED TERMINAL MANAGER)
APPLI-CATION ID
TERMINAL ID
a010
10ac
...
...
APPLICATION "a010" ACTIVATION REQUEST
80
APPLICATION URL
ACTIVATION PERMISSION/ REJECTION
APPLI-CATION URL
ACCESS
ACCESS
10dc
10ac
10cb
(TERMINAL WITH DEBUGGER)
(TERMINAL WITH DEBUGGER)
(TERMINAL WITHOUT DEBUGGER)

FIG. 17

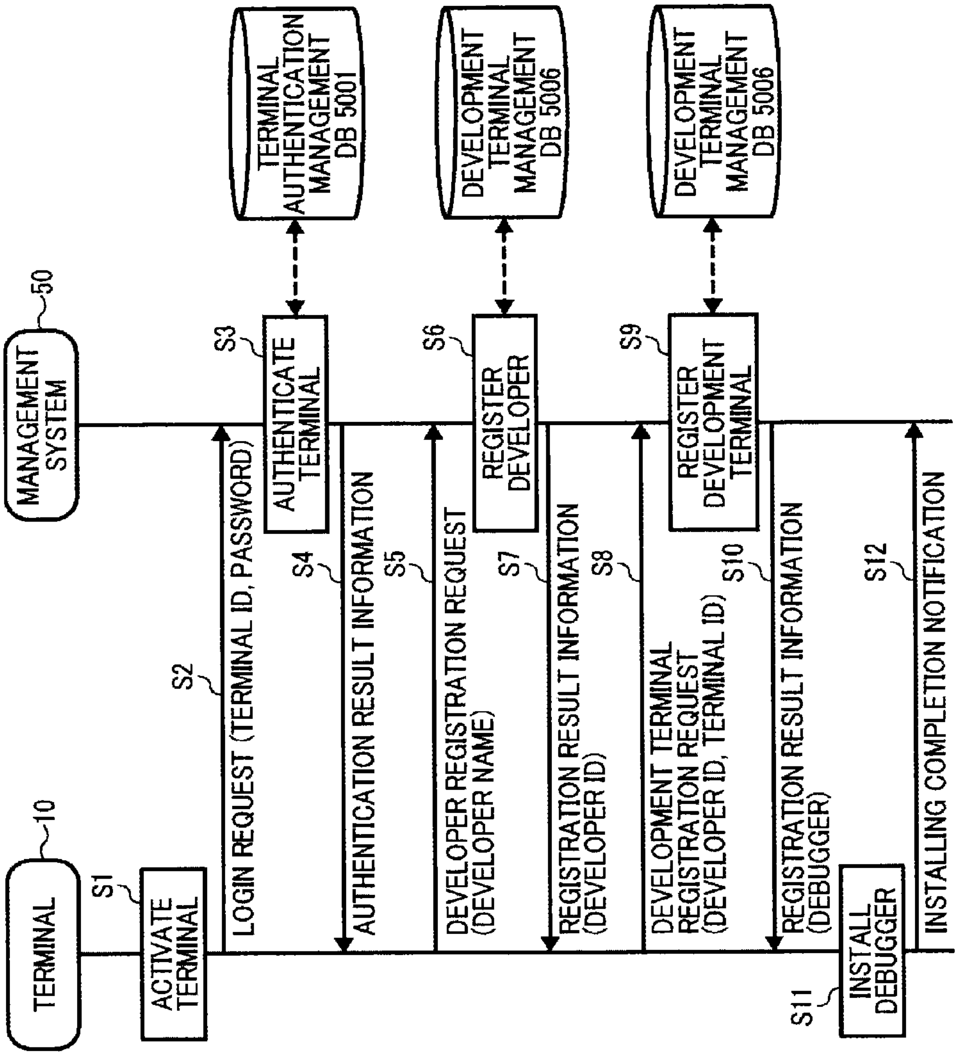

TERMINAL
10
MANAGEMENT SYSTEM
50
S1
ACTIVATE TERMINAL
S2
LOGIN REQUEST (TERMINAL ID, PASSWORD)
S3
AUTHENTICATE TERMINAL
TERMINAL AUTHENTICATION MANAGEMENT DB 5001
S4
AUTHENTICATION RESULT INFORMATION
S5
DEVELOPER REGISTRATION REQUEST (DEVELOPER NAME)
S6
REGISTER DEVELOPER
DEVELOPMENT TERMINAL MANAGEMENT DB 5006
S7
REGISTRATION RESULT INFORMATION (DEVELOPER ID)
S8
DEVELOPMENT TERMINAL REGISTRATION REQUEST (DEVELOPER ID, TERMINAL ID)
S9
REGISTER DEVELOPMENT TERMINAL
DEVELOPMENT TERMINAL MANAGEMENT DB 5006
S10
REGISTRATION RESULT INFORMATION (DEBUGGER)
S11
INSTALL DEBUGGER
S12
INSTALLING COMPLETION NOTIFICATION

FIG. 18

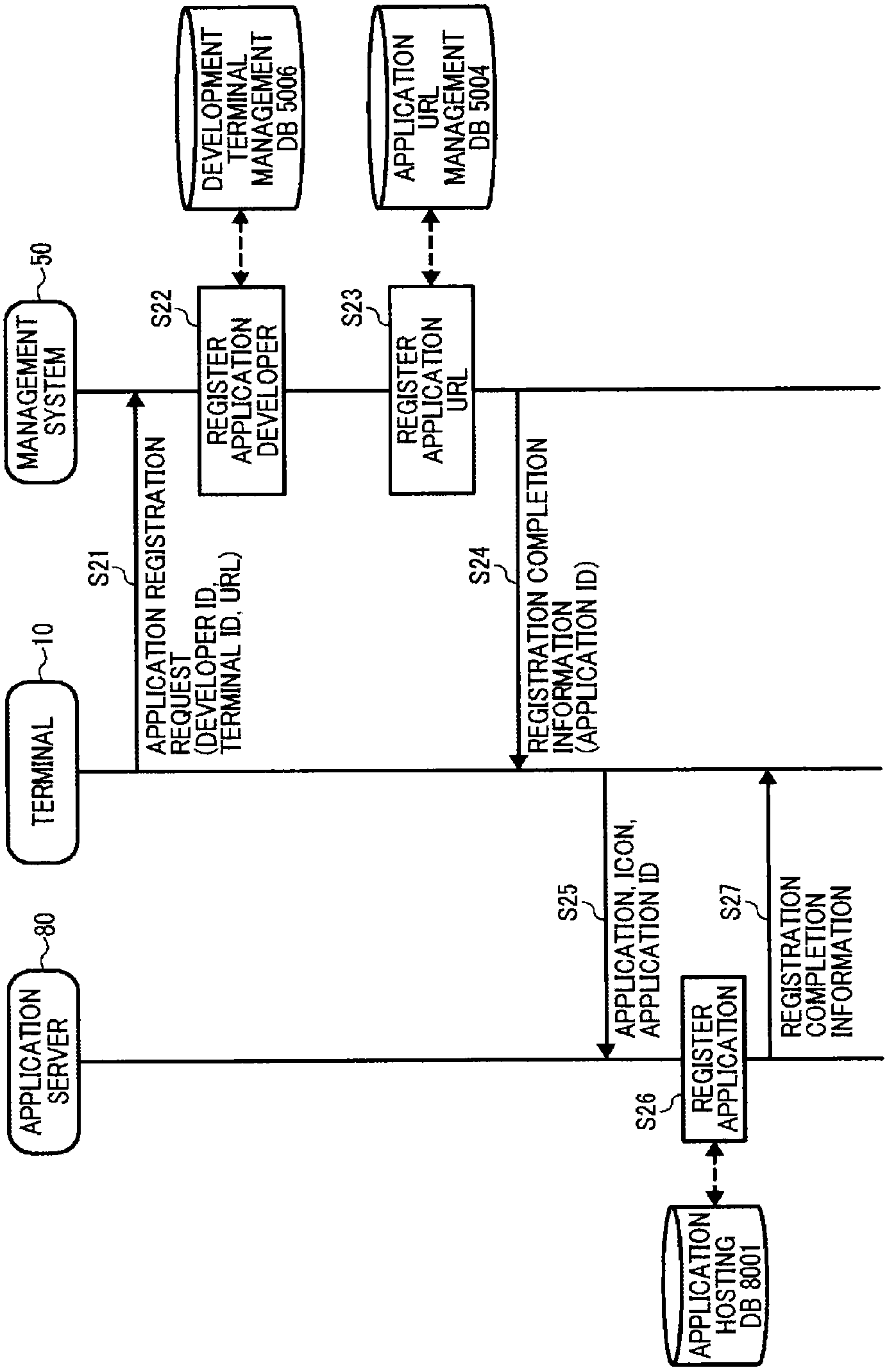

APPLICATION SERVER
80
TERMINAL
10
MANAGEMENT SYSTEM
50
S21
APPLICATION REGISTRATION REQUEST (DEVELOPER ID, TERMINAL ID, URL)
S22
REGISTER APPLICATION DEVELOPER
DEVELOPMENT TERMINAL MANAGEMENT DB 5006
S23
REGISTER APPLICATION URL
APPLICATION URL MANAGEMENT DB 5004
S24
REGISTRATION COMPLETION INFORMATION (APPLICATION ID)
S25
APPLICATION, ICON, APPLICATION ID
S26
REGISTER APPLICATION
APPLICATION HOSTING DB 8001
S27
REGISTRATION COMPLETION INFORMATION

FIG. 19

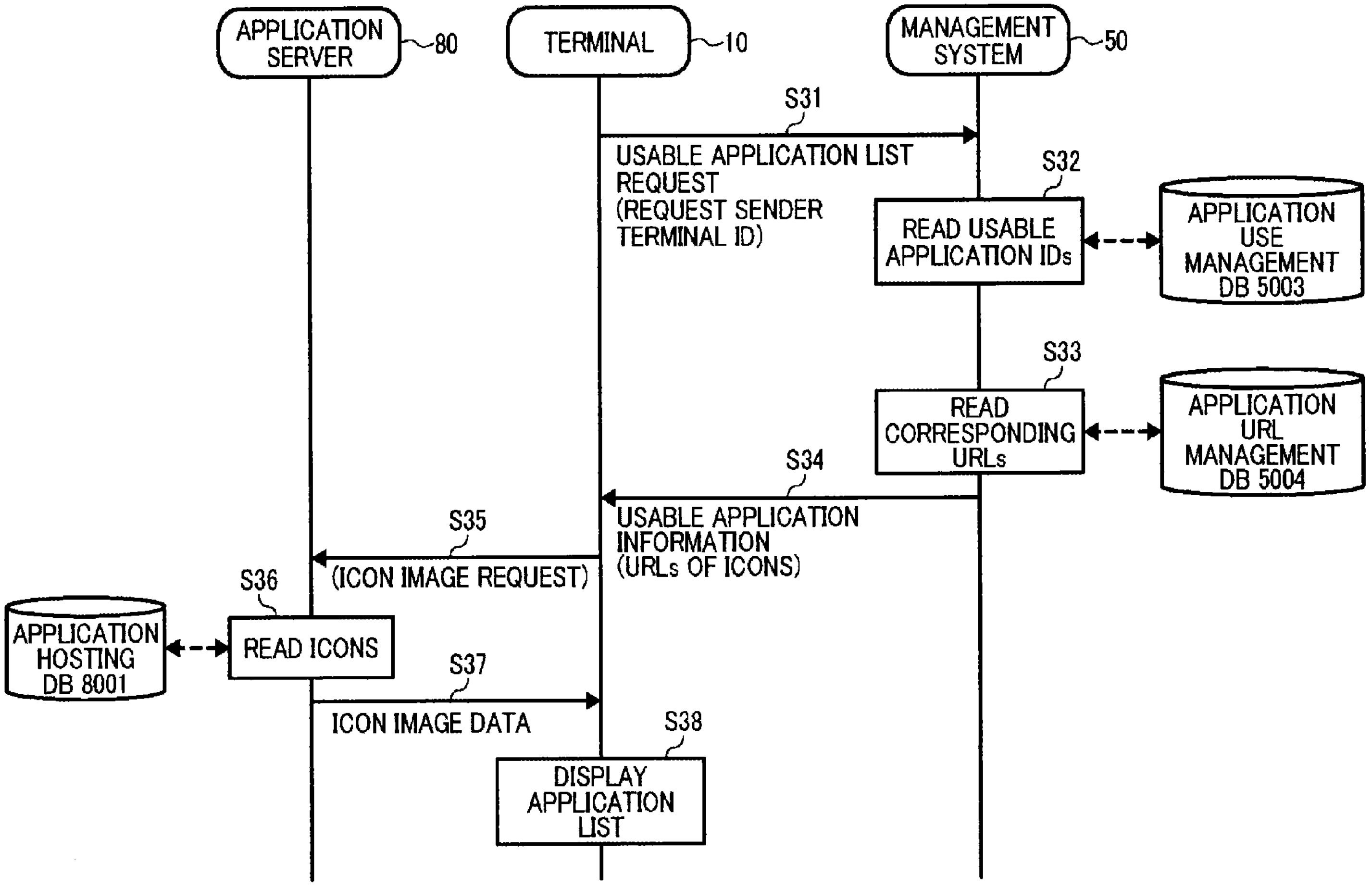

APPLICATION SERVER
80
TERMINAL
10
MANAGEMENT SYSTEM
50
S31
USABLE APPLICATION LIST REQUEST (REQUEST SENDER TERMINAL ID)
S32
READ USABLE APPLICATION IDs
APPLICATION USE MANAGEMENT DB 5003
S33
READ CORRESPONDING URLs
APPLICATION URL MANAGEMENT DB 5004
S34
USABLE APPLICATION INFORMATION (URLs OF ICONS)
S35
(ICON IMAGE REQUEST)
S36
READ ICONS
APPLICATION HOSTING DB 8001
S37
ICON IMAGE DATA
S38
DISPLAY APPLICATION LIST

FIG. 20

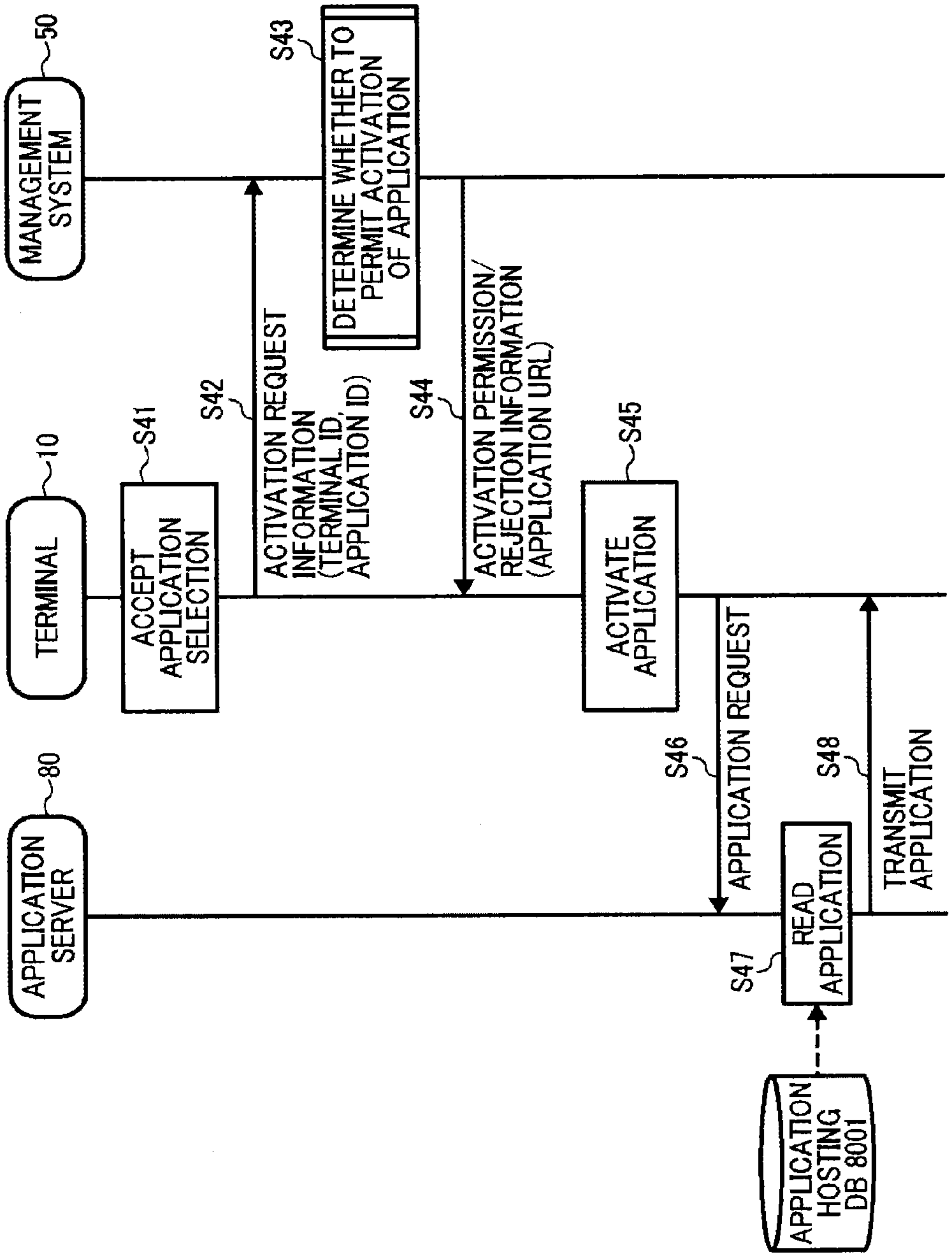

APPLICATION SERVER
80
TERMINAL
10
MANAGEMENT SYSTEM
50
S41
ACCEPT APPLICATION SELECTION
S42
ACTIVATION REQUEST INFORMATION (TERMINAL ID, APPLICATION ID)
S43
DETERMINE WHETHER TO PERMIT ACTIVATION OF APPLICATION
S44
ACTIVATION PERMISSION/ REJECTION INFORMATION (APPLICATION URL)
S45
ACTIVATE APPLICATION
S46
APPLICATION REQUEST
S47
READ APPLICATION
S48
TRANSMIT APPLICATION
APPLICATION HOSTING DB 8001

FIG. 21

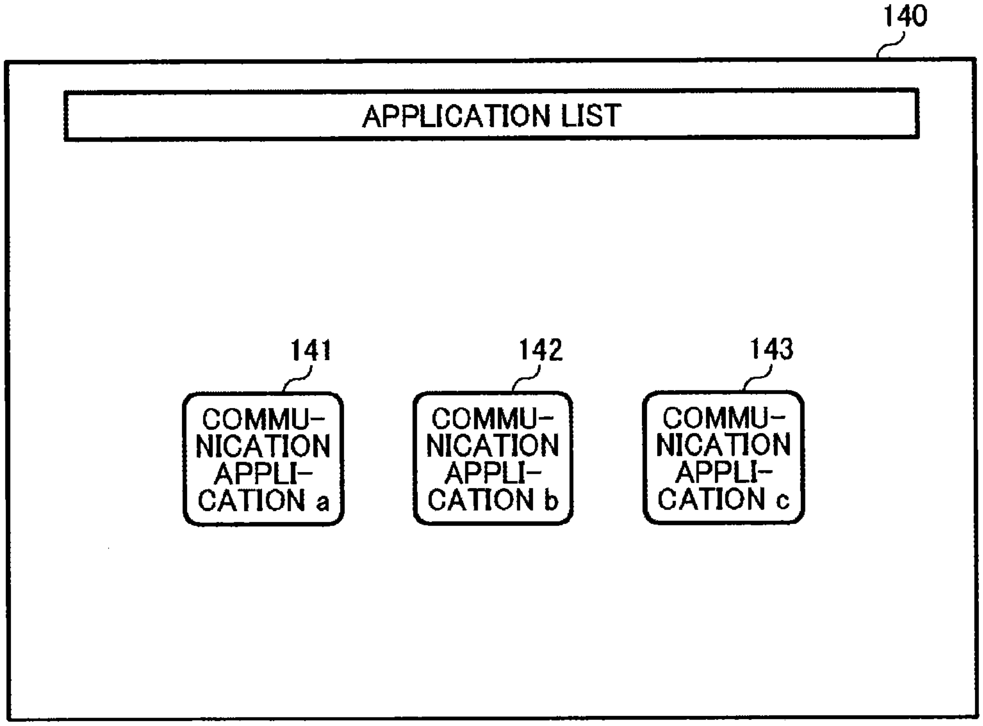

140
APPLICATION LIST
141
142
143
COMMU-NICATION APPLI-CATION a
COMMU-NICATION APPLI-CATION b
COMMU-NICATION APPLI-CATION c

FIG. 22

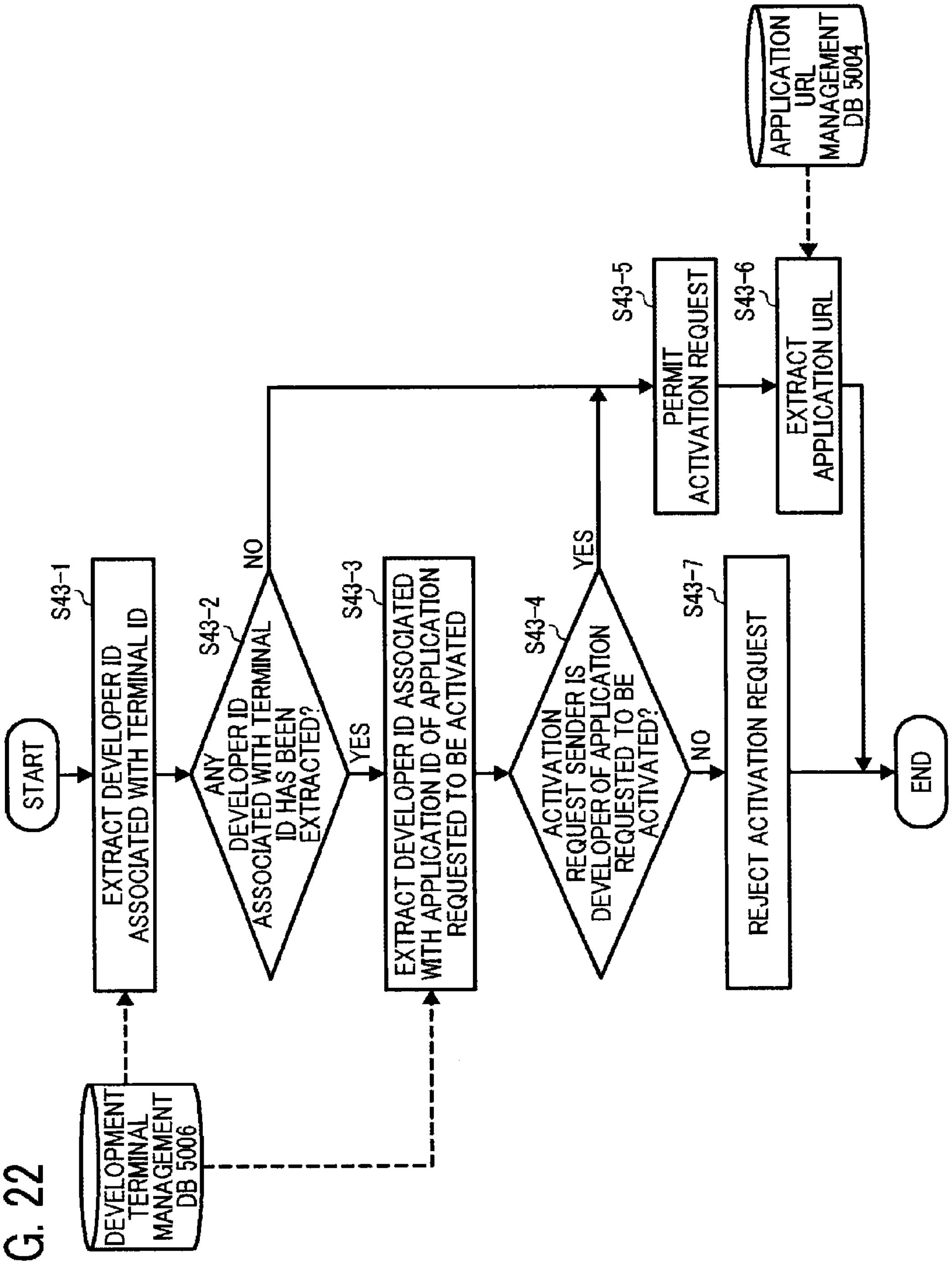

START
S43-1
EXTRACT DEVELOPER ID ASSOCIATED WITH TERMINAL ID
DEVELOPMENT TERMINAL MANAGEMENT DB 5006
S43-2
ANY DEVELOPER ID ASSOCIATED WITH TERMINAL ID HAS BEEN EXTRACTED?
YES
NO
S43-3
EXTRACT DEVELOPER ID ASSOCIATED WITH APPLICATION ID OF APPLICATION REQUESTED TO BE ACTIVATED
S43-4
ACTIVATION REQUEST SENDER IS DEVELOPER OF APPLICATION REQUESTED TO BE ACTIVATED?
YES
NO
S43-5
PERMIT ACTIVATION REQUEST
S43-6
EXTRACT APPLICATION URL
APPLICATION URL MANAGEMENT DB 5004
S43-7
REJECT ACTIVATION REQUEST
END

FIG. 23

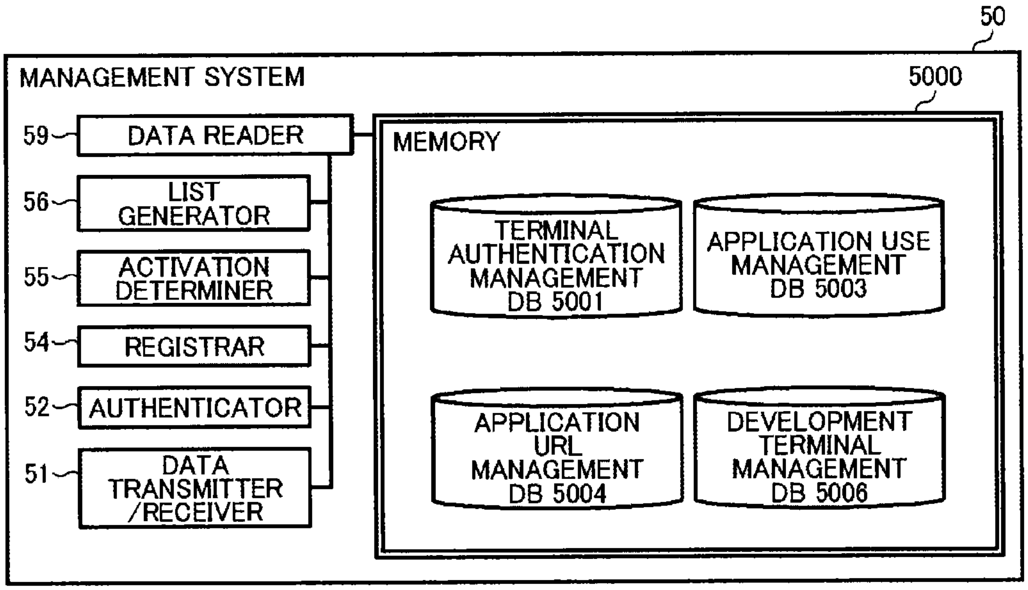

50
MANAGEMENT SYSTEM
5000
59
DATA READER
56
LIST GENERATOR
55
ACTIVATION DETERMINER
54
REGISTRAR
52
AUTHENTICATOR
51
DATA TRANSMITTER /RECEIVER
MEMORY
TERMINAL AUTHENTICATION MANAGEMENT DB 5001
APPLICATION USE MANAGEMENT DB 5003
APPLICATION URL MANAGEMENT DB 5004
DEVELOPMENT TERMINAL MANAGEMENT DB 5006

FIG. 24

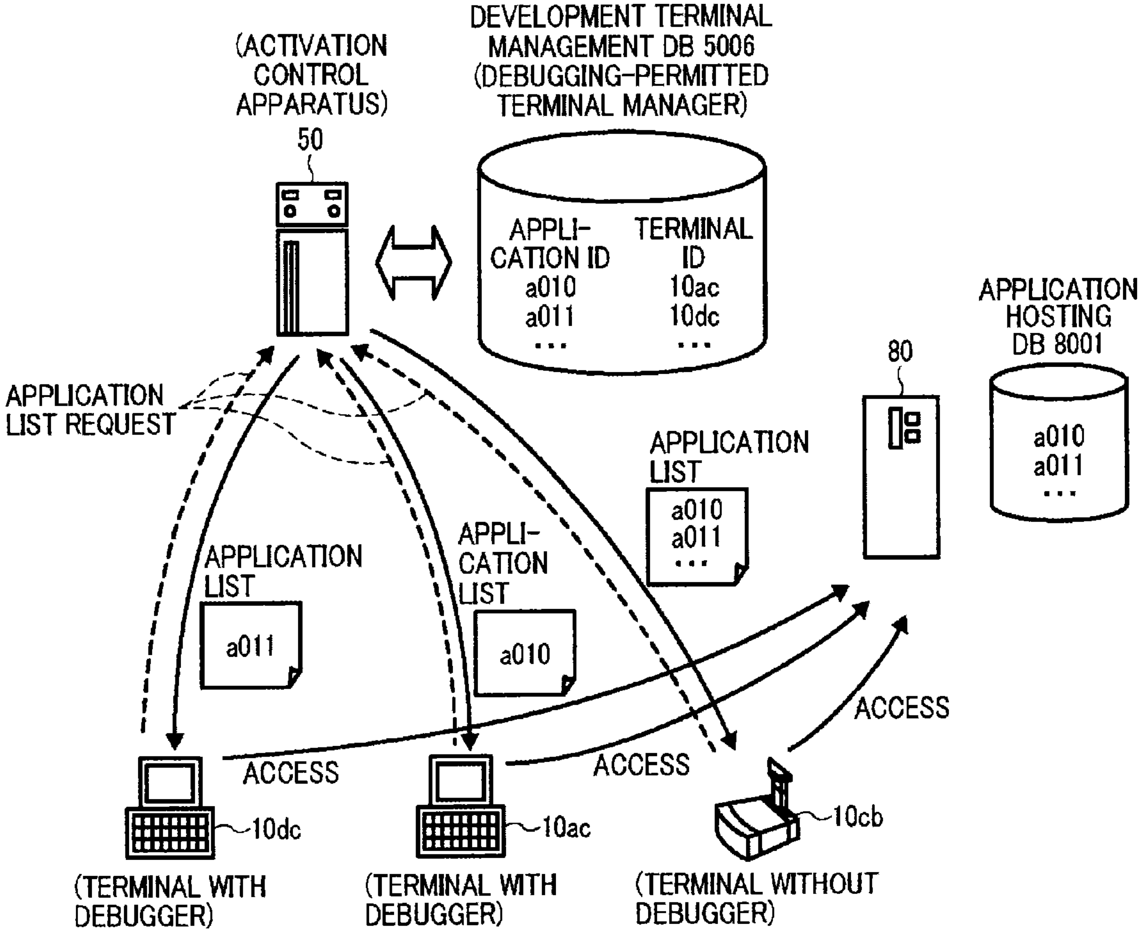

(ACTIVATION CONTROL APPARATUS)
50
DEVELOPMENT TERMINAL MANAGEMENT DB 5006 (DEBUGGING-PERMITTED TERMINAL MANAGER)
APPLI-CATION ID
TERMINAL ID
a010
10ac
a011
10dc
...
...
APPLICATION HOSTING DB 8001
80
a010
a011
...
APPLICATION LIST REQUEST
APPLICATION LIST
a011
APPLI-CATION LIST
a010
APPLICATION LIST
a010
a011
...
ACCESS
ACCESS
ACCESS
10dc
10ac
10cb
(TERMINAL WITH DEBUGGER)
(TERMINAL WITH DEBUGGER)
(TERMINAL WITHOUT DEBUGGER)

FIG. 25

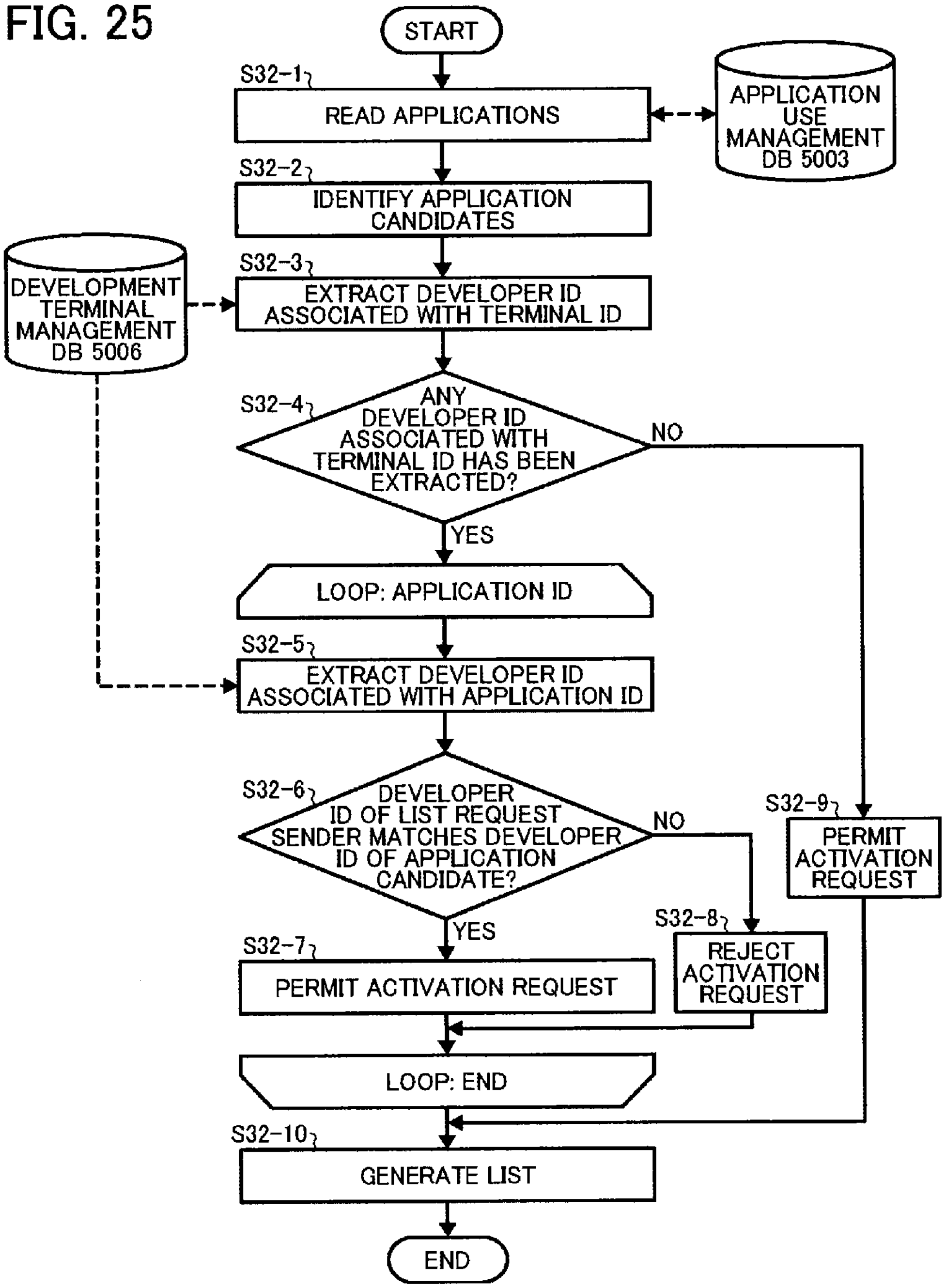

START
S32-1
READ APPLICATIONS
APPLICATION USE MANAGEMENT DB 5003
S32-2
IDENTIFY APPLICATION CANDIDATES
DEVELOPMENT TERMINAL MANAGEMENT DB 5006
S32-3
EXTRACT DEVELOPER ID ASSOCIATED WITH TERMINAL ID
S32-4
ANY DEVELOPER ID ASSOCIATED WITH TERMINAL ID HAS BEEN EXTRACTED?
NO
YES
LOOP: APPLICATION ID
S32-5
EXTRACT DEVELOPER ID ASSOCIATED WITH APPLICATION ID
S32-6
DEVELOPER ID OF LIST REQUEST SENDER MATCHES DEVELOPER ID OF APPLICATION CANDIDATE?
NO
YES
S32-9
PERMIT ACTIVATION REQUEST
S32-8
REJECT ACTIVATION REQUEST
S32-7
PERMIT ACTIVATION REQUEST
LOOP: END
S32-10
GENERATE LIST
END

FIG. 26

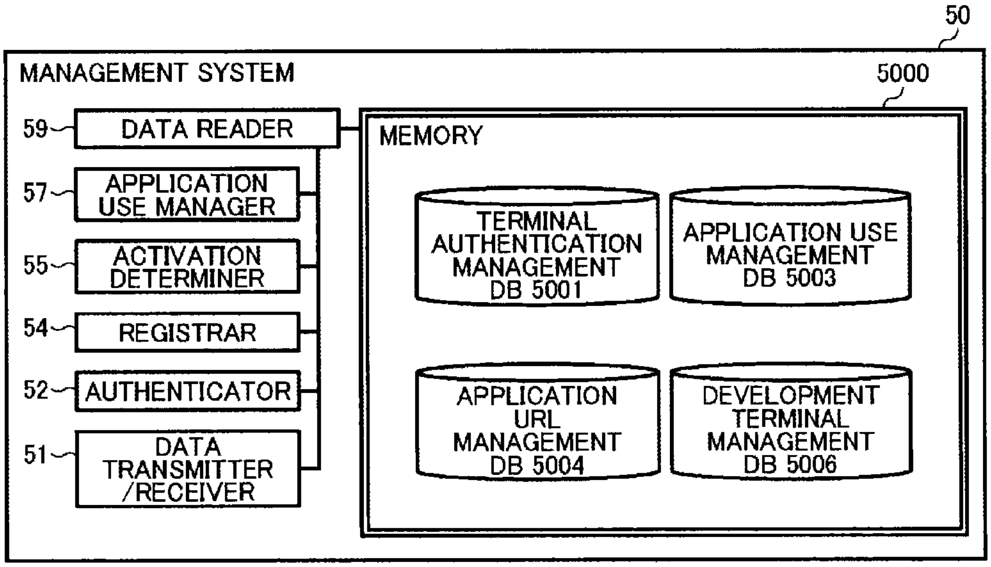

50
MANAGEMENT SYSTEM
5000
59
DATA READER
MEMORY
57
APPLICATION USE MANAGER
55
ACTIVATION DETERMINER
54
REGISTRAR
52
AUTHENTICATOR
51
DATA TRANSMITTER /RECEIVER
TERMINAL AUTHENTICATION MANAGEMENT DB 5001
APPLICATION USE MANAGEMENT DB 5003
APPLICATION URL MANAGEMENT DB 5004
DEVELOPMENT TERMINAL MANAGEMENT DB 5006

FIG. 27

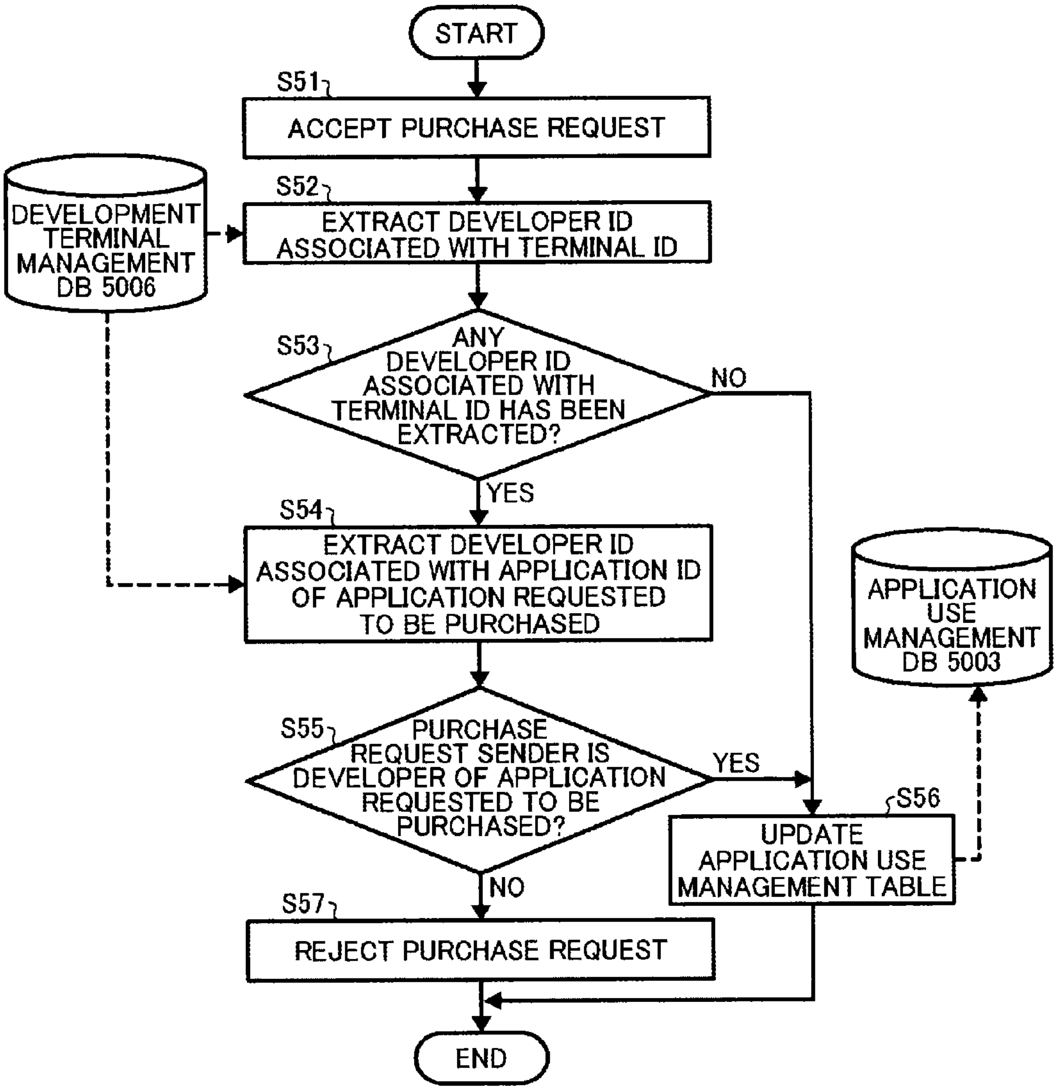

START
S51
ACCEPT PURCHASE REQUEST
DEVELOPMENT TERMINAL MANAGEMENT DB 5006
S52
EXTRACT DEVELOPER ID ASSOCIATED WITH TERMINAL ID
S53
ANY DEVELOPER ID ASSOCIATED WITH TERMINAL ID HAS BEEN EXTRACTED?
NO
YES
S54
EXTRACT DEVELOPER ID ASSOCIATED WITH APPLICATION ID OF APPLICATION REQUESTED TO BE PURCHASED
APPLICATION USE MANAGEMENT DB 5003
S55
PURCHASE REQUEST SENDER IS DEVELOPER OF APPLICATION REQUESTED TO BE PURCHASED?
YES
S56
UPDATE APPLICATION USE MANAGEMENT TABLE
NO
S57
REJECT PURCHASE REQUEST
END

FIG. 28

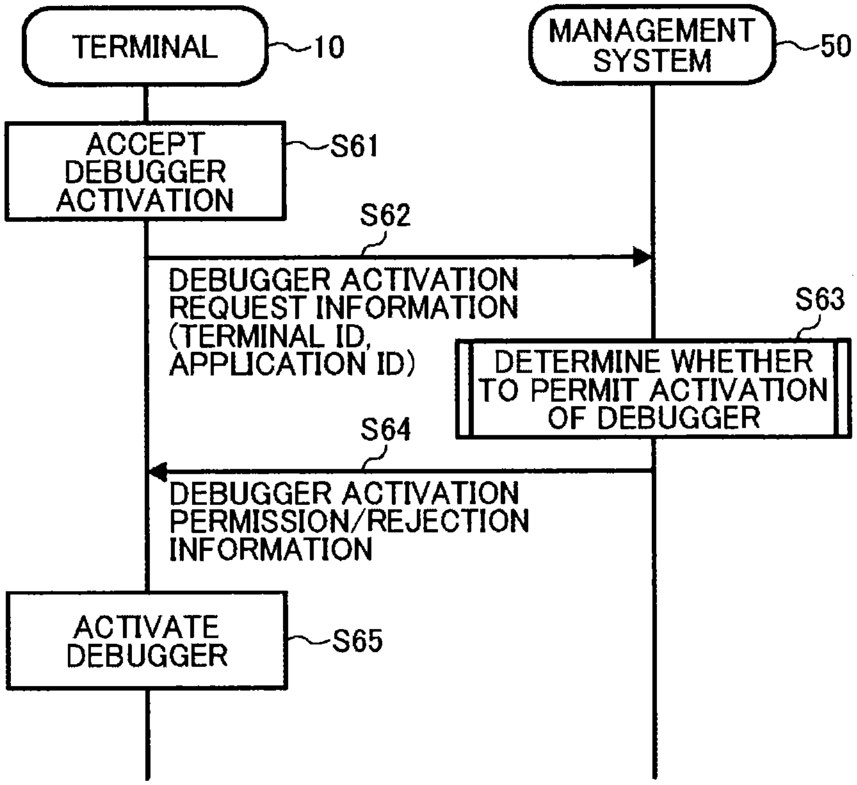
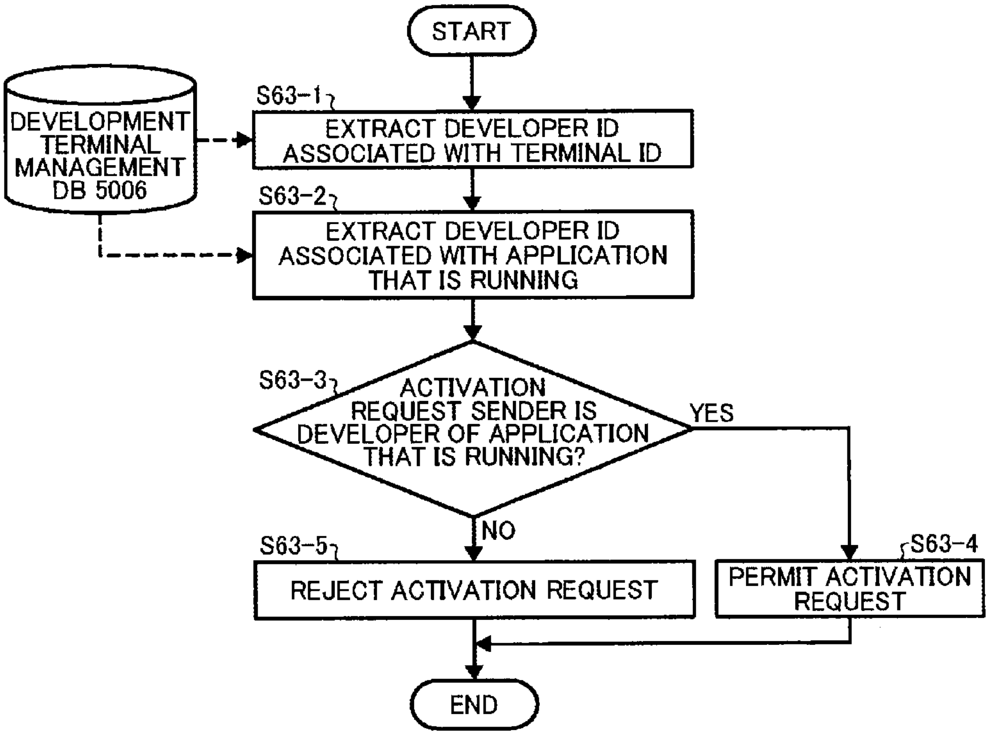

TERMINAL
10
MANAGEMENT SYSTEM
50
ACCEPT DEBUGGER ACTIVATION
S61
S62
DEBUGGER ACTIVATION REQUEST INFORMATION (TERMINAL ID, APPLICATION ID)
S63
DETERMINE WHETHER TO PERMIT ACTIVATION OF DEBUGGER
S64
DEBUGGER ACTIVATION PERMISSION/REJECTION INFORMATION
ACTIVATE DEBUGGER
S65

FIG. 29


START
DEVELOPMENT TERMINAL MANAGEMENT DB 5006
S63-1
EXTRACT DEVELOPER ID ASSOCIATED WITH TERMINAL ID
S63-2
EXTRACT DEVELOPER ID ASSOCIATED WITH APPLICATION THAT IS RUNNING
S63-3
ACTIVATION REQUEST SENDER IS DEVELOPER OF APPLICATION THAT IS RUNNING?
YES
NO
S63-5
REJECT ACTIVATION REQUEST
S63-4
PERMIT ACTIVATION REQUEST
END

APPARATUS, SYSTEM, AND METHOD OF ACTIVATION CONTROL, AND MEDIUM STORING ACTIVATION CONTROL PROGRAM

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This patent application is based on and claims priority pursuant to 35 U.S.C. §119(a) to Japanese Patent Application No. 2014-013315, filed on Jan. 28, 2014, in the Japan Patent Office, the entire disclosure of which is hereby incorporated by reference herein.

BACKGROUND

[0002] 1. Technical Field

[0003] The present invention relates to control of activation of an application that can be activated on a plurality of communication terminals.

[0004] 2. Description of the Related Art

[0005] Communication systems that perform a videoconference or the like via a communication network such as the Internet or a dedicated line have become popular in recent years due to a demand for reducing business trip costs and time. In such communication systems, a videoconference is performed through transmitting/receiving image data and sound data between a plurality of communication terminals.

[0006] In communication systems using personal computers (PCs), mobile phones, or the like, a communication terminal may obtain an application from an application server, and activates the obtained application on the communication terminal. In these communication systems, a debugger may be used in development or maintenance of such application. More specifically, with the debugger, the contents of application that is activated on the communication terminal, such as a source code, can be checked, to detect, for example, a bug.

SUMMARY

[0007] Example embodiments of the present invention include an activation control apparatus that controls activation of a plurality of applications that can be activated on a plurality of communication terminals. The activation control apparatus stores first association information that associates, for each one of the plurality of applications, application identification information for identifying an application with terminal identification information for identifying a communication terminal permitted to debug the application. The activation control apparatus further generates, for a first communication terminal of the plurality of communication terminals, an application list indicating one or more applications that can be activated on the first communication terminal. When the first communication terminal is provided with a debugger, the activation control apparatus determines whether any one of one or more candidate applications available for use by the first communication terminal is associated with terminal identification information of the first communication terminal using the first association information, and excludes from the candidate applications one or more applications that are not associated with the terminal identification information of the first communication terminal to generate the application list of one or more applications that can be activated on the first communication terminal. When the first communication terminal is not provided with a debugger, the activation control apparatus generates the application list that includes all of the one or more candidate applications as one or more applications that can be activated on the first communication terminal. The activation control apparatus sends the application list to the first communication terminal from which an application to be activated on the first communication terminal is selected.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0008] A more complete appreciation of the disclosure and many of the attendant advantages and features thereof can be readily obtained and understood from the following detailed description with reference to the accompanying drawings, wherein:

[0009] FIG. **1** is a schematic diagram of a communication system according to an embodiment of the present invention;

[0010] FIG. **2** is an illustration for explaining transmission or reception of data between communication terminals of the communication system of FIG. **1**;

[0011] FIG. **3** is an example of an outer appearance of the communication terminal of the communication system of FIG. **1**, functioning as a videoconference terminal;

[0012] FIG. **4** is a hardware configuration diagram of the communication terminal of FIG. **1**, functioning as a videoconference terminal;

[0013] FIG. **5** is a hardware configuration diagram of the communication terminal of FIG. **1**, functioning as a development terminal;

[0014] FIG. **6** is a hardware configuration diagram of any one of a communication management system, relay device, application server, program providing system, and maintenance system of the communication system of FIG. **1**;

[0015] FIG. **7** is a software configuration diagram of the communication terminal of FIG. **1**, functioning as a videoconference terminal;

[0016] FIG. **8** is a software configuration diagram of the communication terminal of FIG. **1**, functioning as a development terminal;

[0017] FIG. **9** is a functional block diagram of the communication terminal, the communication management system, and the application server, of the communication system of FIG. **1**;

[0018] FIG. **10** is an illustration of a terminal authentication management table;

[0019] FIG. **11** is an illustration of an application use management table according to a first embodiment;

[0020] FIG. **12** is an illustration of an application uniform resource locater (URL) management table;

[0021] FIG. **13** is an illustration of a developer name management table;

[0022] FIG. **14** is an illustration of an application developer management table;

[0023] FIG. **15** is an illustration of a development terminal management table;

[0024] FIG. **16** is an illustration for explaining transmission/reception of various types of information in the communication system;

[0025] FIG. **17** is a sequence diagram illustrating operation of registering a development terminal, performed by the terminal and the management system, according to an example embodiment of the present invention;

[0026] FIG. **18** is a sequence diagram illustrating operation of registering an application, performed by the communication system of FIG. **1**, according to an example embodiment of the present invention;

[0027] FIG. **19** is a sequence diagram illustrating operation of displaying application icons after activation of a communication terminal, performed by the communication system of FIG. **1**, according to an example embodiment of the present invention;

[0028] FIG. **20** is a sequence diagram illustrating operation of requesting activation of an application, performed by the communication system of FIG. **1**, according to an example embodiment of the present invention;

[0029] FIG. **21** is an illustration of an exemplary screen of an application list;

[0030] FIG. **22** is a flowchart illustrating operation of determining whether to permit activation of an application, performed by the management system of FIG. **1**, according to an example embodiment of the present invention;

[0031] FIG. **23** is a functional block diagram of the communication management system in the communication system of FIG. **1**, according to a second embodiment of the present invention;

[0032] FIG. **24** is an illustration of transmission/reception of various types of information in the communication system of FIG. **1** with the management system of FIG. **23**;

[0033] FIG. **25** is a flowchart illustrating operation of generating a list of applications, performed by the management system of FIG. **23**, according to an example embodiment of the present invention;

[0034] FIG. **26** is a functional block diagram of the communication management system in the communication system of FIG. **1**, according to a third embodiment of the present invention;

[0035] FIG. **27** is a flowchart illustrating operation of updating the application use management table, performed by the management system of FIG. **26**, according to an example embodiment of the present invention;

[0036] FIG. **28** is a sequence diagram illustrating operation of activating a debugger, performed by the terminal and the management system of the communication system of FIG. **1**, according to an example embodiment of the present invention; and

[0037] FIG. **29** is a flowchart illustrating operation of determining whether to permit activation of a debugger, performed by the management system of the communication system **1**, according to an example embodiment of the present invention.

[0038] The accompanying drawings are intended to depict example embodiments of the present invention and should not be interpreted to limit the scope thereof The accompanying drawings are not to be considered as drawn to scale unless explicitly noted.

DETAILED DESCRIPTION

[0039] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the present invention. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "includes" and/or "including", when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0040] In describing example embodiments shown in the drawings, specific terminology is employed for the sake of clarity. However, the present disclosure is not intended to be limited to the specific terminology so selected and it is to be understood that each specific element includes all technical equivalents that operate in a similar manner.

[0041] In the following description, illustrative embodiments will be described with reference to acts and symbolic representations of operations (e.g., in the form of flowcharts) that may be implemented as program modules or functional processes including routines, programs, objects, components, data structures, etc., that perform particular tasks or implement particular abstract data types and may be implemented using existing hardware at existing network elements or control nodes. Such existing hardware may include one or more Central Processing Units (CPUs), digital signal processors (DSPs), application-specific-integrated-circuits, field programmable gate arrays (FPGAs) computers or the like. These terms in general may be referred to as processors.

[0042] Unless specifically stated otherwise, or as is apparent from the discussion, terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical, electronic quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0043] Hereinafter, an embodiment of the present invention will be described using the drawings.

First Embodiment

[0044] First, a first embodiment of the present invention will be described.

<<Overall Configuration of Embodiment>>

[0045] FIG. **1** is a schematic diagram of a communication system **1** according to the first **2 0** embodiment of the present invention. As illustrated in FIG. **1**, the communication system **1** includes a plurality of communication terminals (**10***aa*, **10***ab*, . . .), displays (**120***aa*, **120***ab*, . . .) for the respective communication terminals (**10***aa*, **10***ab*, . . .), a plurality of relay devices (**30***a*, **30***b*, **30***c*, and **30***d*), a communication management system **50**, application servers (**80***ab* and **80***cd*), a program providing system **90**, and a maintenance system **100**.

[0046] In the following description, the term "communication terminal" is simply represented as the term "terminal", and the term "communication management system" is simply represented as the term "management system". In addition, an arbitrary one or ones of the plurality of terminals (**10***aa*, **10***ab*, . . .) is/are represented as a "terminal(s) **10**". An arbitrary one or ones of the plurality of displays (**120***aa*, **120***ab*, . . .) is/are represented as a "display(s) **120**". An arbitrary one or ones of the plurality of relay devices (**30***a*, **30***b*, **30***c*, and **30***d*) is/are represented as a "relay device(s) **30**". Further, a terminal serving as a request sender that gives a request to start a videoconference is represented as a "request sender terminal", and a terminal serving as a counterpart terminal that is a request destination (relay destination) is represented as a "counterpart terminal". In addition, an arbitrary one or ones of routers (**70***a,* **70***b,* **70***c,* **70***d,* **70***ab*, and **70***cd*) is/are represented as a "router(s) **70**". In addition, an arbitrary one of the application servers (**80***ab* and **80***cd*) is represented as an "application server **80**". Note that communication can be achieved using sound, video (image), or sound and video (image).

[0047] With the communication system **1**, a videoconference is performed among remotely-located places, through communicating image data and sound data, serving as examples of communication data. Note that the plurality of routers (**70***a,* **70***b,* **70***c,* **70***d,* **70***ab*, and **70***cd*) select an optimal path for communication data.

[0048] In addition, the terminals (**10***aa*, **10***ab*, **10***ac*, . . .), the relay device **30***a,* and the router **70***a* are connected to be communicable with each other by a local area network (LAN) **2***a*. The communication terminals (**10***ba*, **10***bb*, **10***bc*, . . .), the relay device **30***b,* and the router **70***b* are connected to be communicable with each other by a LAN **2***b*. In addition, the LAN **2***a* and the LAN **2***b* are connected to be communicable with each other by a dedicated line **2***ab* including the router **70***ab*. Further, the LAN **2***a,* the LAN **2***b,* and the dedicated line **2***ab* are configured in a company X, serving as an example of a company. For example, the LAN **2***a* is configured in an office A in the company X, and the LAN **2***b* is configured in an office B in the company X.

[0049] In this example, among the terminals **10** owned by the company X, the terminals (**10***ac* and **10***bc*) are development terminals, each of which are used for development of applications that may run on each terminal **10** of the communication system **1**, and are provided with a debugger. The application server **80***ab* is an application server owned by the company X and is connected to the Internet **2***i*. Applications developed by the terminals (**10***ac* and **10***bc*) are uploaded to the application server **80***ab* owned by the company X. In addition, each terminal **10** of the communication system **1** can download the applications from the application server **80***ab* and use the applications.

[0050] In contrast, the communication terminals (**10***ca*, **10***cb*, **10***cc*, . . .), the relay device **30***c,* and the router **70***c* are connected to be communicable with each other by a LAN **2***c*. The communication terminals **10***d* (**10***da*, **10***db*, **10***dc*, . . .), the relay device **30***d,* and the router **70***d* are connected to be communicable with each other by a LAN **2***d*. In addition, the LAN **2***c* and the LAN **2***d* are connected to be communicable with each other by a dedicated line **2***cd* including the router **70***cd*. Further, the LAN **2***c,* the LAN **2***d,* and the dedicated line **2***cd* are configured in a company Y, serving as an example of a company. For example, the LAN **2***c* is configured in an office C in the company Y, and the LAN **2***d* is configured in an office D in the company Y. The company X and the company Y are connected to be communicable with each other from the routers (**70***ab* and **70***cd*), respectively, via the Internet **2***i*.

[0051] In this example, among the terminals **10** owned by the company Y, the terminals (**10***cc* and **10***dc*) are development terminals, each of which are used for development of applications that may run on each terminal **10** of the communication system **1**, and are provided with a debugger. The application server **80***cd* is an application server owned by the company Y and is connected to the Internet **2***i*. Applications developed by the terminals (**10***cc* and **10***dc*) are uploaded to the application server **80***cd* owned by the company Y. In addition, each terminal **10** of the communication system **1** can download the applications from the application server **80***cd* and use the applications.

[0052] Further, the management system **50**, the program providing system **90**, and the maintenance system **100** are connected to the Internet **2***i*. Note that the locations of the management system **50**, the program providing system **90**, and the maintenance system **100** are not restricted, and these systems may be located in the same area or the same country or may be located in different areas or different countries.

[0053] Also in the embodiment, a communication network **2** of the embodiment includes the LAN **2***a,* the LAN **2***b,* the dedicated line tab, the Internet **2***i,* the dedicated line **2***cd*, the LAN **2***c,* and the LAN **2***d.* The communication network **2** may include not only a wired betwork, but also a network where communication is performed wirelessly, such as Wireless Fidelity (WiFi) or Bluetooth (registered trademark).

[0054] In addition in FIG. **1**, four digits indicated below each of the terminals **10**, each of the relay devices **30**, the management system **50**, each of the routers **70**, the program providing system **90**, and the maintenance system **100** indicates an IP address in an abbreviated form in the general Internet Protocol version 4 (IPv4). For example, the IP address of the terminal **10***aa* is "1.2.1.3". Although IPv6 may be used instead of IPv4, IPv4 is used in order to make the description simple.

[0055] Note that the terminals **10** may be used not only for communication between different offices or for communication between different rooms in the same office, but also for communication within the same room or for outdoor-indoor communication or outdoor-outdoor communication. In the case where the terminals **10** are used outside, wireless communication using a mobile phone communication network or the like is performed.

[0056] The terminals **10** illustrated in FIG. **1** achieve communication between or among users through transmitting/receiving communication data, and may be implemented by videoconference terminals, for example. Further, the terminals **10** transmit/receive communication data by using any desired communication protocol (a call control protocol for connecting or disconnecting to/from a communication destination, and a encoding format for converting communication data to IP packets).

[0057] Examples of the call control protocol includes the following: (1) Session Initiation Protocol (SIP); (2) H.323; (3) the extended SIP; (4) instant messaging (IM) protocol; (5) protocol using the SIP message method; (6) Internet Relay Chat (IRC) protocol; and (7) extended IM based protocol. Among them, (4) IM protocol is a protocol used in, for example, (4-1) Extensible Messaging and Presence Protocol (XMPP) or (4-2) ICQ (registered trademark), AIM (registered trademark), or Skype (registered trademark). In addition, (7) extended IM based on protocol is Jingle, for example.

[0058] Further, in this example, each of the terminals **10** is installed with a plurality of communication applications to be used for carrying out communication before the terminal **10** is shipped to a user side.

[0059] The terminals **10** that communicate with each other using the same communication application are capable of communicating via the communication network **2**. Examples of the communication applications, such as communication and messaging applications, include, but not limited to, Skype, Google Talk, LINE, FaceTime, Kakao Talk, and

Tango (registered or unregistered trademarks), for example, besides videoconference communication applications.

[0060] In addition, as illustrated in FIG. **2**, a management information session sei for transmitting and receiving various types of management information is established via the management system **50** between a request sender terminal and a counterpart terminal in the communication system **1**. A session for transmitting and receiving each of image data and sound data via a corresponding one of the relay devices **30** is also established between the request sender terminal and the counterpart terminal. Here, this session is indicated as an image/sound data session sed.

<<Hardware Configuration of Communication System>>

[0061] Next, the hardware configuration of the communication system **1** of the embodiment will be described. First, the hardware configuration of the terminals **10** will be described. The terminals **10** include videoconference terminals that carry out a videoconference between terminals **10** by using a communication application, and PCs for developing communication applications (hereinafter referred to as "development terminals") that can use a communication application and that have a debugger for the communication application. Hereinafter, the appearance of each of the videoconference terminals will be described.

[0062] FIG. **3** is an external view of a terminal **10** according to the embodiment. As illustrated in FIG. **3**, the terminal **10** includes a casing **1100**, an arm **1200**, and a camera housing **1300**. The casing **1100** has a front wall **1110** with an inlet face including a plurality of inlet holes, and a back wall **1120** having an exhaust face **1121** on which a plurality of exhaust holes are formed. Accordingly, by driving of a cooling fan included in the casing **1100**, air behind the transmission terminal **10** can be taken in via the inlet face and exhausted to the rear of the terminal **10** via the exhaust face **1121**. The casing **1100** has a right-side wall **1130** provided with a sound pickup hole **1131** formed thereon. Through the sound pickup hole **1131**, a built-in microphone **114**, described later, is capable of picking up sound and noise.

[0063] An operation panel **1150** is formed toward the right-side wall **1130** of the casing **1100**. The operation panel **1150** has a plurality of operation keys (**108***a* to **108***e*) described later, a power switch **109** described later, and an alarm lamp **119** described later, which are formed thereon. The operation panel **1150** also has a sound output face **1151** formed thereon, which is formed of a plurality of sound output holes for allowing output sound from a built-in speaker **115**, described later, to pass through. In addition, an accommodation portion **1160** serving as a recess for accommodating the arm **1200** and the camera housing **1300** is formed toward a left-side wall **1140** of the casing **1100**. A plurality of connection ports (**1132***a* to **1132***c*) for electrically connecting cables to an external device connection interface (I/F) **118** described later are provided on the right-side wall **1130** of the casing **1100**. In contrast, a connection port (not illustrated) for electrically connecting a cable **120***c* for a display **120** to the external device connection I/F **118** described later is provided toward the left-side wall **1140** of the casing **1100**.

[0064] The following description uses the term "operation key(s) **108**" for indicating an arbitrary one or ones of the operation keys (**108***a* to **108***e*), and the term "connection port(s) **1132**" for indicating an arbitrary one or ones of the connection ports (**1132***a* to **1132***c*).

[0065] Next, the arm **1200** is attached to the casing **1100** via a torque hinge **1210** so as to be rotatable in the vertical direction within a range of a tilt angle θ1 of 135 degrees with respect to the casing **1100**. FIG. **2** indicates a state in which the tilt angle θ1 is 90 degrees. The camera housing **1300** has a built-in camera **112** provided thereon, which will be described later, and the camera **112** can capture an image of a user, a document, a room, or the like. The camera housing **1300** also has a torque hinge **1310** formed thereon. The camera housing **1300** is attached to the arm **1200** via the torque hinge **1310** and is configured to be rotatable in the vertical and horizontal directions within a range of a pan angle θ2 of ±180 degrees and a tilt angle θ3 of ±45 degrees with respect to the state illustrated in FIG. **2** serving as 0 degrees.

[0066] Note that the external view illustrated in FIG. **3** is only exemplary and the appearance is not restricted thereto. The terminal **10** may be, for example, a general PC, a smart phone, a tablet terminal, an electronic black board, a projector, a car navigation apparatus mounted on a car, an image forming apparatus such as a multifunction peripheral or a printer, or a wearable terminal. The camera **112** and the microphone **114** need not necessarily be built-in devices and may be external devices.

[0067] Since the development terminals among the terminals **10**, the management system **50**, the program providing system **90**, and the maintenance system **100** each have the same appearance as that of a general server computer, descriptions of the appearances thereof are omitted.

[0068] Next, the hardware configuration of each of the videoconference terminals among the terminals **10** will be described. FIG. **4** is a hardware configuration diagram of a terminal **10** according to the embodiment. As illustrated in FIG. **4**, the terminal **10** includes a central processing unit (CPU) **101** that controls the overall operation of the terminal **10**, a read-only memory (ROM) **102** that stores a program used for controlling the CPU **101**, such as an initial program loader (IPL), a random-access memory (RAM) **103** used as a work area for the CPU **101**, a flash memory **104** that stores various types of data, such as a program for the communication terminal **10**, image data, and audio data, a solid state drive (SSD) **105** that controls reading/writing of various types of data from/to the flash memory **104** under control of the CPU **101**, a medium drive **107** that controls reading/writing (storage) of data from/to a recording medium **106** such as a flash memory, the operation keys **108** operated in the case of, for example, selecting a counterpart terminal of the terminal **10**, the power switch **109** for turning on/off the power of the terminal **10**, and a network interface (I/F) **111** for transmitting data using the communication network **2**.

[0069] The terminal **10** also includes the built-in camera **112**, which captures an image of a subject and obtains image data under control of the CPU **101**, an imaging element I/F **113** that controls driving of the camera **112**, the built-in microphone **114**, which receives an audio input, the built-in speaker **115**, which outputs sound, an audio input/output I/F **116** that processes inputting/outputting of an audio signal between the microphone **114** and the speaker **115** under control of the CPU **101**, a display I/F **117** that transmits image data to an external display **120** under control of the CPU **101**, the external device connection I/F **118** for connecting various external devices, the alarm lamp **119**, which indicates an abnormality of various functions of the terminal **10**, and a bus line **110** such as an address bus and a data bus for electrically connecting the above-described elements as illustrated in FIG. **4**.

[0070] Each of the displays **120** displays an image of a subject, an operated image, or the like, and includes a liquid crystal display (LCD) or an organic electroluminescence (EL) display. In addition, the display **120** is connected to the display I/F **117** by the cable **120***c*. The cable **120***c* may be an analog red green blue (RGB) (video graphic array (VGA)) signal cable, a component video cable, a high-definition multimedia interface (HDMI) signal cable, or a digital video interactive (DVI) signal cable.

[0071] The camera **112** includes a lens and a solid-state imaging element that converts an image (video) of a subject to electronic data by converting light to electric charge. As the solid-state imaging element, for example, a complementary metal-oxide-semiconductor (CMOS) or a charge-coupled device (CCD) is used.

[0072] The external device connection I/F **118** is capable of electrically connecting an external device such as an external camera, an external microphone, or an external speaker by using a Universal Serial Bus (USB) cable or the like, which is inserted into the connection port **1132** of the casing **1100** illustrated in FIG. **6**. In the case where an external camera is connected, the external camera is driven in preference to the built-in camera **112** under control of the CPU **101**. Similarly, in the case where an external microphone is connected or an external speaker is connected, the external microphone or the external speaker is driven in preference to the built-in microphone **114** or the built-in speaker **115** under control of the CPU **101**.

[0073] Note that the recording medium **106** is configured to be removable from the terminal **10**. In addition, a non-volatile memory that reads or writes data under control of the CPU **101** is not limited to the flash memory **104**, and an electrically erasable and programmable read-only memory (EEPROM) may be used instead.

[0074] Next, the points of the hardware configuration of each of the development terminals among the terminals **10** that are different from the videoconference terminals will be described. FIG. **5** is a hardware configuration diagram of a terminal **10** serving as the development terminal according to the embodiment. In the development terminal, the camera **112**, the microphone **114**, and the speaker **115** are externally-connected external devices and are connected to the external device connection I/F **118** via respective cables (**112***c*, **114***c*, and **115***c*).

[0075] In addition, the development terminal has no operation keys **108**, and, instead, the development terminal includes a keyboard **122** and a mouse **121** which serve as means for accepting inputs such as characters or selection results from a user. The development terminal also has no alarm lamp **119**, and instead, the development terminal is configured to notify the user of an error in the terminal **10** by using the speaker **115** or the display **120**.

[0076] FIG. **6** is a hardware configuration diagram of the management system **50** according to the embodiment of the present invention. The management system **50** includes a CPU **201** that controls the overall operation of the terminal management system **50**, a ROM **202** that stores a program used for driving the CPU **201**, such as an IPL, a RAM **203** used as a work area for the CPU **201**, an HD **204** that stores various types of data, such as a program for the management system **50**, a hard disk drive (HDD) **205** that controls reading/writing of various types of data from/to the HD **204** under control of the CPU **201**, a medium drive **207** that controls reading/writing (storage) of data from/to a recording medium **206** such as a flash memory, a display **208** that displays various types of information such as a cursor, a menu, a window, characters, or an image, a network I/F **209** for transmitting data using the communication network **2**, a keyboard **211** including a plurality of keys for entering characters, numerals, and various instructions, a mouse **212** that selects and executes various instructions, selects a processing target, and moves the cursor, a compact disc read-only memory (CD-ROM) drive **214** that controls reading/writing of various types of data from/to a

[0077] CD-ROM **213** serving as an example of a removable recording medium, and a bus line **210** such as an address bus and a data bus for electrically connecting the above-described elements, as illustrated in FIG. **6**.

[0078] Meanwhile, since the relay devices **30**, the application server **80**, the program providing system **90**, and the maintenance system **100** each have a hardware configuration that is the same as or similar to that of the above-described management system **50**, descriptions thereof are omitted.

[0079] Next, the software configuration of each of the videoconference terminals among the terminals **10** will be described. FIG. **7** is a software configuration diagram of a videoconference terminal. As illustrated in FIG. **7**, an operating system (OS) **1020**, a communication application **1031**, a communication application **1032**, a communication application **1033**, and a communication application **1034** run on a work area **1010** of the RAM **103**. The OS **1020** is installed in the terminal **10** prior to shipment to a user site, though this is not particularly limited. In addition, the communication applications (**1031**, **1032**, **1033**, and **1034**) may be installed in the terminal **10** prior to shipment or may be obtained from the application server **80** subsequent to shipment and installed in the terminal **10**, though this is not particularly limited.

[0080] In addition, among these applications, the OS **1020** is the basic software which provides basic functions and manages the entire terminal **10**. A browser is software that runs on the OS **1020** and is used to enable displaying and browsing of information to achieve a specific objective. The communication applications (**1031**, **1032**, **1033**, and **1034**) are software that runs on a browser **1021** and are used to perform communication with another terminal **10**. Note that, according to the embodiment of the present invention, the communication applications (**1031**, **1032**, **1033**, and **1034**) may communicate with another terminal **10** using different communication protocols that are different from one another.

[0081] Note that the communication applications (**1031**, **1032**, **1033**, and **1034**) are only exemplary, and other applications may be installed; for the sake of the simplicity of the description, four types of applications are described. In addition, in the case where a plurality of communication applications is installed, communication applications of different protocols may be installed, as in (1) to (7) described above.

[0082] Next, the points of the software configuration of each of the development terminals among the terminals **10** that are different from the videoconference terminals will be described. FIG. **8** is a software configuration diagram of a development terminal. In the development terminal, a development browser **1021** is installed as a browser that runs on the OS **1020**. The development browser **1021** is different from the browser **1021** of a videoconference terminal in the point that a debugger **1021***a* accompanies the development browser

1021. The debugger **1021** a is used to perform debugging of the communication applications (**1031**, **1032**, **1033**, and **1034**) when the communication applications (**1031**, **1032**, **1033**, and **1034**) are activated on the browser **1021**, by checking the contents of the activated applications, such as the source code.

<<Functional Configuration of Communication System>>

[0083] Next, the functional configuration of the communication system **1** of the embodiment will be described. FIG. **9** is a functional block diagram of a terminal **10**, the management system **50**, and the application server **80** that constitute part of the communication system **1** of the embodiment. In FIG. **9**, the terminal **10** and the management system **50** are connected to be capable of communicating data via the communication network **2**.

<Functional Configuration of Terminal>

[0084] The terminal **10** includes a device control **1050** and a communication control **1060**. The device control **1050** is executed with activation of the OS **1020** and the browser **1021** illustrated in FIG. **7** or **8**. In addition, the communication control **1060** is executed with activation of any of the communication applications (**1031**, **1032**, **1033**, and **1034**) illustrated in FIG. **7** or **8**.

[0085] The device control **1050** includes a data transmitter/receiver **11**, an operation input acceptor **12**, a display control **13**, an activation requester **14**, and a data processor **19**. These elements correspond to functions that are realized by operating any of the elements illustrated in FIG. **4** or **5** in response to a command from the CPU **101** in accordance with an activation application (program) expanded from the flash memory **104** to the RAM **103**.

[0086] The communication control **1060** includes a data transmitter/receiver **21**, an activator **22**, a display control **24**, a function executor **25**, and a data processor **29**. These elements are functions that are realized by operating any of the elements illustrated in FIG. **4** or **5** in response to a command from the CPU **101** in accordance with a communication application (program) expanded from the flash memory **104** to the RAM **103**.

[0087] The terminal **10** also includes a memory **1000** configured by the ROM **102**, the RAM **103**, and the flash memory **104**, illustrated in FIG. **4** or **5**.

(Functional Configuration of Device Control)

[0088] Next, using FIG. **9**, functional configuration of the device control **1050** of the terminal **10** will be described. Note that, in the following description of functional configuration of the device control **1050** of the terminal **10**, among elements illustrated in FIG. **4** or **5**, relationships with main elements for realizing each functional configuration of the device control **1050** will also be described.

[0089] The data transmitter/receiver **11** of the terminal **10** illustrated in FIG. **9** is implemented by a command from the CPU **101** illustrated in FIG. **4** or **5** and by the network I/F **111** illustrated in FIG. **4** or **5**, and performs transmission/reception of various types of data (or information) to/from a counterpart terminal, apparatus, or system via the communication network **2**.

[0090] The operation input acceptor **12** is implemented by a command from the CPU **101** illustrated in FIG. **4** or **5** and by the operation keys (**108***a*, **108***b*, **108***c*, **108***d*, and **108***e*) and the power switch **109** illustrated in FIG. **3**, and accepts various inputs or various selections from the user. For example, when the user turns on the power switch **109** illustrated in FIG. **3**, the operation input acceptor **12** illustrated in FIG. **9** accepts the power on operation and turns on the power.

[0091] The display control **13** is implemented by a command from the CPU **101** illustrated in FIG. **4** or **5** and by the display I/F **117** illustrated in FIG. **4** or **5**, and performs control for transmitting image data, sent from a counterpart terminal at the time of communication, to the display **120**.

[0092] The data processor **19** is implemented by a command from the CPU **101** illustrated in FIG. **4** or **5** and by the SSD **105** illustrated in FIG. **4** or **5**, or by a command from the CPU **101**, and performs processing to store various types of data in the memory **1000** or to read various types of data stored in the memory **1000**.

[0093] The activation requester **14** is implemented by the browser **1021** on the basis of a command from the CPU **101** illustrated in FIG. **4** or **5**, and requests the communication control **1060** to activate the communication applications (**1031**, **1032**, **1033**, and **1034**).

(Functional Configuration of Communication Control)

[0094] Next, referring to FIGS. **4**, **5**, and **9**, functional configuration of the communication control **1060** of the terminal **10** will be described. Note that, in the following description of functional configuration of the communication control **1060** of the terminal **10**, among elements illustrated in FIG. **4** or **5**, relationships with main elements for implementing functional configuration of the communication control **1060** will also be described.

[0095] The data transmitter/receiver **21** illustrated in FIG. **9** is implemented by a command from the CPU **101** illustrated in FIG. **4** or **5** and by the network I/F **111** illustrated in FIG. **4** or **5**, and performs transmission/reception of various types of data (or information) to/from a counterpart terminal, apparatus, or system via the communication network **2**.

[0096] The activator **22** is implemented by a command from the CPU **101** illustrated in FIG. **4** or **5**. When the operation input acceptor **12** of the device control **1050** accepts selection of an application by the user, the activator **22** activates the communication control **1060** (communication application) in response to an activation request from the operation input acceptor **12**.

[0097] The display control **24** is implemented by a command from the CPU **101** illustrated in FIG. **4** or **5** and by the display I/F **117** illustrated in FIG. **4** or **5**, and performs control for transmitting screen data to the display **120**.

[0098] The function executor **25** is implemented by a command from the CPU **101** illustrated in FIG. **4** or **5** and by the camera **112**, the microphone **114**, the speaker **115**, or the like illustrated in FIG. **4** or **5**, and controls communication through images, sounds, or the like.

[0099] The data processor **29** is implemented by a command from the CPU **101** illustrated in FIG. **4** or **5** and by the SSD **105** illustrated in FIG. **4** or **5**, or by a command from the CPU **101**, and performs processing to store various types of data in the memory **1000** or to read various types of data stored in the memory **1000**.

<Functional Configuration of Management System>

[0100] The management system **50** includes a data transmitter/receiver **51**, an authenticator **52**, a registrar **54**, an activation determiner **55**, and a data processor **59**. These elements correspond to functions that are realized by operating any of the elements illustrated in FIG. **6** in response to a command from the CPU **201** in accordance with a program for the management system **50**, which is expanded from the HD **204** to the RAM **203**. In addition, the management system **50** includes a memory **5000** configured by the HD **204** illustrated in FIG. **6**. In the memory **5000**, databases (DBs) (**5001**, **5003**, **5004**, and **5006**) are stored in the form of tables as discussed below.

(Terminal Authentication Management Table)

[0101] FIG. **10** is an illustration of a terminal authentication management table. In the memory **5000**, the terminal authentication management DB **5001** configured by a terminal authentication management table such as that illustrated in FIG. **10** is stored. The terminal authentication management table stores, for each of the terminals **10** managed by the management system **50**, a password for authentication in association with the terminal ID of the terminal. For example, the terminal authentication management table illustrated in FIG. **10** indicates that the terminal ID of the terminal **10***aa* is "01aa", and the password of the terminal **10***aa* is "aaaa",

(Application Use Management Table)

[0102] FIG. **11** is an illustration of an application use management table. In the memory **5000**, the application use management DB **5003** configured by an application use management table such as that illustrated in FIG. **11** is stored. The application use management table stores, for each one of applications that are installed in each terminal **10**, the start day and the end day of a period in which the application installed in the terminal **10** can be used (effective period) in association with the terminal ID for identifying the terminal **10** and the application ID for identifying the application. For example, in the application use management table illustrated in FIG. **11**, it is indicated that the terminal **10***aa* whose terminal ID is "01aa" can use an application identified by the application ID "a010" from Jan. 1, 2013 to Dec. 31, 2013.

(Application URL Management Table)

[0103] FIG. **12** is an illustration of an application URL management table. In the memory **5000**, the application URL management DB **5004** configured by an application URL management table such as that illustrated in FIG. **12** is stored. The application URL management table stores, for each of a plurality of applications managed by the management system **50**, URL information indicating a location of icon data on the communication network **2** and URL information indicating a location of the application on the communication network **2**, in association with the application ID.

(Development Terminal Management DB)

[0104] In the memory **5000**, the development terminal management DB **5006** configured by a developer name management table such as that illustrated in FIG. **13**, an application developer management table such as that illustrated in FIG. **14**, and a development terminal management table such as that illustrated in FIG. **15** is stored. Note that FIG. **13** is an illustration of a developer name management table. FIG. **14** is an illustration of an application developer management table. FIG. **15** is an illustration of a development terminal management table.

[0105] The developer name management table stores the name of a developer of an application in association with a developer ID for identifying the developer. For example, in the developer name management table illustrated in FIG. **13**, it is indicated that the name of a developer whose developer ID is "dv**01**" is "Company A". Note that a developer in the communication system **1** of the embodiment is permitted to debug an application developed by the developer using a development terminal owned by the developer. In addition, a developer is not particularly limited and may be a person or an organization. The application developer management table stores a developer ID in association with an application ID for identifying an application. For example, in the application developer management table illustrated in FIG. **14**, it is indicated that an application identified by the application ID "a010" is developed by the company A identified by the developer ID "dv**01**".

[0106] The development terminal management table stores the developer ID of a developer who owns a terminal **10** in association with the terminal ID of this terminal **10**. For example, in the development terminal management table illustrated in FIG. **15**, it is indicated that the terminal **10***ac* identified by the terminal ID "01ac" is owned by the company A identified by the developer ID "dv**01**".

(Functional Configuration of Management System)

[0107] Next, functional configuration of the communication management system **50** will be described in detail. Note that, in the following description of functional configuration of the communication management system **50**, among elements illustrated in FIG. **6**, relationships with main elements for realizing functional configuration of the communication management system **50** will also be described.

[0108] The data transmitter/receiver **51** is implemented by a command from the CPU **201** illustrated in FIG. **6** and by the network I/F **209** illustrated in FIG. **6**, and performs transmission/reception of various types of data (or information) to/from another terminal, apparatus, or system via the communication network **2**.

[0109] The authenticator **52** is implemented by a command from the CPU **201** illustrated in FIG. **6**, and authenticates a terminal **10** by searching the terminal authentication management table (see FIG. **10**) using a terminal ID and a password received by the data transmitter/receiver **51** as search keys and determining whether the same terminal ID and the same password are managed in the terminal authentication management table.

[0110] The registrar **54** is realized by a command from the CPU **201** illustrated in FIG. **6**, and registers various information such as developer IDs, developer names, application IDs, and terminal IDs in the tables of the development terminal management DB **5006**.

[0111] The activation determiner **55** is implemented by a command from the CPU **201** illustrated in FIG. **6**, and determines whether to permit activation of an application requested to be activated.

[0112] The data processor **59** is implemented by a command from the CPU **201** illustrated in FIG. **6** and by the HDD **205** illustrated in FIG. **6**, or by a command from the CPU **201**, and performs processing to store various types of data in the memory **5000** or to extract various types of data stored in the memory **5000**.

<Functional Configuration of Application Server>

[0113] The application server **80** includes a data transmitter/receiver **81** and a data processor **89**. These elements are functions that are realized by operating any of the elements illustrated in FIG. **6** in response to a command from the CPU **201** in accordance with a program for the application server **80**, which is expanded from the HD **204** to the RAM **203**. In addition, the application server **80** includes a memory **8000** configured by the HD **204** illustrated in FIG. **6**.

(Application Hosting DB)

[0114] The memory **8000** stores an application hosting DB **8001**. The application hosting DB **8001** stores an application developed by a development terminal among the terminals **10** and uploaded to the application server **80**.

[0115] The data transmitter/receiver **81** is implemented by a command from the CPU **201** illustrated in FIG. **6** and by the network I/F **209** illustrated in FIG. **6**, and performs transmission/reception of various types of data (or information) to/from another terminal, apparatus, or system via the communication network **2**.

[0116] The data processor **89** is implemented by a command from the CPU **201** illustrated in FIG. **6** and by the HDD **205** illustrated in FIG. **6**, or by a command from the CPU **201**, and performs processing to store various applications in the memory **8000** or to extract various applications stored in the memory **8000**.

<<Process or Operation >>

[0117] Next, referring to FIG. **16**, the outline of an operation of the embodiment will be described. FIG. **16** is a conceptual diagram illustrating transmission/reception of various types of information in the communication system **1**. The management system **50** functions as an activation control apparatus, which controls activation of applications that run on the terminals **10**. The management system **50** is further provided with the development terminal management DB **5006** that functions as a debugging-permitted terminal manager, which stores the application developer management table. The application developer management table stores the developer ID (an example of developer identification information) of a developer permitted to debug each application, in association with this application. The development terminal management DB **5006** further stores the development terminal management table, which stores the terminal ID of a development terminal owned by each developer in association with the developer ID of this developer. That is, for each application, the development terminal management DB **5006** manages a development terminal permitted to debug this application. More specifically, the development terminal management DB **5006** stores first association information that associates, for each application, identification information for identifying a communication terminal that is permitted to debug the application.

[0118] In response to a request from a development terminal to activate an application, if the application requested to be activated is an application other than an application(s) associated with the terminal ID of this development terminal in the development terminal management DB **5006**, the data transmitter/receiver **51** transmits information indicating rejection of activation to the development terminal, which has given the activation request. In contrast, in response to a request from a terminal **10** that is not a development terminal to activate an application, regardless of information managed in the development terminal management DB **5006**, the data transmitter/receiver **51** allows activation by transmitting the URL (an example of information for accessing the application) of the application to the terminal **10**, which has given the activation request.

[0119] With the development terminal management DB **5006**, the management system **50** controls not to activate, on a development terminal, applications other than those permitted to be debugged by this development terminal, thereby preventing copying of applications based on debugging.

[0120] Further, when an additional terminal is registered in the development terminal management table for management by the management system **50**, the management system **50** permits a debugger to be installed on that terminal **10** having a terminal ID that is newly registered. This allows the management system **50** to manage the terminal **10** in which a debugger is installed. Therefore, the activation determiner **55** can determine whether a terminal **10** which has given a request to activate an application is a terminal **10** with a debugger.

[0121] Next, operation of controlling activation of an application is explained according to the embodiment will be described in detail. FIG. **17** is a sequence diagram illustrating example operation of registering a development terminal. In FIG. **17**, various items of management information are transmitted/received by the management information session sei illustrated in FIG. **2**.

[0122] First, when the user turns on the power switch **109** illustrated in FIG. **3**, the operation input acceptor **12** illustrated in FIG. **9** accepts the power on operation and activates the terminal **10** (step S1). In response to acceptance of the power on operation, the data transmitter/receiver **11** transmits a login request to the management system **50** via the communication network **2** (step S2). Accordingly, the data transmitter/receiver **51** of the management system **50** accepts the login request. Note that a login request may be given in response to an instruction input from the user of a terminal **10** serving as a login request sender.

[0123] The login request includes a terminal ID for identifying the terminal **10**, which serves as a request sender, and a password. The terminal ID and the password are read via the data processor **19** from the memory **1000** and sent to the data transmitter/receiver **11**. Note that the terminal ID and the password may be input by the user of the terminal **10**, which serves as a login request sender. In addition, with login request information from a terminal **10**, the management system **50** may receive the IP address of the terminal **10**.

[0124] At step **3**, the authenticator **52** of the management system **50** searches the terminal authentication management table (see FIG. **10**) of the memory **5000** using the terminal ID and the password included in the login request information received via the data transmitter/receiver **51** as search keys, and determines whether the same terminal ID and the same password are stored in the terminal authentication management table to authenticate the terminal **10**.

[0125] The data transmitter/receiver **51** of the management system **50** transmits authentication result information indicating an authentication result obtained by the authenticator **52** to the terminal **10**, which is the login request sender, via the communication network **2** (step S4). Accordingly, the data transmitter/receiver **11** of the terminal **10**, which is the login request sender, receives the authentication result information.

The following explains the case where it has been determined by the authenticator **52** that the terminal **10** is a terminal that has a legitimate use authority.

[0126] The data transmitter/receiver **11** of the terminal **10** transmits information that indicates a request for developer registration to the management system **50** via the communication network **2** (step S**5**). This request includes the developer name "Company A". Accordingly, the data transmitter/receiver **51** of the management system **50** accepts the request for developer registration.

[0127] Next, the registrar **54** of the management system **50** generates the developer ID "dv**01**" indicating the developer, who may be the registration request sender, and registers the developer. More specifically, the registrar **54** stores, in the developer name management table (FIG. **13**), the generated developer ID "dv**01**" in association with the developer name "Company A" transmitted from the registration request sender (step S**6**). In response to completion of developer registration to the developer name management table of FIG. **13**, the data transmitter/receiver **51** transmits registration result information indicating completion of the developer registration to the terminal **10** (step S**7**). The developer registration result information includes the developer ID "dv**01**" generated by the registrar **54**.

[0128] Next, the data transmitter/receiver **11** of the terminal **10** transmits information that indicates a request for development terminal registration to the management system **50** via the communication network **2** (step S**8**). This request includes the developer ID "dv**01**" and the terminal ID of a terminal **10** to be registered as a development terminal. Accordingly, the data transmitter/receiver **51** of the management system **50** accepts the request for development terminal registration.

[0129] Next, the registrar **54** of the management system **50** registers the development terminal by storing, in the development terminal management table (FIG. **15**), the terminal ID of the registration request sender and the developer ID transmitted from the registration request sender in association with each other (step S**9**). In response to completion of development terminal registration to the development terminal management table of FIG. **15**, the data transmitter/receiver **51** transmits development terminal registration result information indicating completion of the registration to the terminal **10** (step S**10**). The development terminal registration result information includes information necessary for installing a browser with a debugger function (such as the URL of an installer). Accordingly, the terminal **10**, which is the registration request sender, is able to install a debugger (step S**11**) that is obtained using the debugger information.

[0130] In response to completion of installing a debugger in the terminal **10**, the terminal **10** transmits information indicating that installing of a debugger is completed to the management system **50** (step S**12**). With this notification, the communication management system **50** knows that the terminal **10**, which is registered in the development terminal management table, is a terminal with a debugger. In order to more accurately determine a terminal **10** with a debugger, the registrar **54** of the management system **50** may register the terminal ID in the development terminal management table (FIG. **15**) after acceptance of information indicating that installing a debugger is completed. Through operation of FIG. **17**, the terminal **10**, which is now installed with a debugger, is registered as a development terminal.

[0131] Next, referring to FIG. **18**, operation of registering, in the application developer management table (see FIG. **14**), an application developed by a terminal **10** that is a development terminal will be described. FIG. **18** is a sequence diagram illustrating operation of registering an application. In FIG. **18**, various items of management information are transmitted/received by the management information session sei illustrated in FIG. **2**.

[0132] First, the data transmitter/receiver **11** of the terminal **10** transmits information that indicates a request for application registration to the management system **50** via the communication network **2** (step S**21**). This request includes the developer ID "dv**01**", the terminal ID of the registration request sender, and the URLs of the application server **80** serving as registration destinations of the icon image of an application and a registration destination of the application. Accordingly, the data transmitter/receiver **51** of the management system **50** accepts the request for application registration.

[0133] Next, the registrar **54** of the management system **50** generates an application ID for identifying an application developed by the registration request sender, and registers the application developer by storing, in the application developer management table (FIG. **14**), the generated application ID in association with the developer ID transmitted from the registration request sender (step S**22**).

[0134] Next, the registrar **54** of the management system **50** stores, in association with the application ID of the application developed by the registration request sender, the URLs of the registration destinations of the icon image of the application and of the application, which is transmitted from the registration request sender, in the application URL management table (FIG. **12**), thereby registering the URL of the application (step S**23**).

[0135] In response to completion of the application registration, the data transmitter/receiver **51** transmits registration completion information indicating completion of the registration to the terminal **10**, which is the registration request sender (step S**24**). The registration completion information includes the application ID generated by the registrar **54** at S**22**. In response to acceptance of the registration completion information by the data transmitter/receiver **11** of the terminal **10**, the data transmitter/receiver **11** transmits to the application server **80** requests for registering the application at the URL of the application and registering the image data of the icon at the URL of the image data of the icon (step S**25**).

[0136] In response to acceptance of these requests by the data transmitter/receiver **81** of the application server **80**, the data processor **89** registers the application and the image data of the icon, transmitted from the terminal **10**, at the specified registration destinations in the application hosting DB **8001** (step S**26**). In response to completion of the registration, the data transmitter/receiver **81** of the application server **80** transmits registration completion information indicating completion of registration of the application and the icon to the terminal **10**, which is the registration request sender.

[0137] Next, referring to FIG. **19**, operation of displaying an application icon indicating a candidate for an application requested to be activated on the terminal **10** will be described. FIG. **19** is a sequence diagram illustrating a process up to displaying an application icon. In FIG. **19** and later-described FIG. **20**, various items of management information are transmitted/received entirely by the management information session sei illustrated in FIG. **2**.

[0138] After the terminal **10** logs in to the management system **50** through processing that is the same as or similar to steps S1 to S4, the data transmitter/receiver **11** of the terminal **10** gives a request for a list of candidates for usable applications to the management system **50** via the communication network **2** (step S**31**). Accordingly, the data transmitter/receiver **51** of the management system **50** accepts the request for usable applications. This request includes the terminal ID of the terminal **10**, which is the list request sender.

[0139] Next, the data processor **59** of the management system **50** searches the application use management table (see FIG. **11**) by using the terminal ID of the terminal **10**, which is the list request sender, received in step S**31** described above, as a search key, thereby reading application IDs corresponding to the terminal ID, and their effective period start days and effective period end days (step S**32**). The data processor **59** further extracts, from among the application IDs read in step S**32** described above, an application that is within its effective period (from the effective period start day to the effective period end day) at the time of this processing, and searches the application URL management table (see FIG. **12**) by using the extracted application ID as a search key, thereby reading URL information of an icon corresponding to the application ID (step S**33**).

[0140] Next, the data transmitter/receiver **11** transmits, as a list of candidates for usable applications, usable application information to the terminal **10**, which is the list request sender, via the communication network **2** (step S**34**). The usable application information includes the URL information of the icon read in step S**33** described above. Accordingly, the data transmitter/receiver **11** of the terminal **10**, which is the list request sender, receives the usable application information.

[0141] Next, the data transmitter/receiver **11** of the terminal **10** accesses a resource in the application hosting DB **8001**, indicated by the URL of the icon received in step S**34** described above, and gives a request to obtain the image data of the icon (step S**35**). Accordingly, the data transmitter/receiver **81** of the application server **80** accepts the request to obtain the image data of the icon.

[0142] Next, the data processor **89** of the application server **80** reads the image data of the icon, requested in step S**35**, from the application hosting DB **8001** of the memory **8000** (step S**36**). The data transmitter/receiver **81** transmits the image data of the icon to the terminal **10**, which is the obtaining request sender, via the communication network **2** (step S**37**). Accordingly, the data transmitter/receiver **11** of the terminal **10**, which is the request sender, receives the image data of the icon.

[0143] Next, the display control **13** displays an "application list" screen **140**, such as that illustrated in FIG. **21**, on the display **120** (step S**38**). Note that FIG. **21** is a diagram illustrating an exemplary screen of an application list. On the screen **140**, the icon of an application within its effective period is displayed. Here, the icons (**141**, **142**, and **143**) of three applications indicated by three application IDs (a**001**, a**002**, and a**003**), respectively, are displayed.

[0144] Next, referring to FIG. **20**, operation of requesting activation of a selected application, which is selected using an application icon from an application list will be described. Note that FIG. **20** is a sequence diagram illustrating operation of requesting activation of a selected application.

[0145] First, in response to selection of a desired icon from among a plurality of icons displayed in FIG. **21** in response to a user operation, the operation input acceptor **12** of the terminal **10** accepts this selection of an application icon by the user (step S**41**). Next, the data transmitter/receiver **11** of the terminal **10** transmits activation request information that indicates a request for activation of the selected application to the management system **50** via the communication network **2** (step S**42**). The activation request information includes the terminal ID of the terminal **10**, which is the request sender terminal, and the application ID of the selected application.

[0146] The data transmitter/receiver **51** of the management system **50** accepts the activation request by receiving the activation request information. In response to acceptance of the activation request, the activation determiner **55** of the management system **50** determines whether to permit activation of the application requested to be activated (step S**43**). The processing in step S**43** will be described in detail using FIG. **22**. FIG. **22** is a flowchart illustrating a process of determining whether to permit activation of an application.

[0147] First, the data processor **59** of the management system **50** searches the development terminal management table (see FIG. **15**) of the development terminal management DB **5006** by using the terminal ID of the terminal **10**, which is the activation request sender, thereby extracting a corresponding developer ID (step S**43**-**1**). Next, the activation determiner **55** determines whether the terminal **10**, which is the activation request sender, is a development terminal on the basis of whether a developer ID corresponding to the terminal ID of the terminal **10**, which is the activation request sender, has been extracted in step S**43**-**1** (step S**43**-**2**).

[0148] In the case where the terminal **10**, which is the activation request sender, is not a development terminal (NO in step S**43**-**2**), that is, in the case where no developer ID associated with the terminal ID of the terminal **10** has been extracted in step S**43**-**1**, the activation determiner **55** permits activation of the application requested to be activated (step S**43**-**5**). This is because the terminal **10**, which is the activation request sender, is not a development terminal, and is incapable of debugging the application.

[0149] In the case where the terminal **10**, which is the activation request sender, is a development terminal (YES in step S**43**-**2**), that is, in the case where a developer ID associated with the terminal ID of the terminal **10** has been extracted in step S**43**-**1**, the data processor **59** of the management system **50** searches the application developer management table (see FIG. **14**) of the development terminal management DB **5006** by using the application ID of the application requested to be activated as a search key, thereby extracting a corresponding developer ID (step S**43**-**3**). Next, the activation determiner **55** determines whether the activation request sender is the developer of the application requested to be activated, by comparing the developer ID extracted in step S**43**-**1** with the developer ID extracted in step S**43**-**3** (step S**43**-**4**).

[0150] In the case where the activation request sender is the developer of the application requested to be activated (YES in step S**43**-**4**), that is, in the case where the developer ID extracted in step S**43**-**1** matches the developer ID extracted in step S**43**-**3**, the activation determiner **55** permits activation of the application requested to be activated (step S**43**-**5**). This is because the application requested to be activated is developed by the activation request sender. In response to permission of the activation request, the data processor **59** searches the application URL management table (see FIG. **12**) by using the application ID of the application requested to be activated as a search key, thereby reading URL information corresponding to the application (step S**43**-**6**).

[0151] In the case where the activation request sender is not the developer of the application requested to be activated (NO in step S**43-4**), that is, in the case where the developer ID extracted in step S**43-1** does not match the developer ID extracted in step S**43-3**, the activation determiner **55** rejects activation of the application requested to be activated (step S**43-7**). This is because the application requested to be activated is not developed by the activation request sender.

[0152] Referring back to FIG. **20**, in response to completion of determination by the activation determiner **55** whether to permit activation of the application, the data transmitter/receiver **51** of the management system **50** transmits activation permission/rejection information indicating whether activation of the application is permitted to the terminal **10**, which is the activation request sender (step S**44**). Note that, in the case where activation is permitted in step S**43-5**, the URL information of the application, extracted in step S**43-6**, is also transmitted as the URL for accessing the application requested to be activated.

[0153] In response to reception of the activation permission/rejection information by the data transmitter/receiver **11** of the terminal **10**, which is the activation request sender, the display control **13** displays the activation permission/rejection information indicating whether activation is permitted, on the display **120**. In the case where activation is permitted, the activation requester **14** of the device control **1050** illustrated in FIG. **9** gives an activation command to the activator **22** of the communication control **1060**, thereby activating the communication control **1060** (step S**45**). That is, the application selected by the user is activated. Note that, in the case where the terminal **10**, which is the activation request sender, is a development terminal, the application is activated in a debug mode. In this example, the processing prior to activation at S**45** is performed by the device control **1050**, and the processing after activation at S**45** is performed by the communication control **1060**.

[0154] In response to activation of the application, the data transmitter/receiver **21** accesses the URL of the application, included in the activation permission/rejection information, and requests to download the application (step S**46**). In response to acceptance by the data transmitter/receiver **81** of the application server **80** of the request to download the application, the data processor **89** reads the application identified by the URL (step S**47**). The read application is transmitted by the data transmitter/receiver **81** to the terminal **10**, which is the download request sender. Accordingly, the terminal **10**, which is the activation request sender, is able to run the application on the browser **1021**.

Second Embodiment

[0155] Referring to FIGS. **23** to **25**, operation of controlling activation of an application is explained according to a second embodiment of the present invention. The operation of FIGS. **23** to **25** is substantially similar to the above-described operation of FIGS. **16** to **22**, except for some differences, which will be described below. The communication system **1** of the second embodiment is different from the first embodiment in that the management system **50** includes a list generator **56**, as illustrated in FIG. **23**. Note that FIG. **23** is a functional block diagram of the management system **50**. The list generator **56** generates a list of applications that can be requested to be activated by a terminal **10**.

[0156] Referring to FIG. **24**, the outline of an operation or a process of the second embodiment will be described. FIG. **24** is a conceptual diagram illustrating transmission/reception of various types of information in the communication system **1**. In the second embodiment, in step S**32**, after application IDs corresponding to the terminal ID of a terminal **10** that serves as a list request sender are read, the list generator **56** identifies, from among the read application IDs, application candidates to be included in the list by excluding application IDs that are not within their effective periods at the time of processing.

[0157] In the case of generating a list of applications that can be requested to be activated by a development terminal, the list generator **56** (an example of an application list generator) of the management system **50** generates a list by excluding applications other than those associated with the terminal ID of the development terminal in the development terminal management DB **5006** from application candidates to be included in the list. In contrast, in the case of generating a list of applications that can be requested to be activated by a terminal **10** that is not a development terminal, the list generator **56** generates a list without excluding applications managed in the development terminal management DB **5006** from application candidates to be included in the list.

[0158] With the development terminal management DB **5006**, the management system **50** is able to exclude, from application candidates that can be requested to be activated by a development terminal, applications other than those permitted to be debugged by the development terminal. Accordingly, applications that can be activated by a terminal **10** with a debugger can be restricted, thereby preventing copying of applications.

[0159] Next, referring to FIG. **25**, the outline of an operation or a process of the second embodiment will be described. FIG. **25** is a flowchart illustrating operation of generating a list of applications. In the second embodiment, the processing in step S**32** (FIG. **19**) of the first embodiment is changed to a series of processes illustrated in FIG. **25**.

[0160] First, the data processor **59** of the management system **50** searches the application use management table (see FIG. **11**) by using the terminal ID of the terminal **10**, which is the list request sender, as a search key, thereby reading application IDs corresponding to the terminal ID, and their effective period start days and effective period end days (step S**32-1**). Next, the list generator **56** identifies, from among the read application IDs, application candidates to be included in the list by excluding application IDs that are not within their effective periods at the time of processing (step S**32-2**).

[0161] Next, the data processor **59** of the management system **50** searches the development terminal management table (see FIG. **15**) of the development terminal management DB **5006** by using the terminal ID of the list request sender as a search key, thereby extracting a corresponding developer ID (step S**32-3**). Next, the list generator **56** determines whether the terminal **10**, which is the list request sender, is a development terminal on the basis of whether a developer ID corresponding to the terminal ID of the list request sender has been extracted in step S**32-2** (step S**32-4**).

[0162] In the case where the terminal **10**, which is the list request sender, is not a development terminal (NO in step S**32-4**), that is, in the case where no developer ID associated with the terminal ID of the terminal **10** has been extracted in step S**32-3**, the list generator **56** includes all the application candidates identified in step S**32-2** in the list of applications that can be requested to be activated by the terminal **10** (step S**32-9**).

[0163] In the case where the terminal **10**, which is the list request sender, is a development terminal (YES in step S**32-4**), that is, in the case where a developer associated with the terminal ID of the terminal **10** has been extracted in step S**32-3**, the data processor **59** extracts, for each of the application IDs of the application candidates identified in step S**32-2**, a corresponding developer ID by searching the application developer management table (see FIG. **14**) by using this application ID as a search key (step S**32-5**). Next, the list generator **56** determines whether the list request sender is the developer of each of the application candidates to be included in the list, by comparing the developer ID extracted in step S**32-3** with a corresponding one of the developer IDs extracted in step S**32-5** (step S**32-6**).

[0164] In the case where the list request sender is the developer of the application candidate (YES in step S**32-6**), that is, in the case where the developer ID extracted in step S**32-3** matches the developer ID extracted in step S**32-5**, the list generator **56** includes the application candidate in the list of applications that can be requested to be activated (step S**32-7**).

[0165] In the case where the list request sender is not the developer of the application candidate (NO in step S**32-6**), that is, in the case where the developer ID extracted in step S**32-3** does not match the developer ID extracted in step S**32-5**, the list generator **56** does not include the application candidate in the list of applications that can be requested to be activated (step S**32-8**).

[0166] In response to completion of determination of whether to include each application in the list, the list generator **56** generates the list of applications that can be requested to be activated by the terminal **10** by using the application ID(s) determined to be included in the list (step S**32-10**).

[0167] In response to completion of the list, the data processor **59** searches the application URL management table (see FIG. **12**) by using each of the application IDs included in the list, generated in step S**32** described above, as a search key, thereby reading the URL information of an icon corresponding to this application ID (step S**33**). Since the processing from this point onward is the same as the first embodiment, a description thereof is omitted.

Third Embodiment

[0168] Referring to FIGS. **26** and **27**, operation of controlling activation of an application is explained according to a third embodiment of the present invention. The operation of FIGS. **26** to **27** is substantially similar to the above-described operation of FIGS. **16** to **22** or the operation of FIGS. **23** to **25**, except for some differences. The third embodiment is different from the first embodiment or the second embodiment in that the management system **50** includes an application use manager **57**, as illustrated in FIG. **26**. FIG. **26** is a functional block diagram of the management system **50**. The application use manager **57** updates various types of information recorded in the application use management DB **5003**. In the third embodiment, in response to a purchase of an application from a terminal **10**, the terminal ID, application ID, and effective period are updated in the application use management DB **5003** on the basis of conditions of purchase. Also in the third embodiment, in response to a purchase of an application from a terminal **10**, it becomes possible for the terminal **10** to use the purchased application within a specified effective period on the basis of conditions of purchase.

[0169] First, the outline of an operation or a process of the third embodiment will be described. The application use management DB **5003** (an example of an application manager, see FIG. **11**) stores, in association with each terminal **10**, the application ID (an example of application information) of an application purchased by that terminal **10**. More specifically, the application use management DB **5003** stores second association information that associates terminal identification information of the terminal with application information indicating one or more applications that can be used by the terminal. The data transmitter/receiver **51** accepts, from a terminal **10**, a request for purchase of a desired application (an example of a request to enable the use of an application).

[0170] In response to acceptance of a request for purchase from a development terminal, the application use manager **57** (an example of an updater) rejects the request for purchase of a desired application in the case where the terminal ID of the development terminal serving as a purchase request sender is not managed in association with the desired application in the development terminal management DB **5006**. In contrast, in response to acceptance of a request for purchase from a terminal **10** that is not a development terminal, the application use manager **57** associates and manages the application ID of a desired application as an application that can be used by this terminal **10**, thereby updating the application use management DB **5003**. Accordingly, applications that can be purchased by a terminal **10** with a debugger can be restricted, thereby preventing copying of applications by activating the purchased applications in a debugger mode.

[0171] Next, referring to FIG. **27**, the outline of an operation or a process of the third embodiment will be described. FIG. **27** is a flowchart illustrating operation of updating an application use management table. First, the data transmitter/receiver **51** of the management system **50** accepts a request for purchase of a desired application from a terminal **10** (step S**51**). Next, the data processor **59** of the management system **50** searches the development terminal management table (see FIG. **15**) of the development terminal management DB **5006** by using the terminal ID of the terminal **10**, which is the purchase request sender, as a search key, thereby extracting a corresponding developer ID (step S**52**). Next, the application use manager **57** determines whether the terminal **10**, which is the purchase request sender, is a development terminal on the basis of whether a developer ID corresponding to the terminal ID of the purchase request sender has been extracted in step S**52** (step S**53**).

[0172] In the case where the terminal **10**, which is the purchase request sender, is not a development terminal (NO in step S**53**), that is, in the case where no developer ID associated with the terminal ID of the terminal **10** has been extracted in step S**52**, the application use manager **57** updates the application use management table (step S**56**). In this case, the application use manager **57** updates the application use management table by storing the application ID of the desired application in association with the terminal ID of the terminal **10**, which is the purchase request sender. In addition, the application use manager **57** updates the application use management table by managing the effective period start day and the effective period end day, which are based on conditions of purchase, in association with the terminal ID of the terminal **10**, which is the purchase request sender, and the application ID of the desired application.

[0173] In the case where the terminal **10**, which is the purchase request sender, is a development terminal (YES in step **S53**), that is, in the case where a developer ID associated with the terminal ID of the terminal **10** has been extracted in step **S52**, the data processor **59** of the management system **50** searches the application developer management table (see FIG. **14**) of the development terminal management DB **5006** by using the application ID of the desired application requested to be purchased as a search key, thereby extracting a corresponding developer ID (step **S54**). Next, the application use manager **57** determines whether the purchase request sender is the developer of the desired application requested to be purchased, by comparing the developer ID extracted in step **S52** with the developer ID extracted in step **S54** (step **S55**).

[0174] In the case where the purchase request sender is the developer of the desired application requested to be purchased (YES in step **S55**), that is, in the case where the developer ID extracted in step **S52** matches the developer ID extracted in step **S54**, the application use manager **57** updates the application use management table, as in step **S56** described above (step **S56**).

[0175] In the case where the purchase request sender is not the developer of the desired application requested to be purchased (NO in step **S55**), that is, in the case where the developer ID extracted in step **S52** does not match the developer ID extracted in step **S54**, the application use manager **57** rejects the purchase of the application (step **S57**).

[0176] The determination result indicating whether the purchase of the application is permitted is transmitted by the data transmitter/receiver **51** to the terminal **10**, which is the purchase request sender.

Fourth Embodiment

[0177] Referring to FIGS. **28** and **29**, operation of controlling activation of an application is explained according to a fourth embodiment of the present invention. In the fourth embodiment, a development terminal activates an application without functioning a debugger. In such case, the management system **50** does not determine whether to permit application activation in step **S43** of the first embodiment, and performs control to activate an application requested to be activated. In response to a request for activation of a debugger from a development terminal after activation of an application, the management terminal **50** determines whether to permit activation of a debugger.

[0178] Next, referring to FIG. **28**, the outline of an operation or a process of the fourth embodiment will be described. FIG. **28** is a sequence diagram illustrating operation of activating a debugger. First, while an application is running on a development terminal, in response to acceptance of a request for activation of a debugger by the operation input acceptor **12** (step **S61**), the data transmitter/receiver **11** transmits to the management system **50** debugger activation request information including the terminal ID of a terminal **10** that serves as the debugger activation request sender and the application ID of the application that is running (step **S62**).

[0179] The data transmitter/receiver **51** of the management system **50** accepts the debugger activation request by receiving the debugger activation request information. In response to acceptance of the debugger activation request, the activation determiner **55** of the management system **50** determines whether to permit activation of the debugger requested to be activated (step **S63**). The processing in step **S63** will be described referring to FIG. **29**. FIG. **29** is a flowchart illustrating a process of determining whether to permit activation of a debugger.

[0180] First, the data processor **59** of the management system **50** searches the development terminal management table (see FIG. **15**) of the development terminal management DB **5006** by using the terminal ID of the debugger activation request sender as a search key, thereby extracting a corresponding developer ID (step **S63-1**).

[0181] Next, the data processor **59** of the management system **50** searches the application developer management table (see FIG. **14**) of the development terminal management DB **5006** by using the application ID of the application running on the debugger activation request sender, thereby extracting a corresponding developer ID (step **S63-2**). Next, the activation determiner **55** determines whether the debugger activation request sender is the developer of the application running on the debugger activation request sender, by comparing the developer ID extracted in step **S63-1** with the developer ID extracted in step **S63-2** (step **S63-3**).

[0182] In the case where the debugger activation request sender is the developer of the application activated on the debugger activation request sender (YES in step **S63-3**), that is, in the case where the developer ID extracted in step **S63-1** matches the developer ID extracted in step **S63-2**, the activation determiner **55** permits activation of the debugger (step **S63-4**).

[0183] In the case where the debugger activation request sender is not the developer of the application activated on the debugger activation request sender (NO in step **S63-3**), that is, in the case where the developer ID extracted in step **S63-1** does not match the developer ID extracted in step **S63-2**, the activation determiner **55** rejects activation of the debugger (step **S63-5**).

[0184] In response to completion of determination by the activation determiner **55** whether to permit activation of the debugger, the data transmitter/receiver **51** of the management system **50** transmits activation permission/rejection information indicating whether activation of the debugger is permitted to the development terminal, which is the debugger activation request sender (step **S64**).

[0185] In response to reception of the debugger activation permission/rejection information by the data transmitter/receiver **11** of the terminal **10**, which is the debugger activation request sender, the display control **13** displays the debugger activation permission/rejection information indicating whether activation of the debugger is permitted, on the display **120**. In the case where activation of the debugger is permitted, the activation requester **14** of the device control **1050** illustrated in FIG. **9** gives a command to activate the debugger to the activator **22** of the communication control **1060**, thereby activating the debugger (step **S65**). Accordingly, the use who is the developer at the terminal **10** is able to debug the application that is running.

[0186] As has been described above, according to the above-described embodiment, the activation control apparatus manages, in association with each application, a communication terminal permitted to debug this application. Accordingly, the management system is able to control so as not to activate, on a communication terminal with a debugger, applications other than those permitted to be debugged. This prevents copying of applications based on debugging.

[0187] For example, assuming that an application, which is released, may be installed onto any communication terminal.

If the application is activated on a communication terminal with a debugger, a developer of such communication terminal can not only check the contents of the application, but can easily copy such application. According to the above-described embodiment, control is performed not to activate applications other than those permitted to be debugged, thereby preventing copying of applications through debugging.

[0188] The relay devices **30**, the management system **50**, the program providing system **90**, and the maintenance system **100** in the above-described embodiments may be configured by a single computer or a plurality of computers to which divided portions (functions) are arbitrarily allocated. In addition, in the case where the program providing system **90** is configured by a single computer, a program transmitted by the program providing system **90** may be separately transmitted in units of a plurality of modules, or may be transmitted in its entirety. Further, in the case where the program providing system **90** is configured by a plurality of computers, a program may be divided into a plurality of modules, and the modules may be individually transmitted from the respective computers.

[0189] In addition, a recording medium such as a compact disc read-only memory (CD-ROM) storing the terminal program, relay device program, or communication management program in the above-described embodiments, the HD **204** storing these programs, and the program providing system **90** including the HD **204** are each used in the case where the terminal program, relay device program, or communication management program is provided as a program product to users within a certain country or outside that country.

[0190] Further, each ID in the above-described embodiments indicates identification information, which may be expressed in the form of any language, a character(s), a symbol(s), or various marks as long as such identification information can uniquely identify a corresponding device. In addition, each ID may be identification information including a combination of at least two of the above-mentioned character (s), symbol(s), and various marks.

[0191] For example, an application ID is an example of application identification information, and the application identification information includes the name of the application, besides the application ID. In addition, a terminal ID is an example of terminal identification information, and the terminal identification information includes the serial number of the terminal **10** and a user ID allocated to the user of the terminal **10**, besides the terminal ID.

[0192] In addition, icons illustrated in FIG. **21** may include characters, picture symbols, or picture symbols including characters or symbols.

[0193] In addition, although videoconference terminals have been described as examples of the terminals **10** in the above description, the embodiments are not limited thereto. For example, one or both of a request sender terminal and a counterpart terminal may be IP phones, Internet phones, or PCs. Further, the communication terminals may be information processing terminals, such as smart phones, tablet terminals, game machines, or car navigation apparatuses, which can perform not only communication but also various types of data communication other than communication. In this case, the communication management system **50** executes various processes as a communication management system.

[0194] Numerous additional modifications and variations are possible in light of the above teachings. It is therefore to be understood that within the scope of the appended claims, the disclosure of the present invention may be practiced otherwise than as specifically described herein. For example, elements and/or features of different illustrative embodiments may be combined with each other and/or substituted for each other within the scope of this disclosure and appended claims.

[0195] Each of the functions of the described embodiments may be implemented by one or more processing circuits or circuitry. Processing circuitry includes a programmed processor, as a processor includes circuitry. A processing circuit also includes devices such as an application specific integrated circuit (ASIC) and conventional circuit components arranged to perform the recited functions.

[0196] The present invention can be implemented in any convenient form, for example using dedicated hardware, or a mixture of dedicated hardware and software. The present invention may be implemented as computer software implemented by one or more networked processing apparatuses. The network can comprise any conventional terrestrial or wireless communications network, such as the Internet. The processing apparatuses can compromise any suitably programmed apparatuses such as a general purpose computer, personal digital assistant, mobile telephone (such as a WAP or 3G-compliant phone) and so on. Since the present invention can be implemented as software, each and every aspect of the present invention thus encompasses computer software implementable on a programmable device. The computer software can be provided to the programmable device using any storage medium for storing processor readable code such as a floppy disk, hard disk, CD ROM, magnetic tape device or solid state memory device.

[0197] The hardware platform includes any desired kind of hardware resources including, for example, a central processing unit (CPU), a random access memory (RAM), and a hard disk drive (HDD). The CPU may be implemented by any desired kind of any desired number of processor. The RAM may be implemented by any desired kind of volatile or non-volatile memory. The HDD may be implemented by any desired kind of non-volatile memory capable of storing a large amount of data. The hardware resources may additionally include an input device, an output device, or a network device, depending on the type of the apparatus. Alternatively, the HDD may be provided outside of the apparatus as long as the HDD is accessible. In this example, the CPU, such as a cache memory of the CPU, and the RAM may function as a physical memory or a primary memory of the apparatus, while the HDD may function as a secondary memory of the apparatus.

1. An activation control apparatus that controls activation of a plurality of applications that can be activated on a plurality of communication terminals, the apparatus comprising:

a memory that stores first association information that associates, for each one of the plurality of applications, application identification information for identifying an application with terminal identification information for identifying a communication terminal permitted to debug the application;

processing circuitry that generates, for a first communication terminal of the plurality of communication terminals, an application list indicating one or more applications that can be activated on the first communication terminal,

wherein, when the first communication terminal is provided with a debugger, the processing circuitry determines whether any one of one or more candidate applications available for use by the first communication terminal is associated with terminal identification information of the first communication terminal using the first association information, and excludes from the candidate applications one or more applications that are not associated with the terminal identification information of the first communication terminal to generate the application list of one or more applications that can be activated on the first communication terminal, and

when the first communication terminal is not provided with a debugger, the processing circuitry generates the application list that includes all of the one or more candidate applications as one or more applications that can be activated on the first communication terminal; and

a transmitter that sends the application list to the first communication terminal from which an application to be activated on the first communication terminal is selected.

2. The activation control apparatus of claim **1**, further comprising:

a receiver that receives an activation request for activating the selected application selected from the application list, from the first communication terminal,

wherein, in response to the activation request, the transmitter further sends information used for accessing the selected application to the first communication terminal.

3. The activation control apparatus of claim **1**, wherein

the memory further stores second association information that associates, for each one of the plurality of communication terminals, terminal identification information of the communication terminal with application information indicating one or more applications available for use by the communication terminal, and

the processing circuitry obtains the one or more candidate applications available for use by the first communication terminal using the second association information.

4. The activation control apparatus of claim **1**, wherein the first association information includes

association information that associates, for each one of the plurality of applications, developer identification information for identifying a developer permitted to debug the application with the application identification information of the application; and

association information that associates the developer identification information of the developer with the terminal identification information for identifying a communication terminal owned by the developer.

5. The activation control apparatus of claim **4**, wherein the processing circuitry determines whether the first communication terminal is provided with a debugger, based on a determination of whether terminal identification information of the first communication terminal is associated with any developer identification information.

6. The activation control apparatus of claim **4**, wherein, in response to a registration request for registering an additional communication terminal as a communication terminal owned by the developer,

the processing circuitry requests the additional communication terminal to install a debugger, and additionally stores terminal identification information for identifying the additional communication terminal in association with the developer identification information of the developer.

7. A communication system comprising:

the activation control apparatus of claims **1**; and

the first communication terminal that sends an activation request for activating the selected application selected from the application list to the activation control apparatus through a network.

8. The communication system of claim **7**, further comprising:

an application server including a memory that stores the plurality of applications,

wherein the activation control apparatus sends information used for accessing the selected application to the first communication terminal in response to the activation request.

9. A method of controlling activation of a plurality of applications that can be activated on a plurality of communication terminals, comprising:

storing in a memory first association information that associates, for each one of the plurality of applications, application identification information for identifying an application with terminal identification information for identifying a communication terminal permitted to debug the application;

generating, for a first communication terminal of the plurality of communication terminals, an application list indicating one or more applications that can be activated on the first communication terminal,

wherein, when the first communication terminal is provided with a debugger, the generating includes:

determining whether any one of one or more candidate applications available for use by the first communication terminal is associated with terminal identification information of the first communication terminal using the first association information; and

excluding from the candidate applications one or more applications that are not associated with the terminal identification information of the first communication terminal to generate the application list of one or more applications that can be activated on the first communication terminal, and

when the first communication terminal is not provided with a debugger, the generating includes

generating the application list that includes all of the one or more candidate applications as one or more applications that can be activated on the first communication terminal; and

sending the application list to the first communication terminal from which an application to be activated on the first communication terminal is selected.

10. The activation control method of claim **9**, further comprising:

receiving an activation request for activating the selected application selected from the application list, from the first communication terminal; and

sending information used for accessing the selected application to the first communication terminal in response to the activation request.

11. The activation control method of claim **9**, further comprising:

storing in the memory second association information that associates, for each one of the plurality of communication terminals, terminal identification information of the communication terminal with application information indicating one or more applications available for use by the communication terminal; and

obtaining the one or more candidate applications available for use by the first communication terminal using the second association information.

12. The activation control method of claim **9**, wherein the first association information includes association information that associates, for each one of the plurality of applications, developer identification information for identifying a developer permitted to debug the application with the application identification information of the application; and

association information that associates the developer identification information of the developer with the terminal identification information for identifying a communication terminal owned by the developer.

13. The activation control method of claim **12**, wherein the determining whether the first communication terminal is provided with a debugger includes

determining whether terminal identification information of the first communication terminal is associated with any developer identification information.

14. The activation control method of claim **12**, wherein, in response to a registration request for registering an additional communication terminal as a communication terminal owned by the developer, the method further comprising;

requesting the additional communication terminal to install a debugger; and

additionally storing terminal identification information for identifying the additional communication terminal in association with the developer identification information of the developer.

15. A non-transitory recording medium storing a plurality of instructions which, when executed, cause one or more processors to perform a method of controlling activation of a plurality of applications that can be activated on a plurality of communication terminals, the method comprising:

storing in a memory first association information that associates, for each one of the plurality of applications, application identification information for identifying an application with terminal identification information for identifying a communication terminal permitted to debug the application;

generating, for a first communication terminal of the plurality of communication terminals, an application list indicating one or more applications that can be activated on the first communication terminal,

wherein, when the first communication terminal is provided with a debugger, the generating includes:

determining whether any one of one or more candidate applications available for use by the first communication terminal is associated with terminal identification information of the first communication terminal using the first association information; and

excluding from the candidate applications one or more applications that are not associated with the terminal identification information of the first communication terminal to generate the application list of one or more applications that can be activated on the first communication terminal, and

when the first communication terminal is not provided with a debugger, the generating includes

generating the application list that includes all of the one or more candidate applications as one or more applications that can be activated on the first communication terminal; and

sending the application list to the first communication terminal from which an application to be activated on the first communication terminal is selected.

* * * * *