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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 October 2007 (11.10.2007)

PCT

(10) International Publication Number
WO 2007/114669 A1

(51) International Patent Classification:
H04N 5/45 (2006.01) *H04N 5/00* (2006.01)

(21) International Application Number:
PCT/KR2007/001675

(22) International Filing Date: 5 April 2007 (05.04.2007)

(25) Filing Language: Korean

(26) Publication Language: English

(30) Priority Data:

60/789,577	6 April 2006 (06.04.2006)	US
60/812,090	9 June 2006 (09.06.2006)	US
60/870,471	18 December 2006 (18.12.2006)	US
60/918,894	20 March 2007 (20.03.2007)	US
10-2007-0033464	4 April 2007 (04.04.2007)	KR

(71) Applicant: **SAMSUNG ELECTRONICS CO., LTD.** [KR/KR]; 416, Maetan-dong, Yeongtong-gu, Suwon-si, Gyeonggi-do 442-742 (KR).

(72) Inventors: **LEE, Jong-Ho**; 102-701, Lotte Castle Empire, Yeouido-dong, Yeongdeungpo-gu, Seoul 150-936 (KR). **LEE, Kwang-Kee**; 102-3204, The# starliver 11-8, Sincheon-dong, Songpa-gu, Seoul 138-240 (KR). **BYUN, Sung-Wook**; 304-2306, Dongsuwon LG Village 3-cha, Mangpo-dong, Yeongtong-gu, Suwon-si, Gyeonggi-do 443-773 (KR). **ADAMS, Glenn A.**; 77 Strawberry Hill Road, Acton, Ma, Massachusetts 01720 (US). **JUNG, Un-Gyo**; 106-1003, Pungseong Sinmiju Apt., 895, Gian-ri, Taean-eup, Hwaseong-si, Gyeonggi-do 445-971 (KR).

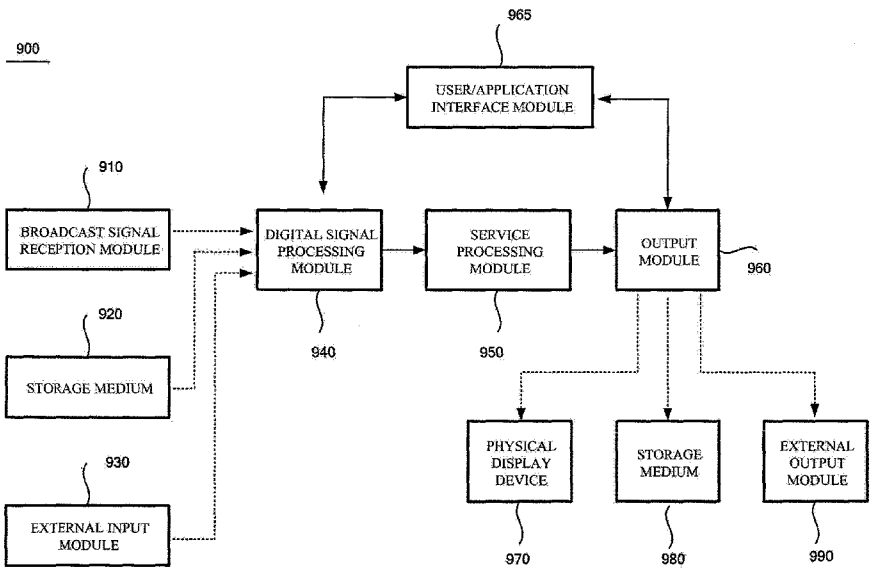
(74) Agent: **KASAN IP & LAW FIRM**; Youngpoong Building-6th Floor, 142, Nonhyun-dong, Gangnam-gu, Seoul 135-822 (KR).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: APPARATUS AND METHOD FOR IDENTIFYING AN APPLICATION IN THE MULTIPLE SCREENS ENVIRONMENT



(57) Abstract: A method and apparatus for identifying an application in a multi-screen environment are provided. The apparatus includes a digital signal processing module receiving video information, audio information, or data information and restoring a service based on the video information, the audio information or the data information, a service processing module producing a plurality of logical screens for displaying the restored service, and an output module mapping the plurality of logical screens provided by the service processing module to different locations on a display screen. An application present in the service is identified by a service context in which the application is executed and by identification inform.

WO 2007/114669 A1

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

APPARATUS AND METHOD FOR IDENTIFYING AN APPLICATION IN THE MULTIPLE SCREENS ENVIRONMENT

Technical Field

The present invention relates to a method and apparatus for identifying an application in a multi-screen environment.

Background Art

Conventional broadcast receivers such as digital TVs or digital set-top boxes provide only one content element on a single physical display device or simultaneously display a main screen and a sub-screen on a single physical display device.

Even though conventional broadcast receivers can simultaneously display both the main screen and the sub-screen on the same display screen, they can only arrange the main screen and the sub-screen in a limited number of manners. In the case of a content displayed within the main screen, all elements of the content, i.e., video data, audio data, and other data, are displayed. On the other hand, in the case of a content displayed within the sub-screen, only some of the elements of the content are displayed.

Content sources include a broadcast service such as a satellite broadcaster, a terrestrial broadcaster, or a cable broadcaster, a storage medium such as DVDs, or an external device connected to an input terminal. However, it is quite difficult to display contents provided by such various content sources on a display screen using the existing broadcast receivers.

In an interactive TV application program environment such as Multimedia Home Platform (MHP), Advanced Common Application (ACAP), Open Cable Application Platform (OCAP), it is assumed that only one screen is output on a physical display device.

In the interactive TV application program environment, for example, a Home Audio/Video Interoperability (HAVi)-based user interface (UI) is adopted. According to the HAVi UI standard, even though no restriction is imposed on the number of screens displayed on a physical display device, only one screen is generally displayed on a physical display device.

Disclosure

Technical Problem

In such an environment, it is difficult to perform operations, such as decoding, digital signal processing, user interaction processing, etc. with respect to one among multimedia contents displayed on a screen while displaying the multimedia contents on independent screens. In addition, it is also difficult to dynamically control the life cycles of application programs and the use of resources in the units of the screens.

Accordingly, there exists a need for a method of identifying an application that operates in a multi-screen environment while displaying a variety of contents on a display screen in various manners.

Technical Solution

The present invention provides identifying an application that is executed on a plurality of screens in a multi-screen environment in which a plurality of content items are displayed on a physical display screen.

The above and other objects of the present invention will be described in or be apparent from the following description of the preferred embodiments.

According to an aspect of the present invention, there is provided an apparatus for identifying an application in a multi-screen environment, the apparatus including a digital signal processing module receiving video information, audio information, or data information and restoring a service based on the video information, the audio information or the data information, a service processing module producing a plurality of logical screens for displaying the restored service, and an output module mapping the plurality of logical screens provided by the service processing module to different locations on a display screen, wherein an application present in the service is identified by a service context in which the application is executed and by identification information.

According to another aspect of the present invention, there is provided an apparatus for identifying an application in a multi-screen environment, the apparatus including a service processing module producing a logical screen displaying a service and a display screen that allows the logical screen to be associated with the service and displays the logical screen; and an output module mapping the produced logical screen to a certain area on the produced display screen, wherein an application in the service is

identified by a service context in which the application is executed and identification information.

According to another aspect of the present invention, there is provided a method of identifying an application in a multi-screen environment, the method including receiving video information, audio information, or data information and restoring a service, producing a plurality of logical screens displaying the restored service, and mapping the logical screens to different locations on a display screen, wherein an application in the service is identified by a service context in which the application is executed and identification information.

According to another aspect of the present invention, there is provided a method of identifying an application in a multi-screen environment, the method including producing a plurality of logical screens displaying a service and a display screen displaying the plurality of logical screen, and mapping the logical screens to arbitrary areas on the display screen, wherein an application in the service is identified by a service context in which the application is executed and identification information.

Description of Drawings

The above and other features and advantages of the present invention will become more apparent by describing in detail preferred embodiments thereof with reference to the attached drawings in which:

FIG. 1 is a diagram illustrating a configuration of a PiP screen according to an exemplary embodiment of the present invention;

FIG. 2 is a diagram illustrating the relationship between a logical screen and a display screen according to an exemplary embodiment of the present invention;

FIGS. 3 to 7 are diagrams illustrating a configuration of a screen including a mapper according to an exemplary embodiment of the present invention;

FIG. 8 is a block diagram illustrating service sources according to an exemplary embodiment of the present invention;

FIGS. 9 and 10 are diagrams illustrating a non-abstract service and an abstract service according to an exemplary embodiment of the present invention;

FIG. 11 is a diagram illustrating attribute information and interfaces of a logical screen and a display screen;

FIG. 12 is a diagram illustrating an attribute 'z-order' of a logical screen according to an exemplary embodiment of the present invention;

FIGS. 13 and 14 are diagrams each illustrating an attribute 'Display_Area' of a logical screen according to exemplary embodiments of the present invention;

FIG. 15 is a diagram illustrating a method of mapping two services to a display screen according to an exemplary embodiment of the present invention;

FIG. 16 is a block diagram illustrating a configuration of an apparatus for providing multiple screens according to an exemplary embodiment of the present invention;

FIG. 17 is a flowchart illustrating a method of dynamically configuring multiple screens according to an exemplary embodiment of the present invention;

FIG. 18 is a diagram illustrating a software architecture for providing multiple screens according to an exemplary embodiment of the present invention;

FIG. 19 is a diagram illustrating the relationships among modules constituting an application program interface (API) layer according to an exemplary embodiment of the present invention;

FIG. 20 is a flowchart illustrating a method of displaying a plurality of services that are displayed on respective corresponding logical screens on a display screen by the modules illustrated in FIG. 19;

FIG. 21 is a flowchart illustrating a method of exchanging services displayed on logical screens between the modules illustrated in FIG. 19

FIG. 22 illustrates the operation of an application on a plurality of screens;

FIG. 23 illustrates a data structure for identifying an application according to an exemplary embodiment of the present invention; and

FIG. 24 illustrates a method of identifying an application according to an exemplary embodiment of the present invention.

<Reference Names of Major Components Shown in the Drawings>

900: apparatus for providing multiple screens

910: broadcast signal reception module

920: storage medium

930: external input module

940: digital signal processing module

950: service processing module

960: output module

965: user/application interface module

970: physical display device

980: storage medium

990: external output module

Mode for Invention

Advantages and features of the present invention and methods of accomplishing the same may be understood more readily by reference to the following detailed description of preferred embodiments and the accompanying drawings. The present invention may, however, be embodied in many different forms and should not be construed as being limited to the embodiments set forth herein. Rather, these embodiments are provided so that this disclosure will be thorough and complete and will fully convey the concept of the invention to those skilled in the art, and the present invention will only be defined by the appended claims. Like reference numerals refer to like elements throughout the specification.

The present invention is described hereinafter with reference to flowchart illustrations of user interfaces, methods, and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations, and combinations of blocks in the flowchart illustrations, can be implemented by computer program instructions. These computer program instructions can be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions specified in the flowchart block or blocks.

These computer program instructions may also be stored in a computer usable or computer-readable memory that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer usable or computer-readable memory produce an article of manufacture including instruction means that implement the function specified in the flowchart block or blocks.

The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer implemented process such that the instructions that execute on the computer or other programmable apparatus provide steps for implementing the functions specified in the flowchart block or blocks.

And each block of the flowchart illustrations may represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function(s). It should also be noted that in some alternative implementations, the functions noted in the blocks may occur out of the order. For example, two blocks illustrated in succession may in fact be executed substantially concurrently or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved.

For a better understanding of the present invention, the terms used in this disclosure will now be defined.

The term 'service' indicates a group of multimedia contents displayed together, i.e., a group of service components.

Service components are elements of a service and include a video component, an audio component, and a data component. A data component is an application program in a service.

The term 'service context' indicates an object which can control the executing of a service and includes various resources, devices, and execution state information needed for providing a service.

The term 'physical display device' indicates a physical space which actually displays the content of a service.

The term 'display screen' indicates a screen actually displayed on a physical display device. An arbitrary service may be directly set in the display screen, and the display screen may be displayed on a physical display device. Alternatively, at least one logical screen which is mapped to a certain area of the display screen may be displayed on the physical display device.

The term 'logical screen' indicates a space in which an arbitrary service is displayed. A logical screen is a virtual screen before being mapped to a display screen and thus is not displayed on a physical display device.

The logical screen and the display screen may be a combination of a background still image, a video raster, and a graphic raster. The graphic raster may be a combination of text, lines, colors, and images or a mixture of video frames.

The term 'main service' indicates a service that is selected as a main service through a menu displayed on the physical display device or a remote controller by a user or through an API by an application, and the screen on which the main service is displayed is referred to as a 'main screen'.

The term 'Picture-in-Picture service (PiP service)' indicates a service that is selected as a sub-service in the main service through a menu displayed on a physical display device or a remote controller by a user via an API by an application, and the PiP service may be displayed on a picture-in-picture screen (PiP screen) or a main screen.

The PiP screen includes a screen that occupies a part of another screen as illustrated in FIGS. 1(a) to 1(d) and a screen that is simultaneously displayed with another screen without overlapping the other screen as illustrated in FIGS. 1(e) to 1(f). In this case, it is understood that the PiP screen may include a screen that overlaps another screen on an arbitrary location or area in the physical display device, as illustrated in FIGS. 1(g) and 1(h). FIG. 2 is a diagram illustrating the relationship between a logical screen and a display screen according to an exemplary embodiment of the present invention.

Referring to FIG. 2, a service may be displayed using logical screens 210, 212, and 214. The logical screens 210, 212, and 214 are mapped to display screens 220, 222, and 224 through a mapping block 230.

In detail, the logical screens 210 and 212 are mapped to the display screen 220, the logical screens 212 and 214 are mapped to the display screen 222, and the logical screens 210, 212, and 214 are mapped to the display screen 224.

In short, at least one logical screen which displays a service is mapped to an arbitrary area of a display screen.

The mapping block 230 is a group of various pieces of information needed for mapping a logical screen to a display screen. Examples of the various pieces of information include coordinate information of a predetermined area on a display screen to

which each of a plurality of logical screens is mapped, identification information of the logical screens and the display screen, and information specifying in what order the logical screens are displayed on the display screen.

The mapping block 230 can change the size of the logical screen so to be allocated in an arbitrary area of the display screen. That is, the mapping block 230 can perform scaling of the logical screen and allocating of the position thereof, and FIGS. 3 to 7 are diagrams illustrating a configuration of the screen including a mapper as the mapping block.

Referring to FIG. 3, the main screen including a combination of a background still image B, a video raster V, and a graphic raster G is mapped to the entire display screen by a mapper with a normal size. The PiP screen including only video components is mapped to the entire display screen by the mapper with a reduced size. In this case, the mapped PiP screen is displayed on the main screen, which is determined depending on a Z value. The reference character Z refers to z-order value which will be described later. An overlay screen may be combined with the display screen. The overlay screen is a specific screen disposed at the outmost side, and may be used when providing a caption function. The PiP screen may have only a video component as illustrated in FIG. 3, or may have a combination of the background still image B, the video raster V, and the graphic rater G as illustrated in FIG. 4.

Referring to FIG. 5, the main screen including the combination of the background still image B, the video raster V, and the graphic rater G is mapped to the entire display screen by the mapper with a normal size. Two PiP screens #1 and #2 having only video component is mapped to an arbitrary area of the display screen by the mapper with a reduced size. In this case, the mapped PiP screen is disposed on the main screen and the Z value can be constantly maintained. Further, the overlay screen may be combined with the display screen. The configuration of the screen may have a plurality of PiP screens including only video components as illustrated in FIG. 5 or a plurality of PiP screens including a combination of the background still image B, the video raster V, and the graphic rater G.

POP screens are illustrated in FIG. 7. It can be understood that the PiP screen is displayed inside the main screen and the POP screen is displayed outside the main screen. Referring to FIG. 7, the plurality of PiP screens #1 and #2 including a combination of the

background still image B, the video raster V, and the graphic rater G are mapped to arbitrary areas of the display screen by the mapper with a reduced size. In this case, the Z value of the mapped POP screens #1 and #2 may be constantly maintained. Further, the overlay screen may be combined with the display screen.

The mapping block 230 may be realized by interfaces or functions prepared by various computer program languages to be executed and create or change the relationship between the logical screen and the display screen by using the above information as parameters.

Further, services provided by various service sources may be displayed on a display screen, and the display screen may be displayed on a physical display device, as illustrated in FIG. 8.

There are service sources which provide broadcast services such as a terrestrial broadcaster 320 and a cable broadcaster 330, service sources which provide services stored in a storage medium such as a personal video recorder (PVR) 340, and service sources (not illustrated in FIG. 8) which provide services via a wired network or a wireless network.

A broadcast receiver 310 receives services from the service sources and produces logical screens displaying each of the received services.

Then, an arbitrary service is directly set on the display screen to be displayed on a physical display device using a predefined method or a method set by a user or an application. Otherwise, at least one logical screen that is mapped to an arbitrary area on the display screen is displayed on a physical display device 350. In short, services provided by the terrestrial broadcaster 320, the cable broadcaster 330, and the PVR are displayed on the physical display device 350.

The terrestrial broadcaster 320, the cable broadcaster 330, and the PVR 340 are illustrated in FIG. 8 as being service sources, but the present invention is not limited to it. Any type of multimedia content source which provides multimedia contents that can be displayed together can be a service source according to an exemplary embodiment of the present invention.

Services according to an exemplary embodiment of the present invention can be classified into abstract services and non-abstract services, as illustrated in FIGS. 9 and 10.

The abstract services are not services provided by broadcast signals transmitted in real time but services independent of broadcast channels. The abstract services include only data components, i.e., application programs, without video components and audio components. Examples of the abstract services include services having unbound applications based on the open cable application platform (OCAP) standard.

The non-abstract services are understood as services other than abstract services.

According to the current embodiment of the present invention, both abstract services and non-abstract services have independency. For example, abstract services may be directly set on the physical display device not through logical screens and non-abstract services may be displayed on the logical screens. Then, the logical screens may be mapped to the display screen in which the abstract services are set. Thereafter, the display screen may be output through the physical display device. By doing so, the abstract services can be displayed on the display screen independently of the non-abstract services. In addition, the abstract services and non-abstract services may be mapped to different logical screens. Thereafter, the logical screens may be mapped to a single display screen. In other words, the abstract services can be displayed on the display screen independently of non-abstract services.

According to the current embodiment of the present invention, the logical screen and the display screen may be categorized as being different objects. Alternatively, a screen may serve as a logic screen or a display screen according to attribute information of one screen object.

In detail, if screen object attributes corresponding to the characteristics of a logical screen has a predefined value and other screen object attributes have fixed values, a screen object may serve as a logical screen. On the other hand, if screen object attributes corresponding to the characteristics of a display screen have predefined values and other screen object attributes have fixed values, a screen object may serve as a display screen.

Attribute information of the screen object includes a plurality of attributes 'z-Order', 'Display_Area', 'Visibility', 'Associated_Display_Screen', 'Associated_Service_Contexts', 'Associated_Logical_Screens', and 'OutputPort'.

If the attributes 'z-Order', 'Display_Area', 'Visibility', 'Associated_Display_Screen', and 'Associated_Service_Contexts' have predefined values

and the attributes 'Associated_Logical_Screens', and 'OutputPort' have fixed values, a screen object may serve as a logical screen.

On the other hand, if the attributes 'Associated_Service_Contexts', 'Associated_Logical_Screens', and 'OutputPort' have predefined values and the attributes 'z-Order', 'Display_Area', 'Visibility', and 'Associated_Display_Screen' have fixed values, a screen object may serve as a display screen.

FIG. 11 illustrates attribute information and interfaces of a logical screen and a display screen.

Referring to FIG. 11, an attribute 'Type' 510 is for determining a screen type – a logical screen or a display screen.

Referring to FIG. 11, an attribute 'z-Order' 520 is for determining in what order a plurality of logical screens are arranged along the z-axis. FIG. 12 illustrates different configurations of logical screens on a physical display device for different combinations of the values of attributes 'z-Order' of the logical screens.

Referring to FIG. 12, first and second logical screens 620 and 630 are respectively mapped to predetermined areas of a display screen 610. In detail, the first logical screen 620 is displayed on the display screen 610, and the second logical screen 630 is displayed on the display screen partially overlapping the first logical screen 620. In other words, the display screen 610, the first logical screen 620, and the second logical screen 630 are sequentially arranged in the direction of the z-axis. In this case, an attribute 'z-Order' of the first logical screen 620 may be set to a value of 1, and an attribute 'z-Order' of the second logical screen 630 may be set to a value of 2. The attributes 'z-Order' of the first and second logical screens 620 and 630 may be set to any numbers or characters as long as they can represent a certain order in which the first and second logical screens 620 and 630 are to be arranged along the z-axis.

An attribute 'Display_Area' 520 is information regarding a display screen area of a logical screen, as to be illustrated in FIGS. 13 and 14.

FIG. 13 illustrates that a logical screen 710 is mapped to an entire area of the display screen 720, and FIG. 14 illustrates that a logical screen 730 is mapped to a partial area of the display screen 740.

The attribute 'Display_Area' may include information specifying the 2-dimensional coordinates of a predetermined portion of a display screen to which the

logical screen is to be mapped or may include information specifying a predetermined location on the display screen and an offset value indicating how much the logical screen deviates from the predetermined location on the display screen.

An attribute 'Visibility' 530 determines whether a logical screen is to be visibly or invisibly displayed on a display screen. It is possible to make a logical screen appear on or disappear from a display screen by altering the value of the attribute 'Visibility' 530.

An attribute 'Associated_Display_Screen' 540 is information regarding display screens associated with a logical screen. A logical screen which is not associated with any display screens may not be displayed on a physical display device nor be transmitted to external output devices.

An attribute 'Associated_Service_Contexts' 550 is information regarding service contexts connected to a logical screen or a display screen. Services set in such service contexts may be displayed on a logical screen or a display screen.

An attribute 'Associated_Logical_Screens' 560 is information regarding logical screens associated with a display screen.

An attribute 'OutputPort' 570 is information regarding devices by which a display screen is to be output, and such devices include display screens, wired/wireless communication media, and various storage media.

Interfaces for identifying or altering the values of the attributes illustrated in FIGS. 6A and 6B may be provided. Referring to FIGS. 6A and 6B, the interfaces may include an interface 'SET' for setting attribute values or connecting a logical screen to a display screen, an interface 'ADD' for adding attribute values or connecting a logical screen to a service, an interface 'GET' for identifying attribute values, and an interface 'REMOVE' for deleting attribute values. These interfaces may include processes, functions, procedures, or methods that perform their functions, respectively.

For example, a method 'getDisplayScreen(void)' returns a display screen associated with the current screen. In detail, if the current screen is a logical screen, the method 'getDisplayScreen(void)' returns the associated display screen. If the current screen is display screen, the method 'getDisplayScreen(void)' returns reference information regarding the current screen. Further, if the current screen is a logical screen, but there is no associated screen, the method 'getDisplayScreen(void)' returns a value of 'NULL'.

According to another example, a method 'public void setDisplayArea(HScreenRectangle rect) throws SecurityException, IllegalStateException' provides a function for mapping the current logical screen to a predetermined area of the associated display screen. An instance that is provided as a parameter is of a class 'HScreenRectangle' of a package 'org.havi.ui', and has 2-dimensional position information. The execution of the methods 'SecurityException' and 'IllegalStateException' may be conducted as an exceptional operation for the method 'setOutputScreen(HScreen screen)'. The method 'IllegalStateException' may be executed when the current screen is a logical screen or when a portion of a display screen associated with a current logical screen cannot change due to the characteristics of a host platform.

According to still another example, a method 'getOutputArea(void)' returns regional information of a current screen as HScreenRectangle information. If the current screen corresponds to a display screen, the method 'getOutputArea(void)' returns HScreenRectangle information having the same value as HScreenRectangle (0,0,1,1). If the current screen is a logical screen, the method 'getOutputArea(void)' returns information regarding an area on a display screen occupied by the current screen. If the current screen is a logical screen but is not associated with any display screen, the method 'getOutputArea(void)' returns a value 'NULL'.

Certain terms are used throughout the following description to refer to particular interfaces. However, one skilled in the art will appreciate that a particular function is named just to indicate its functionality. This document does not intend to distinguish between functions that differ in name but not function.

FIG. 15 is a diagram illustrating a process that two services are set on a display screen to be mapped to a single screen.

Referring to FIG. 15, a first service includes all the three service components, i.e., video, audio, and data components, and a second service includes only video and audio components. However, the present invention does not impose any restrictions on service components, and the first and second services illustrated in FIG. 8 are exemplary.

As illustrated in FIG. 15, the first and second services are displayed on a physical display device in almost the same manner as in the related art. According to the current embodiment of the present invention, it is possible to display a plurality of services on a

physical display device independently of one another without imposing any restrictions on the number of services that can be displayed on a single display screen.

FIG. 16 is a block diagram of an apparatus for providing multiple screens according to an exemplary embodiment of the present invention.

Referring to FIG. 16, an apparatus 900 for providing multiple screens includes a digital signal processing module 940, a service processing module 950, an output module 960, and a user interface module 965.

Also, the apparatus 900 includes a broadcast signal reception module 910, a storage medium 920, and an external input module 930 as service sources, and includes a display screen 970, a storage medium 980, and an external output module 990 as service output media.

The term 'module', as used herein, means, but is not limited to, a software or hardware component, such as a Field Programmable Gate Array (FPGA) or Application Specific Integrated Circuit (ASIC), which performs certain tasks. A module may advantageously be configured to reside on the addressable storage medium and configured to execute on one or more processors. Thus, a module may include, by way of example, components, such as software components, object-oriented software components, class components and task components, processes, functions, attributes, procedures, subroutines, segments of program code, drivers, firmware, microcode, circuitry, data, databases, data structures, tables, arrays, and variables. The functionality provided for in the components and modules may be combined into fewer components and modules or further separated into additional components and modules.

The digital signal processing module 940 receives various information of a service such as a multimedia content, e.g., video information, audio information, or data information, from the broadcast signal reception module 910, the storage medium 920, or the external input module 930.

The broadcast signal reception module 910 receives a satellite, terrestrial, or cable broadcast signal and transmits the received broadcast signal, the storage medium 920 stores video information, audio information, or data information of a service, and the external input module 930 receives video information, audio information, or data information of a service from an external device such as a network interface module connected to a network.

The digital signal processing module 940 restores a plurality of services using received service components. The restored services include abstract or non-abstract services.

Here, the 'a plurality of services' refers to two or more services transmitted by the broadcast signal reception module 910 or two or more services respectively transmitted by the broadcast signal reception module 910 and the storage medium 920.

The digital signal processing module 940 may restore services according to selection by a user or an application with the aid of the user/application interface module 965. In this case, the user or the application may select the connection between an arbitrary service and a screen.

The service processing module 950 produces a logical screen to display a service restored by the digital signal processing module 940.

The output module 960 maps a plurality of logical screens produced by the service processing module 950 to the display screen. The mapping of the logical screens to the display screen may be conducted using a predefined method or a method set by the user with the aid of the user/application interface module 965.

A service restored by the digital signal processing module 940 may not be processed by the service module 950. Instead, a service restored by the digital signal processing module 940 may be directly mapped to a certain portion of a display screen produced by the output module 960.

A display screen provided by the output module 960 may be displayed on the physical display device 970 or may be stored in the storage medium 980. Examples of the storage medium 980 include computer readable floppy discs, hard discs, CD-ROM. DVD, DVD-ROM, BD(Blu-ray Disc), and semiconductor memories.

Also, a display screen provided by the output module 960 may be transmitted to an external device connected to a network via the external output module 990.

For this, the output module 960 may include a plurality of output ports via which a display screen can be provided. In this case, a display screen can be provided via an output port set in advance as a default or an output port chosen by the user with the aid of the user interface module 965.

The user or the application can choose one of a plurality of services or restore desired services using the user interface module 965. Also, the user can choose one of a plurality of display screens using the user interface module 965.

Since the modules illustrated in FIG. 16 are divided according to their functions, it is possible to be connected to the other modules.

FIG. 17 is a flowchart illustrating a method of dynamically configuring multiple screens according to an exemplary embodiment of the present invention.

In general, video information, audio information, and data information constituting a multimedia content are transmitted in a predetermined format, for example, an MPEG stream format. In operation S1010, an apparatus for providing a service such as a multimedia content service receives video information, audio information, and data information and restores a service based on the video information, the audio information, and the data information. Here, the service restored in operation S1010 may be selected or previously determined by a user or an application. The user may use a menu displayed on the display device or a remote controller to select the connections between an arbitrary screen and a screen. The application may select the connections using an API.

Further, data information includes application information regarding application program for a service, and these application information includes signal information indicating whether the application program can be executed on a PiP screen. Examples of the application information include application information table (AIT) based on MHP standard and eXtended application information table (XAIT) based on OCAP standard. The signal information may be added to the application.

Thereafter, in operation S1020, the restored service is set such that it can be displayed on a logical screen. In operation S1030, the logical screen is mapped to a display screen. In operation S1040, the display screen is provided to the user using a display screen, a storage medium, or a network.

The restored service is illustrated in FIG. 17 as being displayed on a physical display device via a logical screen. However, the restored service may be directly displayed on a physical display device without passing through the logical screen.

When the user selects the PiP service, the PiP service is realized in two modes. In the first mode, only video component for PiP service selected on the main screen is

provided without creating a separate logical screen for PiP service, that is, PiP screen. In the second mode, a separate logical screen for PiP service is created to provide the PiP service selected on the created PiP screen.

FIG. 17 illustrates a method of mapping only one service to a display screen for simplicity. However, a plurality of services may be mapped to a display screen with or without passing through a plurality of logical screens.

When a display screen is provided to the user in this manner, the user can perform a plurality of services.

FIG. 18 is a diagram illustrating a software architecture for providing multiple screens according to an exemplary embodiment of the present invention.

Referring to FIG. 18, a software architecture 1100 includes a device driver layer 1110, an application program interface (API) layer 1120, and an application layer 1130.

The device driver layer 1110 receives service components from various multimedia content sources and decodes the received service components. Examples of the received service components include video information, audio information, and data information.

The API layer 1120 generates a logical screen and a display screen and maps a service, the logical screen, and the display screen to one another.

The application layer 1130 provides a user interface so that a user can dynamically configure a logical screen which displays a service or transmits a user command to the API layer 1120 so that the API layer 1120 can execute the user command.

The user enables the device driver layer 1110 with the aid of the application layer 1130 to provide a display screen via a physical display device or to store the display screen in a storage medium. In addition, the user can enable the device driver layer 1110 to transmit a display screen to an external device via a network.

For this, the device driver layer 1110 may include a plurality of output ports which can provide a display screen. Otherwise, API layer 1120 may include the plurality of output ports.

In order to dynamically configure a plurality of logical screens on a display screen, the API layer 1120 may include a plurality of software modules, e.g., a multiscreen manager module 'MultiScreenManager' 1210, a multiscreen context module 'MultiScreenContext' 1230, a multiscreen context listener module

'MultiScreenContextListener' 1250, and a multiscreen context event module 'MultiScreenContextEvent' 1240, as illustrated in FIG. 19.

The multiscreen manager module 1210 manages the multiscreen context module 1230, searches for a desired screen, displays information specifying what devices are shared by screens, registers the multiscreen context listener module 1250, or cancels the registration of the screen context listener module 1250.

The multiscreen context module 1230 is an interface object associated with a screen object 1220 and determines whether the screen object 1220 is to become a logical screen or a display screen according to an interface operation performed by the multiscreen context module 1230. Various attributes such as the attributes 510 through 570 illustrated in FIGS. 6A and 6B may be set in the multiscreen context module 1230. The multiscreen context module 1230 can provide the functions 'SET', 'ADD', 'GET', and 'REMOVE' described above with reference to FIGS. 6A and 6B.

When attribute information of the screen object 1220 is altered by the multiscreen context module 1230, the multiscreen context event module 1240 serves as an event class announcing that the attribute information of the screen object 1220 has been changed, and the multiscreen context listener module 1250 serves as a listener interface object which can be realized in a predetermined application class which attempts to receive an event prompted by the multiscreen context event module 1240.

An application 1260 is a module which is driven on the application layer 1130. The application 1260 allows the user to choose a desired service and to freely arrange a plurality of logical screens on a display screen.

In detail, the application 1260 transmits various commands which allow the user to dynamically configure and manage logical screens to the multiscreen manager module 1210, and the multiscreen manager module 1210 controls operations corresponding to the various commands to be executed through the multiscreen context module 1230.

The multiscreen context module 1230 is associated with the screen object 1220 and manages the attribute information of the screen object 1220 illustrated in FIGS. 6A and 6B. In order to manage the attribute information of the screen object 1220, the multiscreen context module 1230 may include a variety of functions or methods.

The multiscreen manager module 1210 (not illustrated) receives service components provided by various service sources from the device driver layer 1110 and

performs operations to display the received service components on a logical screen or a display screen.

FIG. 20 is a flowchart illustrating a method of displaying a plurality of services displayed on respective corresponding logical screens by the modules illustrated in FIG. 19 on a display screen according to an exemplary embodiment of the present invention.

Referring to FIG. 20, in operation S1310, the multiscreen manager module 1210 produces a display screen and a number of logical screens corresponding to the number of services to be performed.

In operation S1320, the multiscreen manager module 1210 connects the logical screens to respective corresponding services received from the device driver layer 1110. The multiscreen manager module 1210 may call a method 'addServiceContext' for each of the logical screens by setting service context objects of the received services as parameters for the logical screens services. The method 'addServiceContext' connects a logical screen to a service and may be provided by the multiscreen context module 1230.

In operation S1330, once the logical screens are connected to the respective services, the multiscreen manager module 1210 connects the logical screens to the display screen. At this time, the multiscreen manager module 1210 may call a method 'setDisplayScreen' for each of the logical screens by setting a display screen object to which the logical screens are connected as a parameter. The method 'setDisplayScreen' connects a logical screen to a display screen and may be provided by the multiscreen context module 1230.

A method 'setDisplayScreensetDisplayScreen' may be set to 'public void setDisplayScreensetDisplayScreen(HScreen screen) throws SecurityException, Illegal StateException', and this method allows an instance 'HScreen' that is provided as a parameter to be associated with the current logical screen. In this case, the instance 'HScreen' is preferably a display screen.

A parameter of the method 'setDisplayScreen(HScreen screen)' may include a value of 'NULL'. In this case, when the method 'setDisplayScreen(HScreen screen)' is executed without exception handling, the current logical screen is no longer associated with the display screen.

The execution of the methods 'SecurityException' and 'IllegalStateException' may be conducted as an exceptional operation for the method 'setOutputScreen(HScreen screen)'.

The method 'IllegalStateException' may be executed when a current screen is a logical screen or when a portion of a display screen associated with a current logical screen cannot change due to the characteristics of a host platform.

In operation S1340, areas on the display screen to which the logical screens are to be respectively mapped are determined. At this time, a predetermined method provided by the multiscreen context module 1230 can be called to determine an area on the display screen where the logical screens are to be displayed.

FIG. 21 is a flowchart illustrating a method of exchanging services displayed on logical screens between the modules illustrated in FIG. 19 according to an exemplary embodiment of the present invention.

In operation S1410, the multiscreen manager module 1210 temporarily terminates a service to be exchanged between two logical screens.

Thereafter, in operation S1420, information regarding the service is exchanged between the two logical screens. At this time, a method 'removeServiceContext' and a method 'addServiceContext' are called for each of the two logical screens, thereby exchanging service contexts set in the two logical screens between the two logical screens. The method 'removeContext' removes a service context connected to a logical screen, and the method 'addServiceContext' adds a new service context to a logical screen. The methods 'removeServiceContext' and 'addServiceContext' may be provided by the multiscreen context module 1230. According to the current embodiment of the present invention, service information regarding a service connected to logical services is exchanged between the logical services, thereby obtaining the effect of exchanging a main screen and a sub-screen. In addition, according to the current embodiment of the present invention, even when 3 or more services are performed on a display screen, they can be exchanged between an arbitrary number of logical screens.

In operation S1430, occurrence of an event in which the connection between the service and the two logical screens has been changed is announced. Thereafter, in operation S1440, the multiscreen context event module 1240 transmits the event to the

multiscreen context listener module 1250, and a service newly connected to the two logical screens begins to be performed.

The same application may be executed by a plurality of service contexts belonging to different screens, and this will hereinafter be described in detail with reference to FIG. 22. For example, referring to FIG. 22, the same application may be executed on a main screen HScreen1 and on a PiP screen HScreen2.

It may be determined whether a plurality of applications respectively executed on a plurality of screens are the same applications or different applications by comparing application provider identification information Org_ID and application identification information App_ID of the applications. Referring to FIG. 22, the applications respectively executed on the main screen HScreen1 and the PiP screen HScreen2 share the same application provider identification information Org_ID and the same application identification information App_ID, thereby making it difficult to precisely determine whether an application currently being executed is being executed on the main screen HScreen1 or on the PiP screen HScreen2.

That is, additional information, other than Org_ID and App_ID, is necessary to precisely determine whether a plurality of applications respectively executed on a plurality of screens are the same applications or different applications, even though the applications share the same Org_ID and the same App_ID. A data structure of the additional information is illustrated in FIG. 23.

Referring to FIG. 23, a class 'OcapAppID' inherits a class 'AppID'. The class 'AppID' includes integer variables 'AID' and 'OID' and functions 'getAID()' and 'getOID()' which return the values of the variables 'AID' and 'OID'.

The class 'OcapAppID', which inherits the class 'AppID', includes a variable 'sc' which identifies a service context and a function 'getServiceContext()' which returns the value of the variable 'sc'. The variable 'sc' may be interpreted as being pointer information or reference information that indicates a service context for executing an application.

Since the class 'OcapAppID' not only includes 'Org_ID' and 'App_ID' but also includes identification information of a service context for executing an application, it is possible to identify each of a plurality of instances of an application even when the

application is executed on a plurality of screens. This will hereinafter be described in further detail with reference to FIG. 24.

Referring to FIG. 24, assuming that application A has an Org_ID value of '0xaaaaaaaa' and an App_ID value of '0x4000', a class 'OcapApp_ID_1' and a class 'OcapApp_ID_2' have the same Org_ID and App_ID values.

However, since a variable SC_1 of the class 'OcapApp_ID_1' includes identification information of a service context 'Service Context_1' and a variable SC_2 of the class 'OcapApp_ID_2' includes identification information of a service context 'Service Context_2', it is possible to identify each of a plurality of instances of an application even when the application is executed on a plurality of screens.

Industrial Applicability

According to the present invention, it is possible to perform a plurality of services provided by various sources such as cable broadcasts, terrestrial broadcasts, various storage media, and external inputs, in various manners using a single physical display screen.

In addition, according to the present invention, it is possible to identify each of a plurality of instances of an application in a multi-screen environment even when the application is executed on a plurality of screens.

While the present invention has been particularly illustrated and described with reference to exemplary embodiments thereof, it will be understood by those of ordinary skill in the art that various changes in form and details may be made therein without departing from the spirit and scope of the present invention as defined by the following claims. Therefore, it is to be understood that the above-described embodiments have been provided only in a descriptive sense and will not be construed as placing any limitation on the scope of the invention.

CLAIMS

1. An apparatus for identifying an application in a multi-screen environment, the apparatus comprising:

a digital signal processing module receiving video information, audio information, or data information and restoring a service based on the video information, the audio information or the data information;

a service processing module producing a plurality of logical screens for displaying the restored service; and

an output module mapping the plurality of logical screens provided by the service processing module to different locations on a display screen,

wherein an application present in the service is identified by a service context in which the application is executed and by identification information.

2. The apparatus of claim 1, wherein the identification information comprises information regarding an application provider that provides the application.

3. The apparatus of claim 1, wherein the identification information includes information regarding the application.

4. An apparatus for identifying an application in a multi-screen environment, the apparatus comprising:

a service processing module producing a logical screen displaying a service and a display screen that allows the logical screen to be associated with the service and displays the logical screen; and

an output module mapping the produced logical screen to a certain area on the produced display screen,

wherein an application in the service is identified by a service context in which the application is executed and identification information.

5. The apparatus of claim 4, wherein the identification information comprises information regarding an application provider that provides the application.

6. The apparatus of claim 4, wherein the identification information includes information regarding the application.

7. A method of identifying an application in a multi-screen environment, the method comprising:

receiving video information, audio information, or data information and restoring a service;

producing a plurality of logical screens displaying the restored service; and

mapping the logical screens to different locations on a display screen,

wherein an application in the service is identified by a service context in which the application is executed and identification information.

8. The method of claim 7, wherein the identification information comprises information regarding an application provider that provides the application.

9. The method of claim 7, wherein the identification information includes information regarding the application.

10. A method of identifying an application in a multi-screen environment, the method comprising:

producing a plurality of logical screens displaying a service and a display screen displaying the plurality of logical screen; and

mapping the logical screens to arbitrary areas on the display screen,

wherein an application in the service is identified by a service context in which the application is executed and identification information.

11. The method of claim 10, wherein the identification information comprises information regarding an application provider that provides the application.

12. The method of claim 10, wherein the identification information includes information regarding the application.

FIG. 1

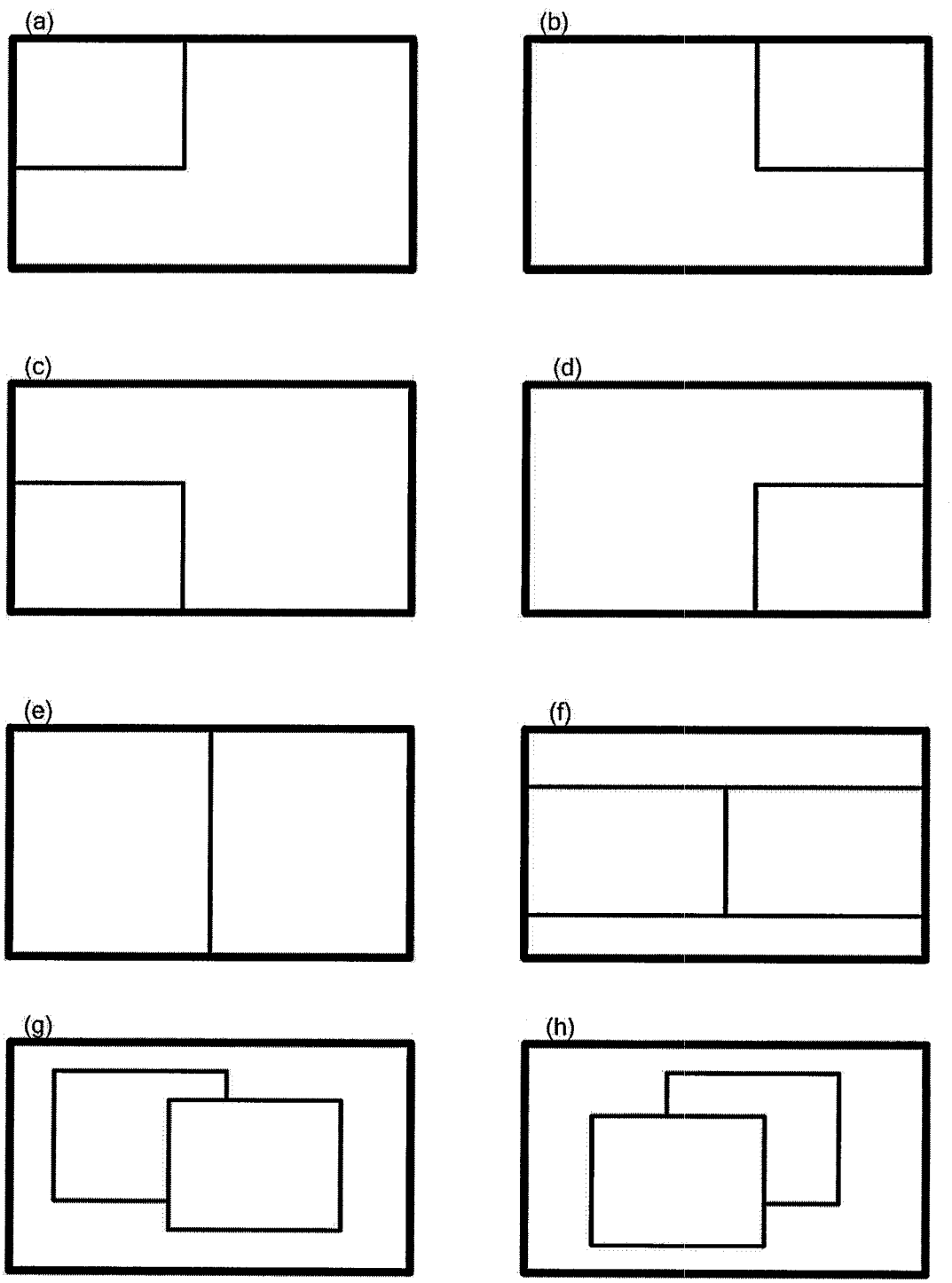


FIG. 2

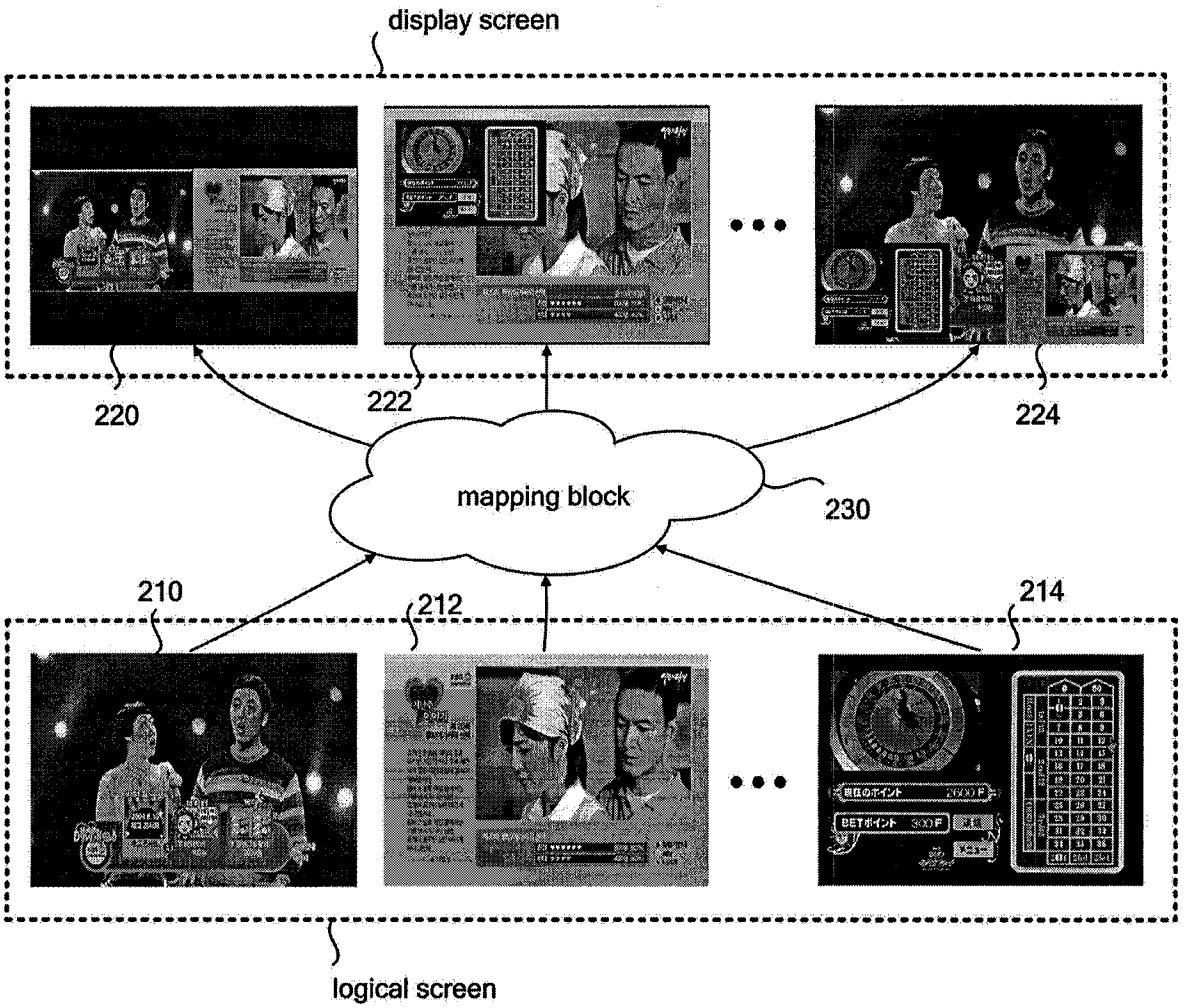

display screen
220
222
224
mapping block
230
210
212
214
logical screen

FIG. 3

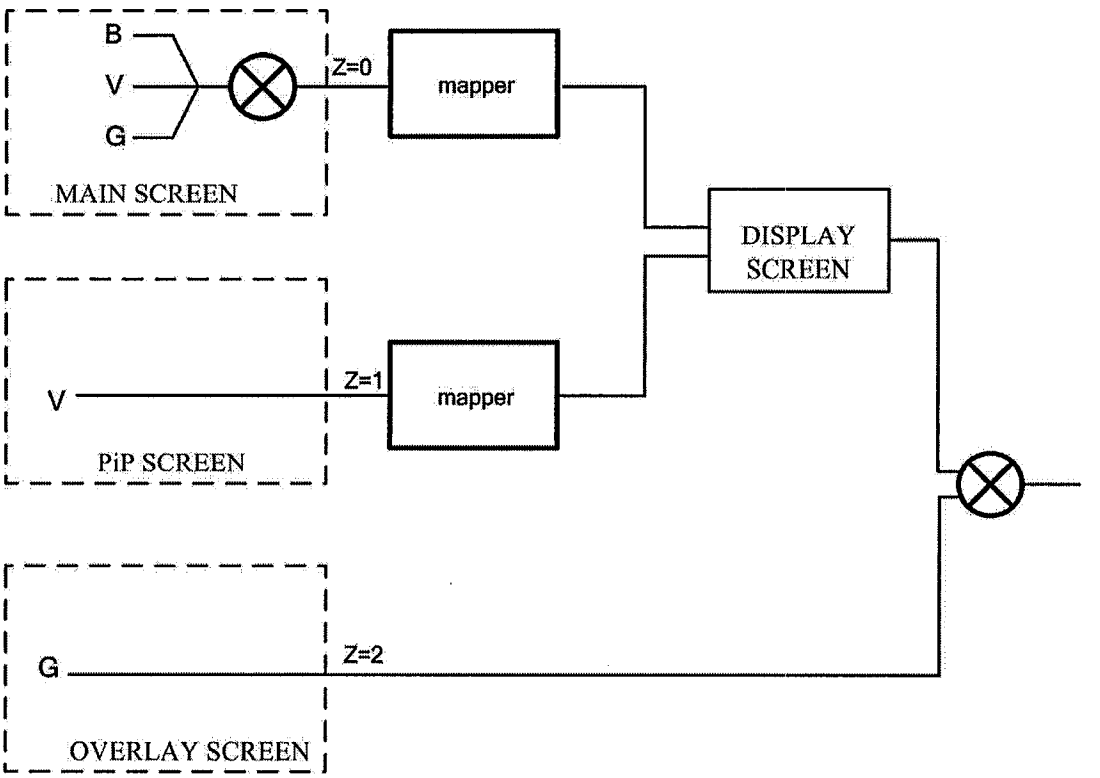

B
V
G
Z=0
mapper
MAIN SCREEN
DISPLAY SCREEN
V
Z=1
mapper
PiP SCREEN
G
Z=2
OVERLAY SCREEN

FIG. 4

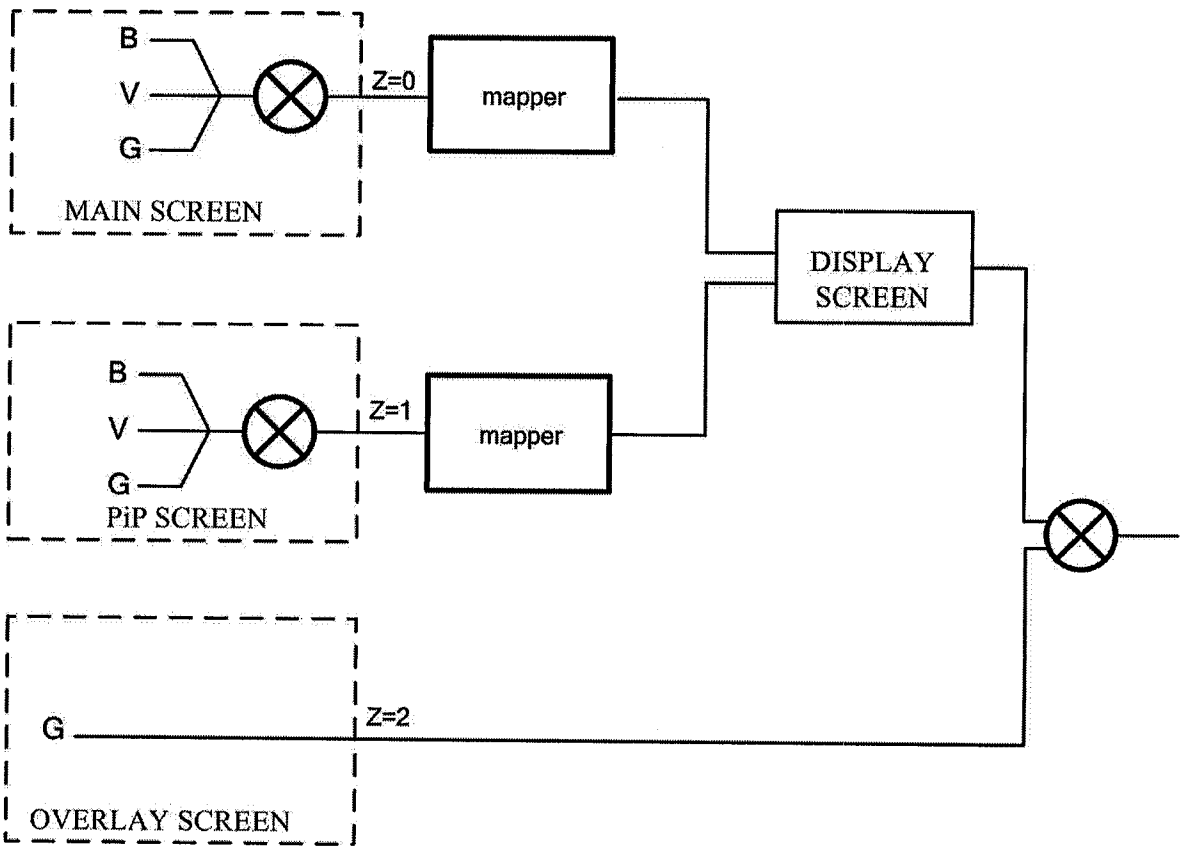

B
V
G
MAIN SCREEN
Z=0
mapper
DISPLAY SCREEN
B
V
G
PiP SCREEN
Z=1
mapper
G
Z=2
OVERLAY SCREEN

FIG. 5

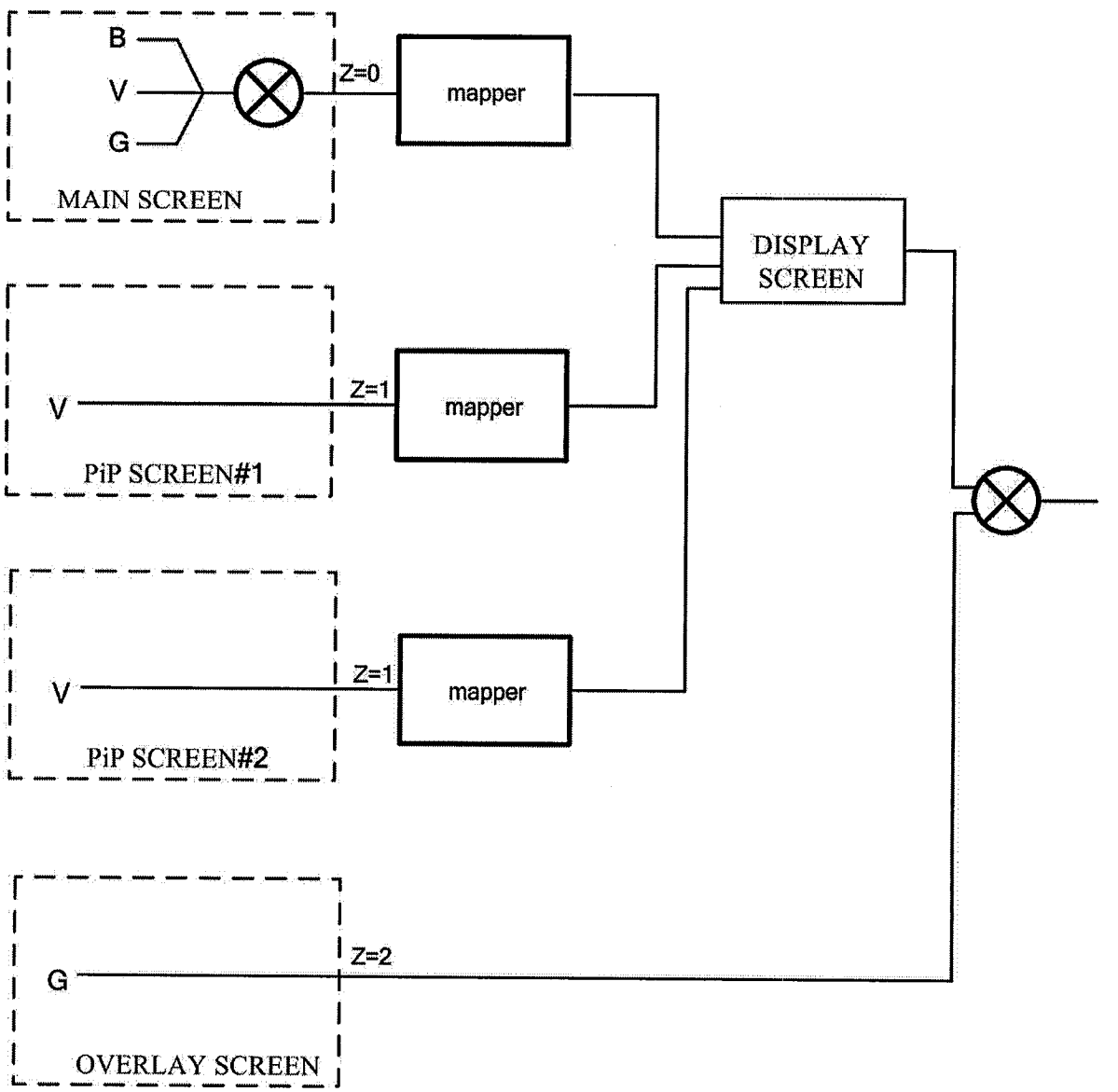

B
V
G
Z=0
mapper
MAIN SCREEN
DISPLAY SCREEN
V
Z=1
mapper
PiP SCREEN#1
V
Z=1
mapper
PiP SCREEN#2
G
Z=2
OVERLAY SCREEN

FIG. 6

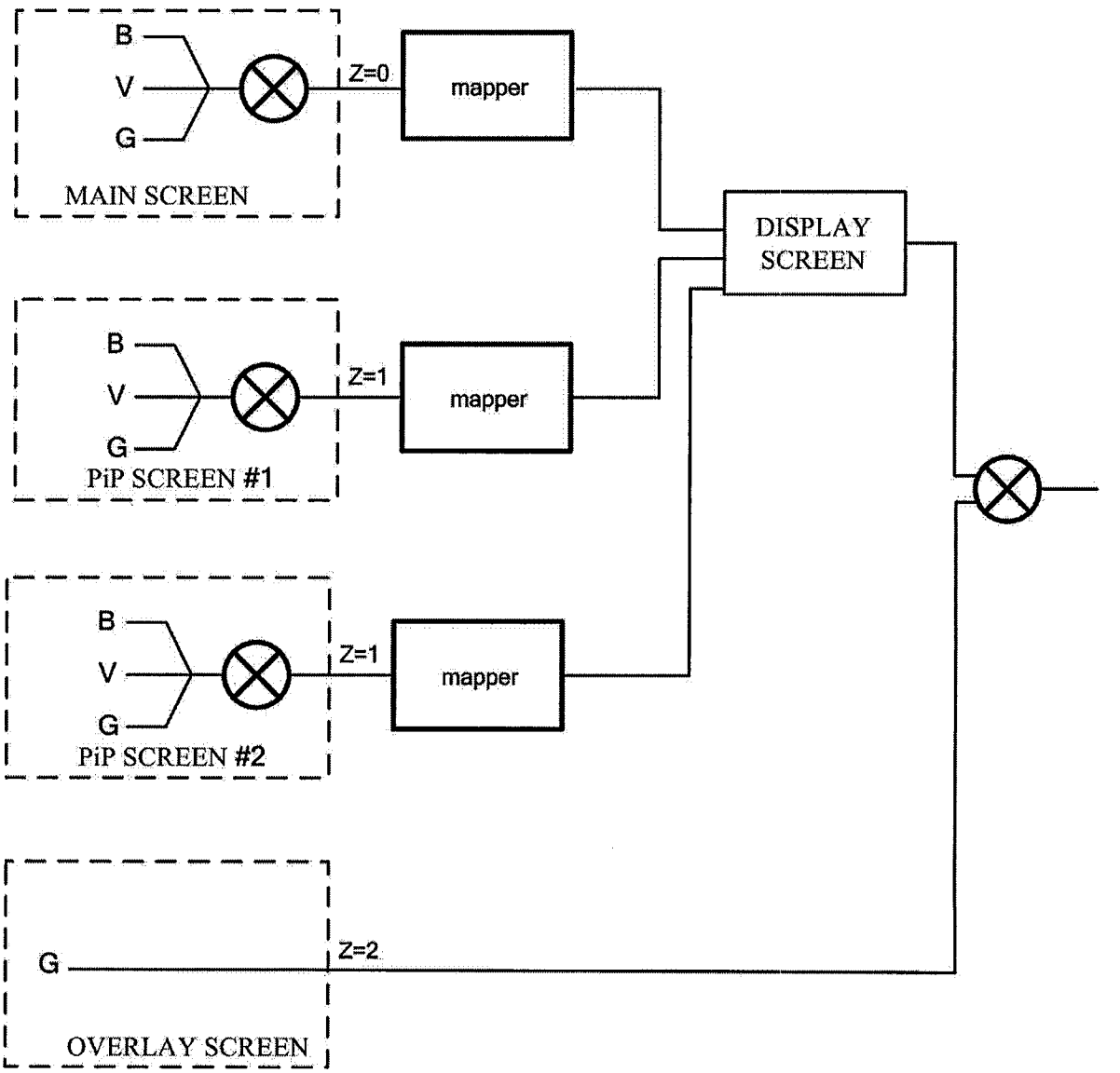

B
V
G
MAIN SCREEN
Z=0
mapper
DISPLAY SCREEN
B
V
G
PiP SCREEN #1
Z=1
mapper
B
V
G
PiP SCREEN #2
Z=1
mapper
G
OVERLAY SCREEN
Z=2

FIG. 7

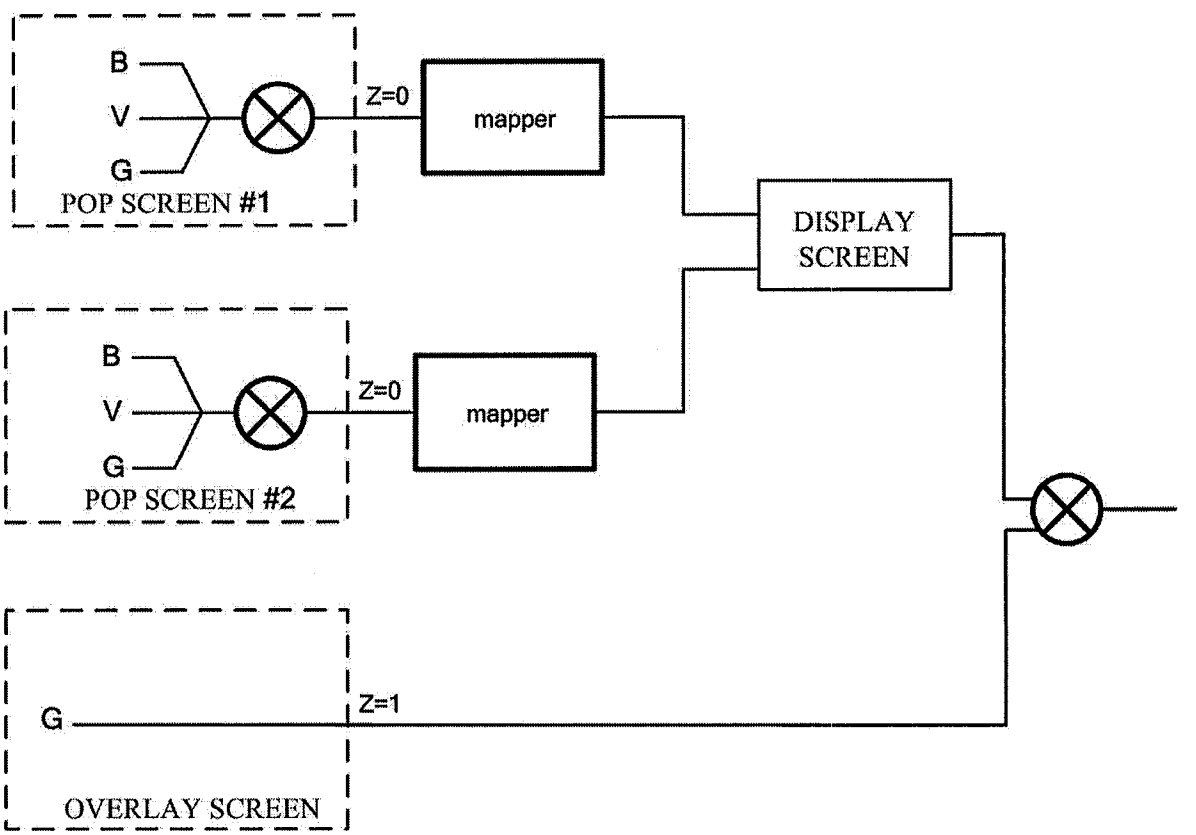

B
V
G
POP SCREEN #1
Z=0
mapper
DISPLAY SCREEN
B
V
G
POP SCREEN #2
Z=0
mapper
G
OVERLAY SCREEN
Z=1

FIG. 8

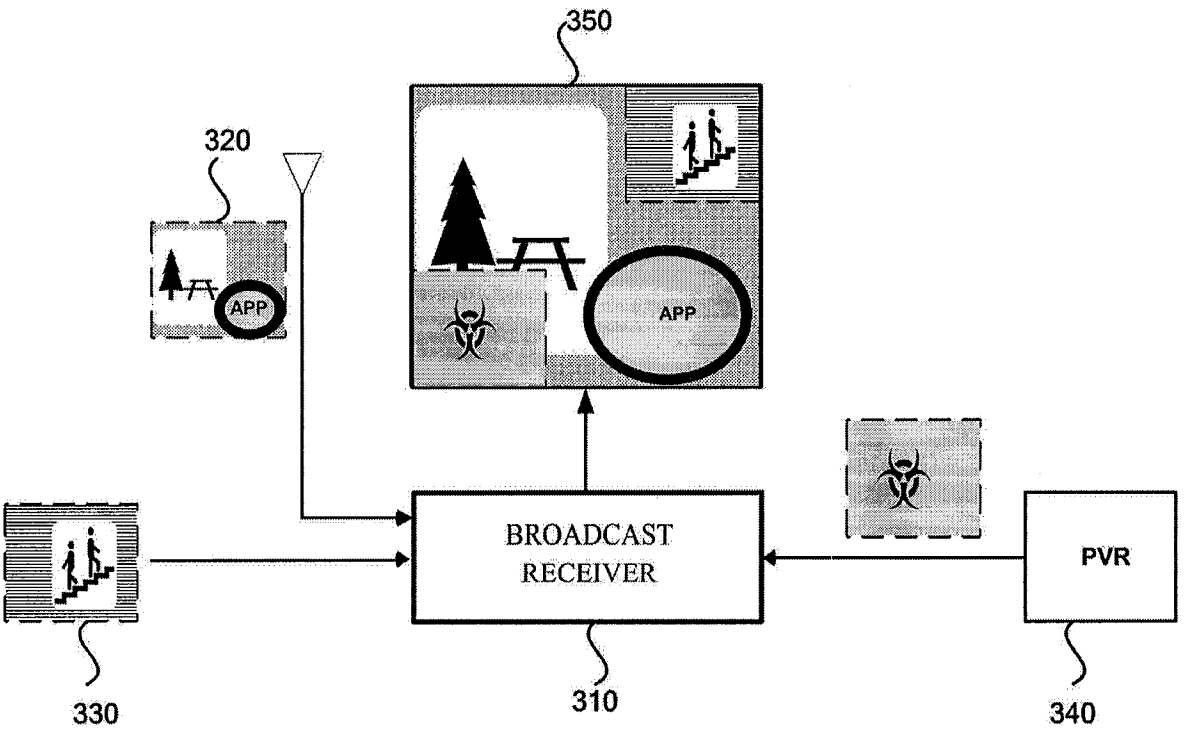


FIG. 9

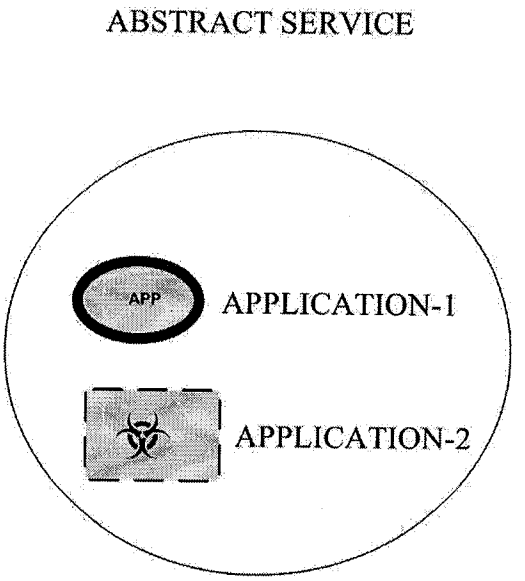

ABSTRACT SERVICE
APP
APPLICATION-1
APPLICATION-2

FIG. 10

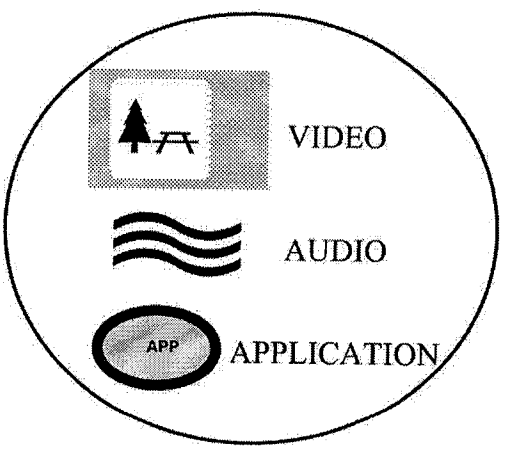

NON-ABSTRACT SERVICE
VIDEO
AUDIO
APP
APPLICATION

FIG. 11

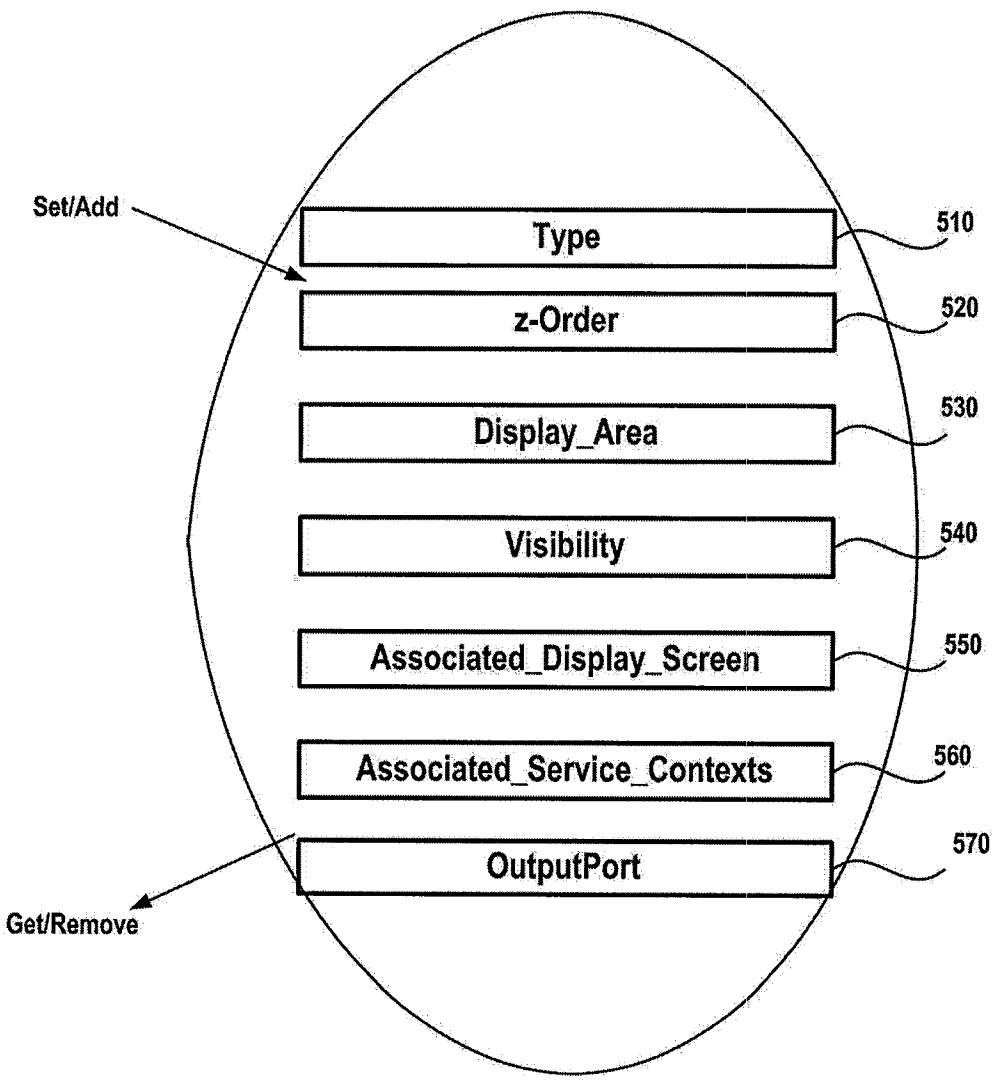

Set/Add
Type
510
z-Order
520
Display_Area
530
Visibility
540
Associated_Display_Screen
550
Associated_Service_Contexts
560
OutputPort
570
Get/Remove

FIG. 12

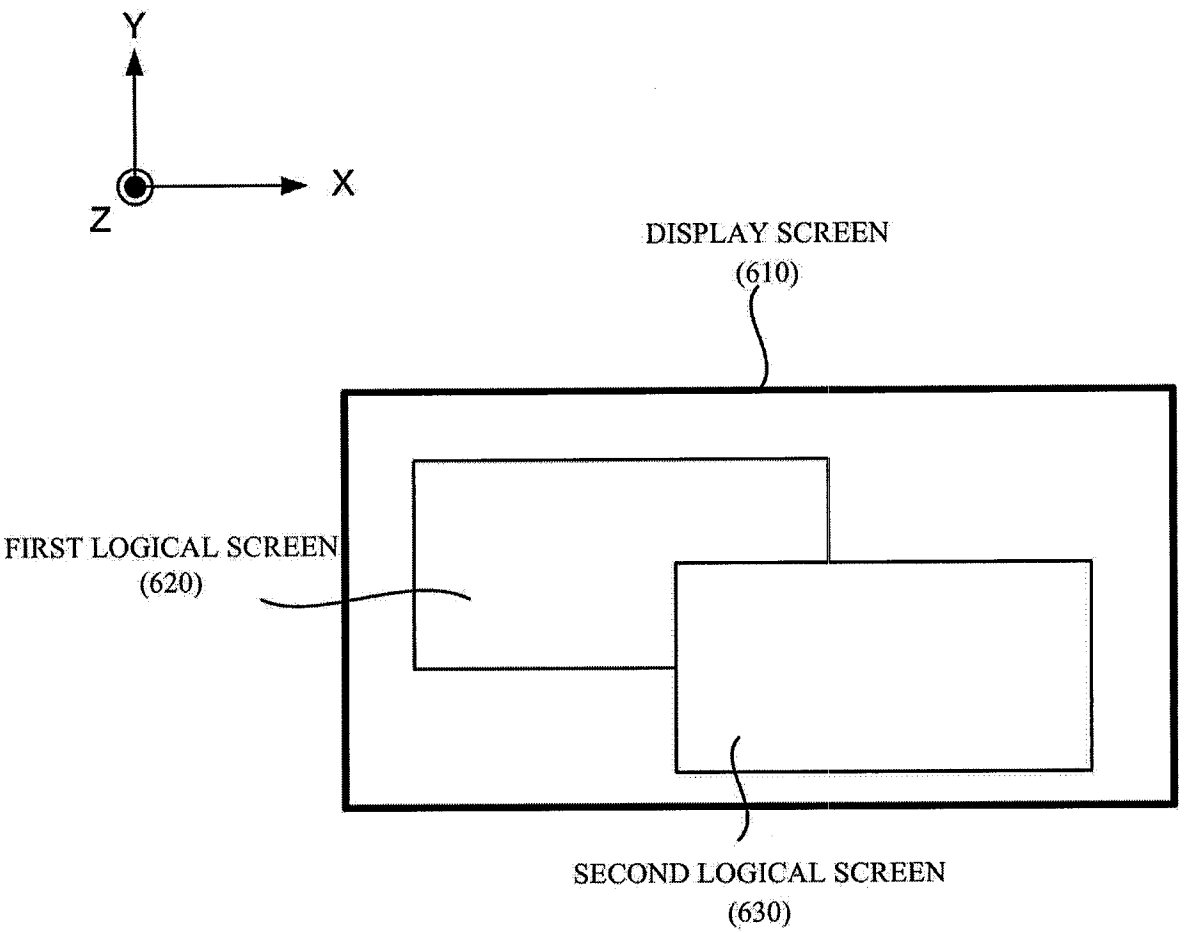

Y
X
Z
DISPLAY SCREEN
(610)
FIRST LOGICAL SCREEN
(620)
SECOND LOGICAL SCREEN
(630)

FIG. 13

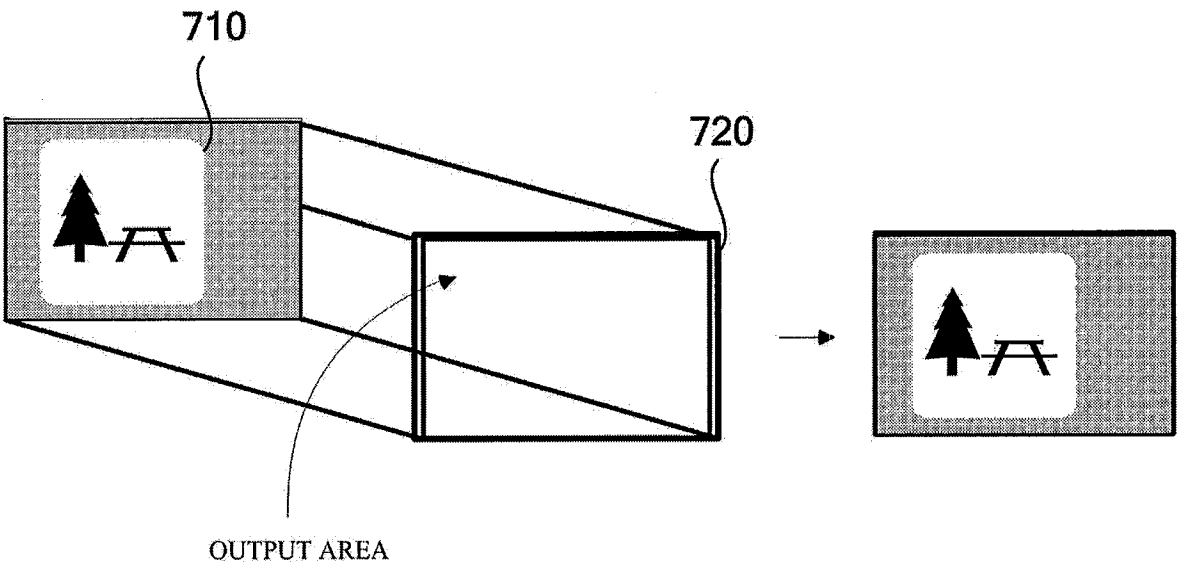


FIG. 14

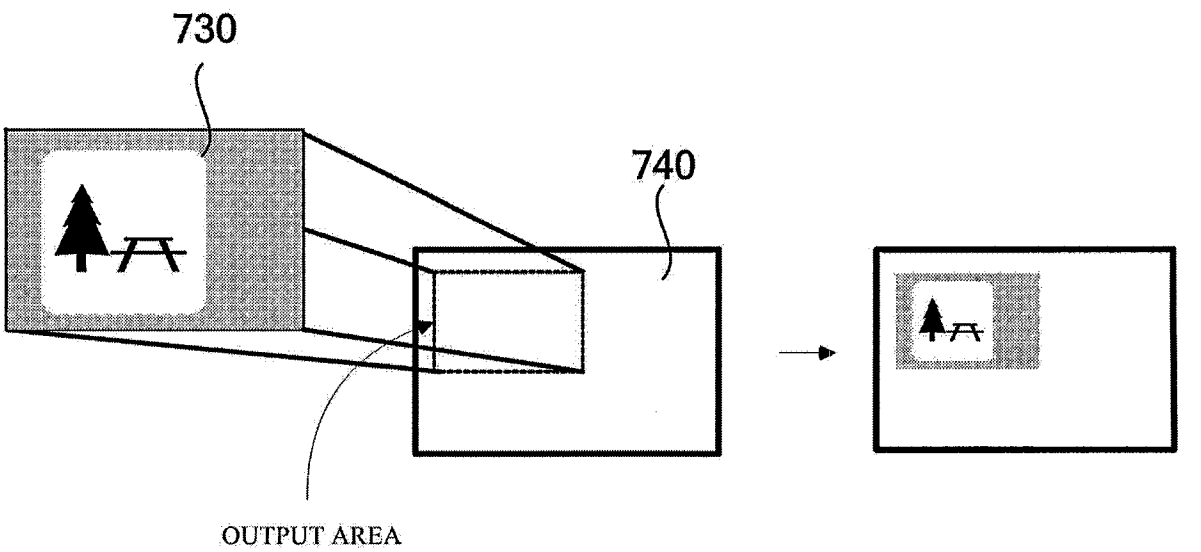


FIG. 15

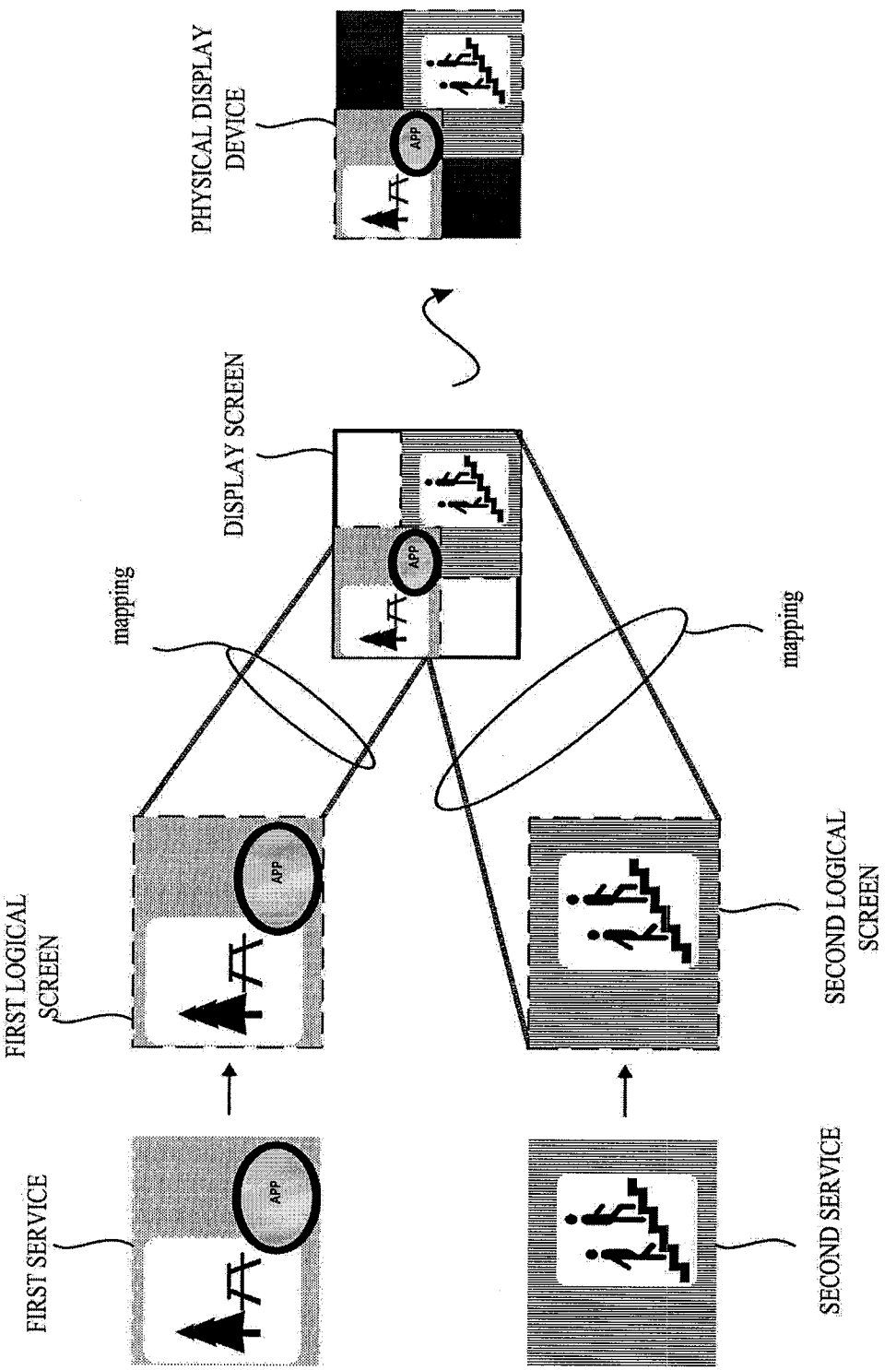

PHYSICAL DISPLAY DEVICE
APP
DISPLAY SCREEN
APP
mapping
mapping
FIRST LOGICAL SCREEN
APP
SECOND LOGICAL SCREEN
FIRST SERVICE
APP
SECOND SERVICE

FIG. 16

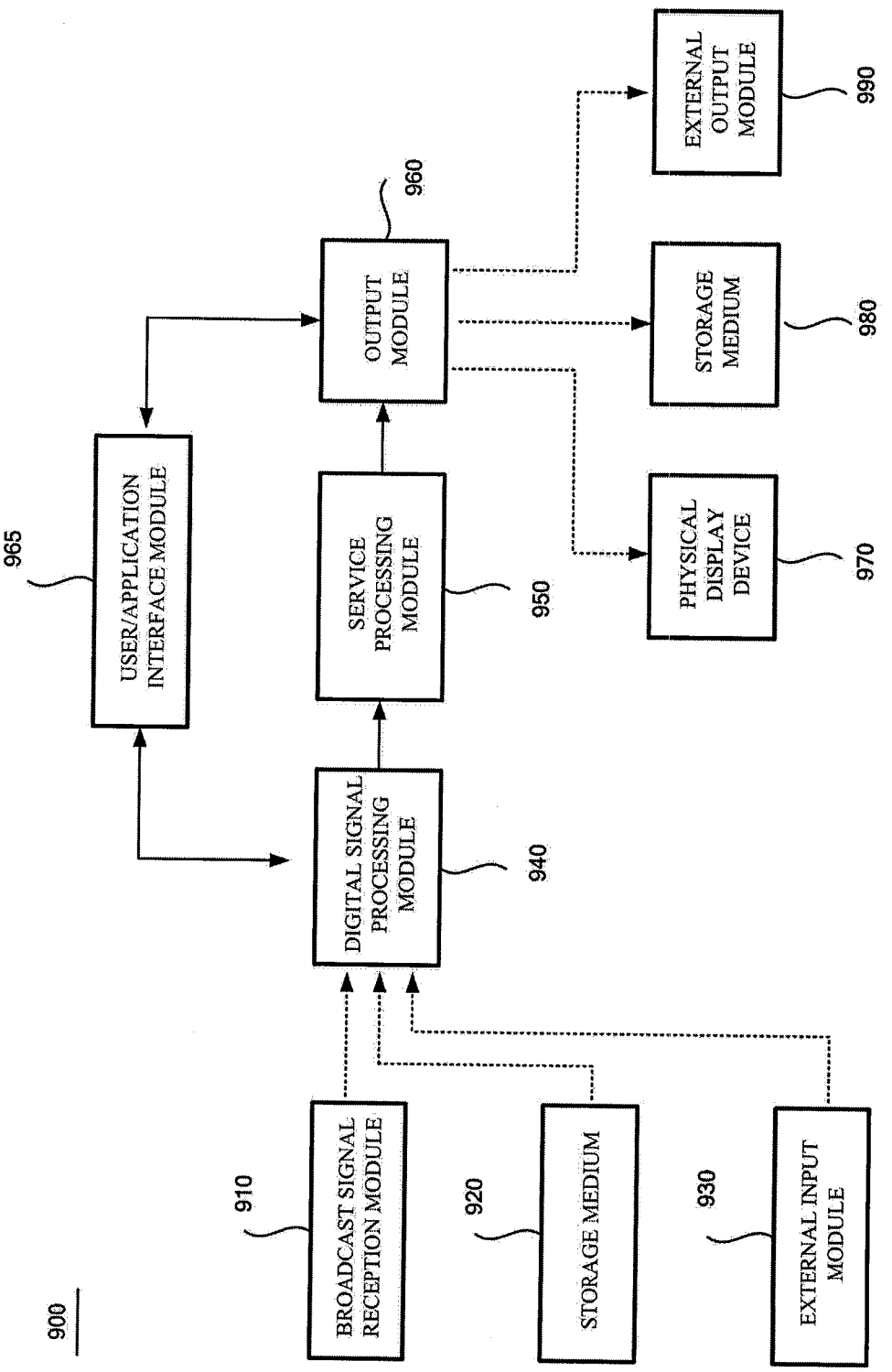

900
BROADCAST SIGNAL RECEPTION MODULE
910
STORAGE MEDIUM
920
EXTERNAL INPUT MODULE
930
DIGITAL SIGNAL PROCESSING MODULE
940
SERVICE PROCESSING MODULE
950
USER/APPLICATION INTERFACE MODULE
965
OUTPUT MODULE
960
PHYSICAL DISPLAY DEVICE
970
STORAGE MEDIUM
980
EXTERNAL OUTPUT MODULE
990

FIG. 17

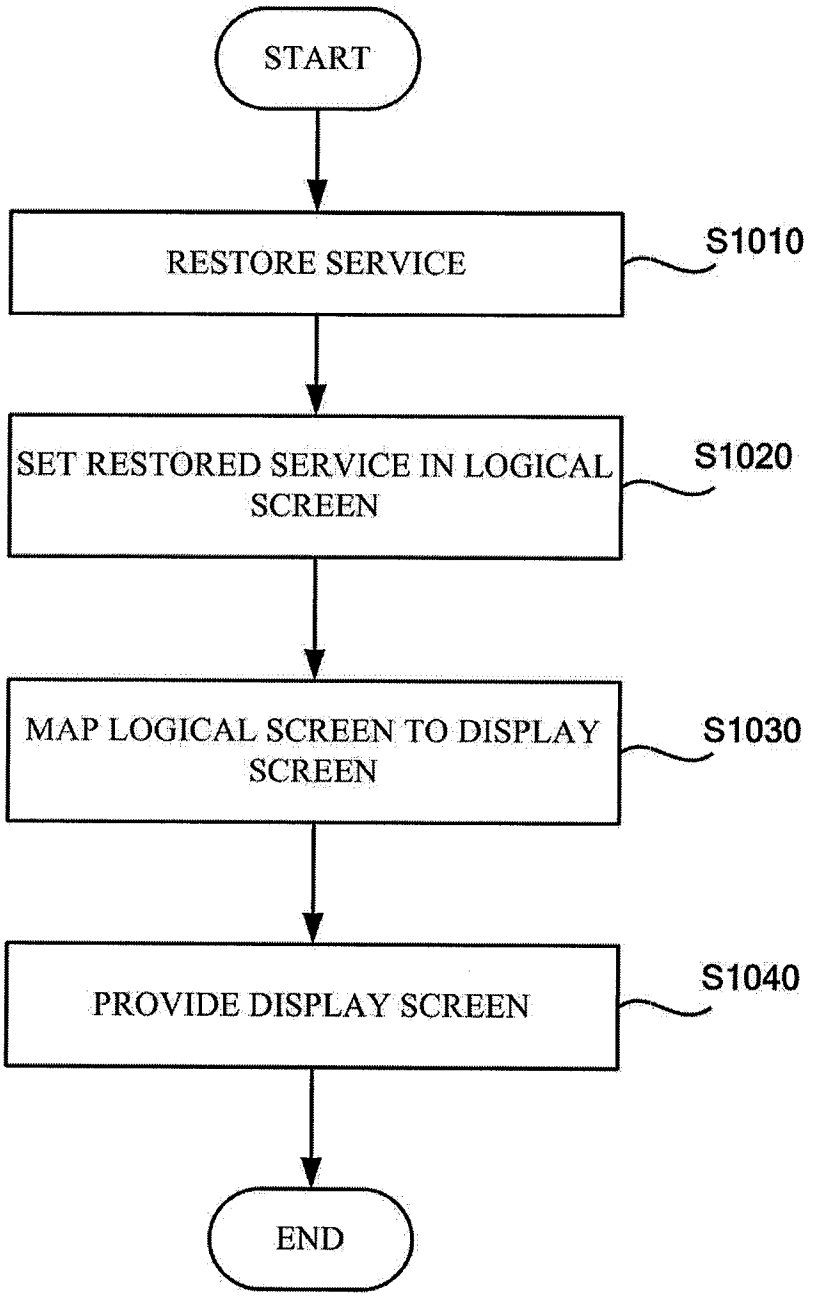

START
RESTORE SERVICE
S1010
SET RESTORED SERVICE IN LOGICAL SCREEN
S1020
MAP LOGICAL SCREEN TO DISPLAY SCREEN
S1030
PROVIDE DISPLAY SCREEN
S1040
END

FIG. 18

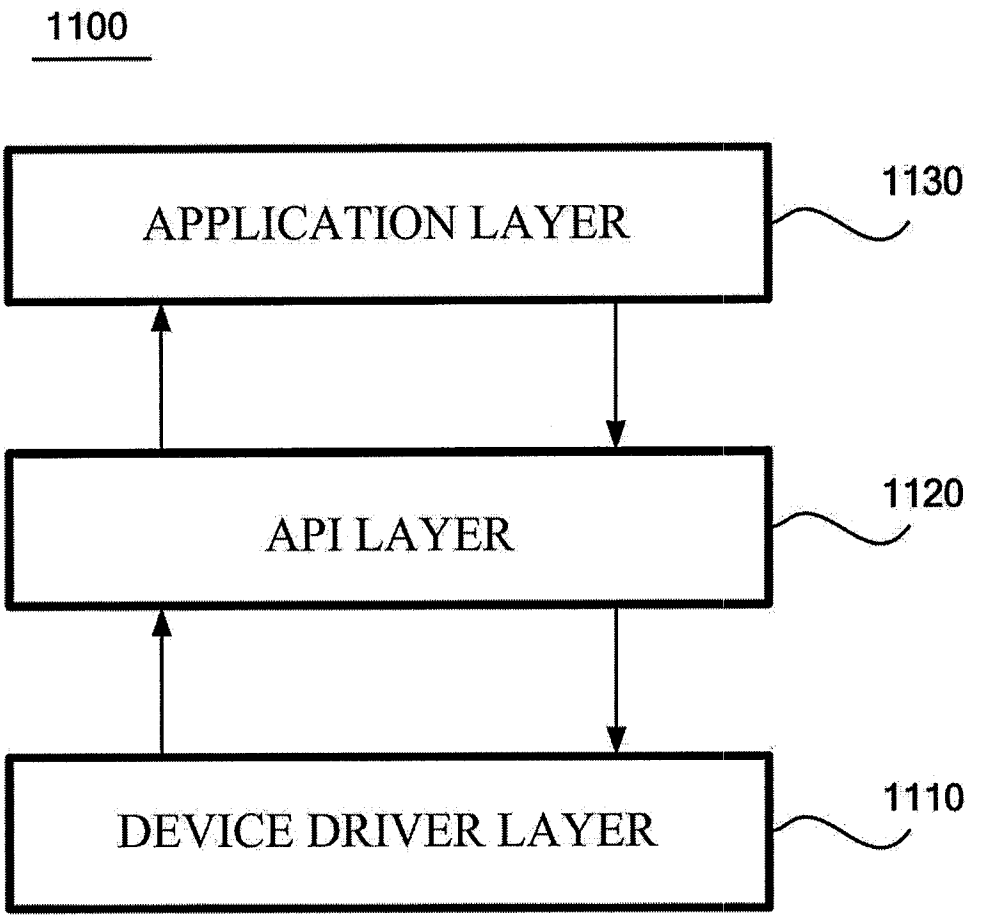

1100
APPLICATION LAYER
1130
API LAYER
1120
DEVICE DRIVER LAYER
1110

FIG. 19

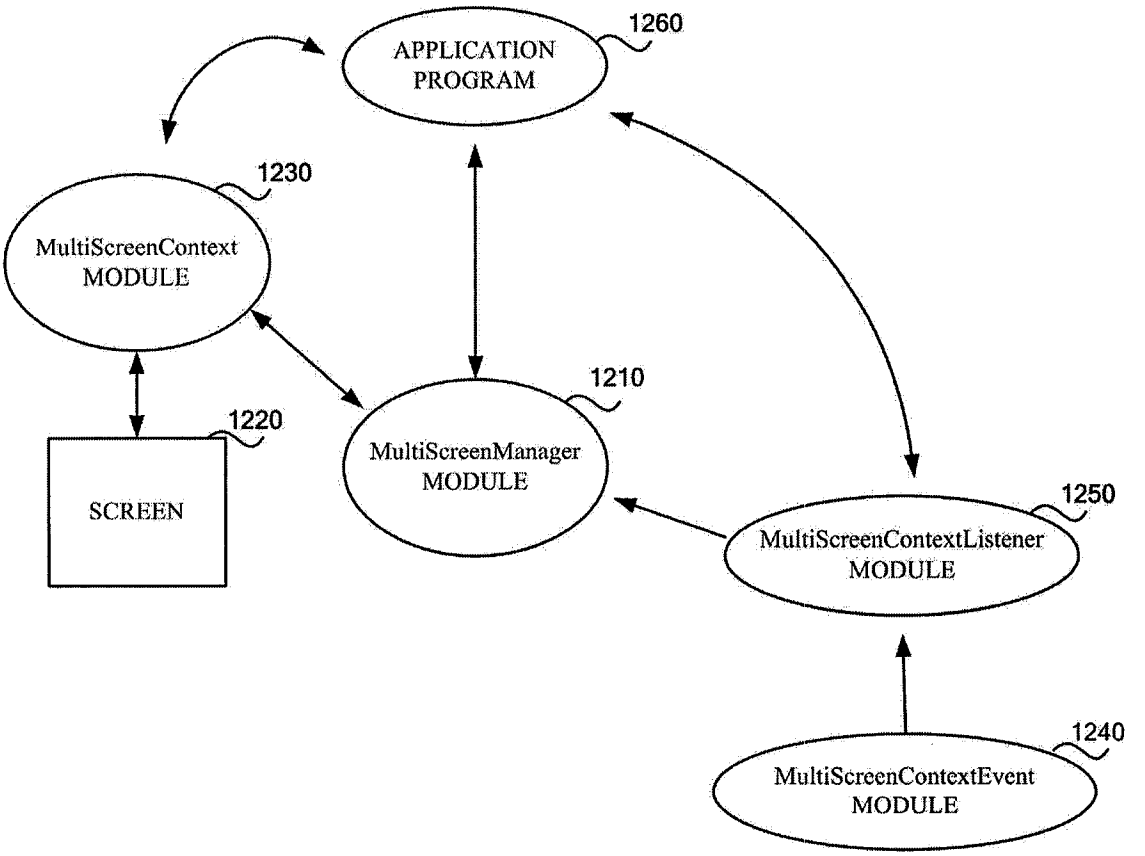

APPLICATION PROGRAM
1260
MultiScreenContext MODULE
1230
SCREEN
1220
MultiScreenManager MODULE
1210
MultiScreenContextListener MODULE
1250
MultiScreenContextEvent MODULE
1240

FIG. 20

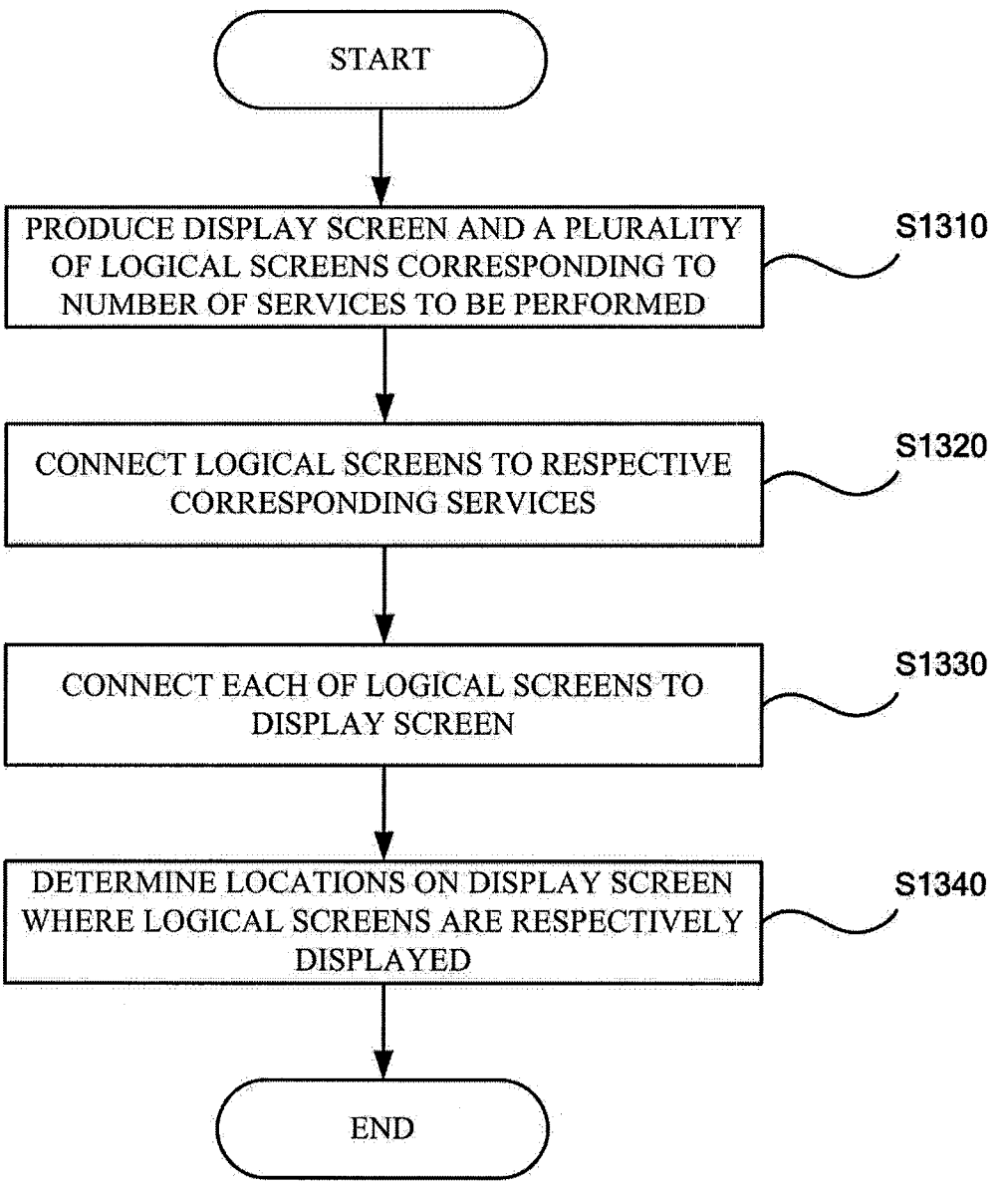

START
PRODUCE DISPLAY SCREEN AND A PLURALITY OF LOGICAL SCREENS CORRESPONDING TO NUMBER OF SERVICES TO BE PERFORMED
S1310
CONNECT LOGICAL SCREENS TO RESPECTIVE CORRESPONDING SERVICES
S1320
CONNECT EACH OF LOGICAL SCREENS TO DISPLAY SCREEN
S1330
DETERMINE LOCATIONS ON DISPLAY SCREEN WHERE LOGICAL SCREENS ARE RESPECTIVELY DISPLAYED
S1340
END

FIG. 21

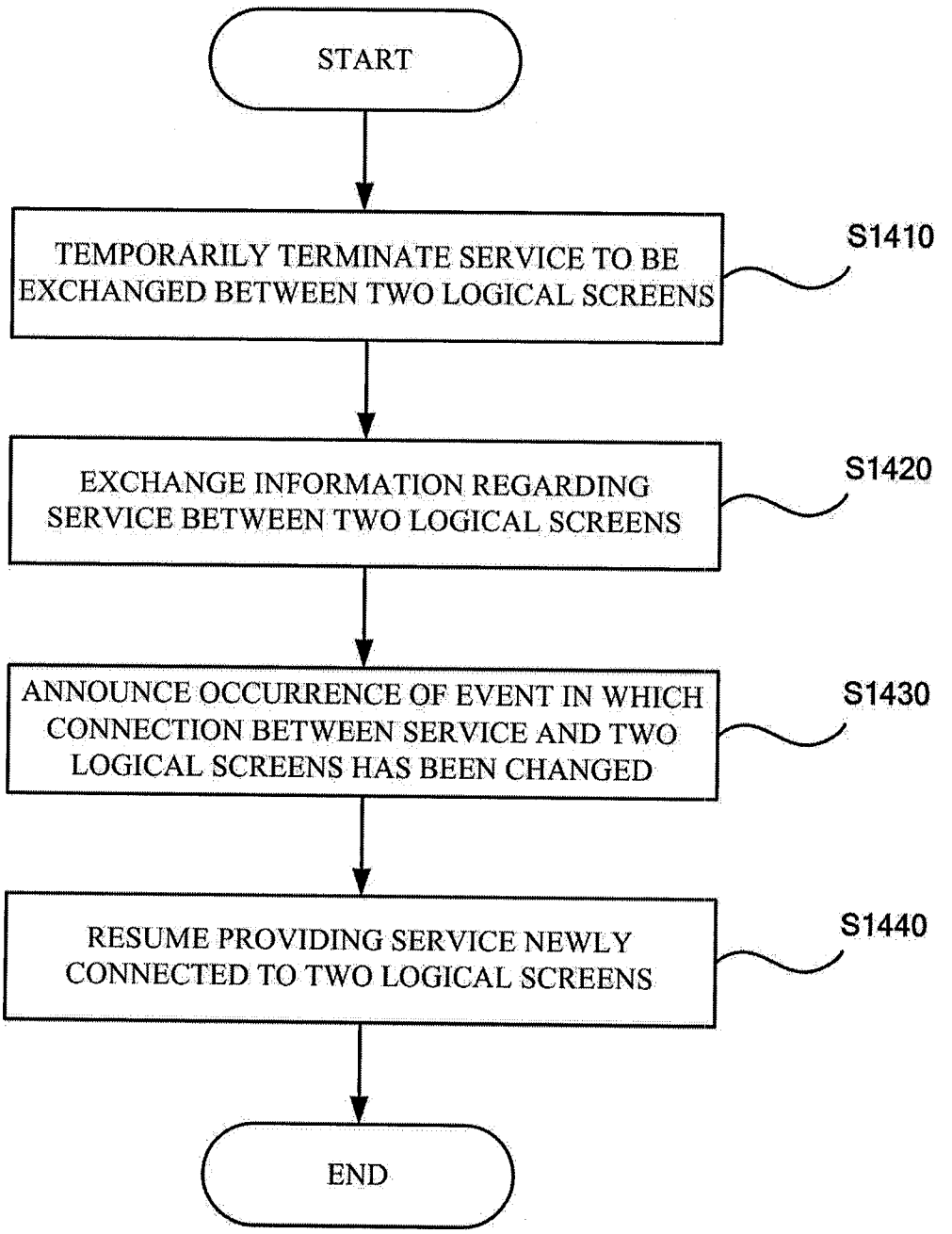

START
TEMPORARILY TERMINATE SERVICE TO BE EXCHANGED BETWEEN TWO LOGICAL SCREENS
S1410
EXCHANGE INFORMATION REGARDING SERVICE BETWEEN TWO LOGICAL SCREENS
S1420
ANNOUNCE OCCURRENCE OF EVENT IN WHICH CONNECTION BETWEEN SERVICE AND TWO LOGICAL SCREENS HAS BEEN CHANGED
S1430
RESUME PROVIDING SERVICE NEWLY CONNECTED TO TWO LOGICAL SCREENS
S1440
END

FIG. 22

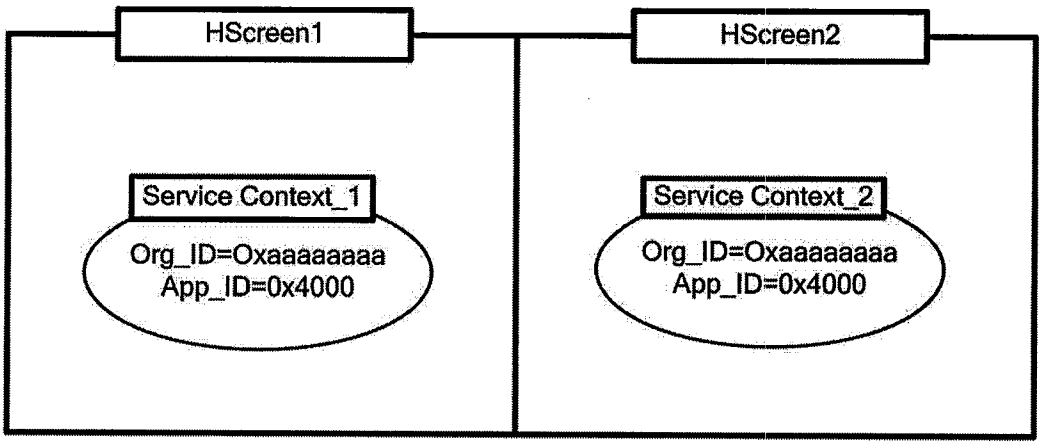

HScreen1
HScreen2
Service Context_1
Org_ID=Oxaaaaaaaa
App_ID=0x4000
Service Context_2
Org_ID=Oxaaaaaaaa
App_ID=0x4000

FIG. 23

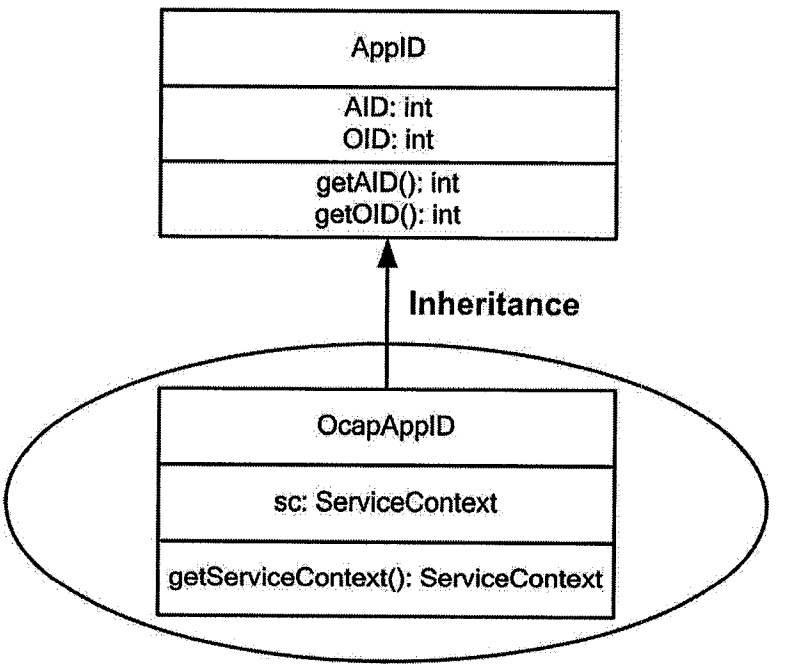
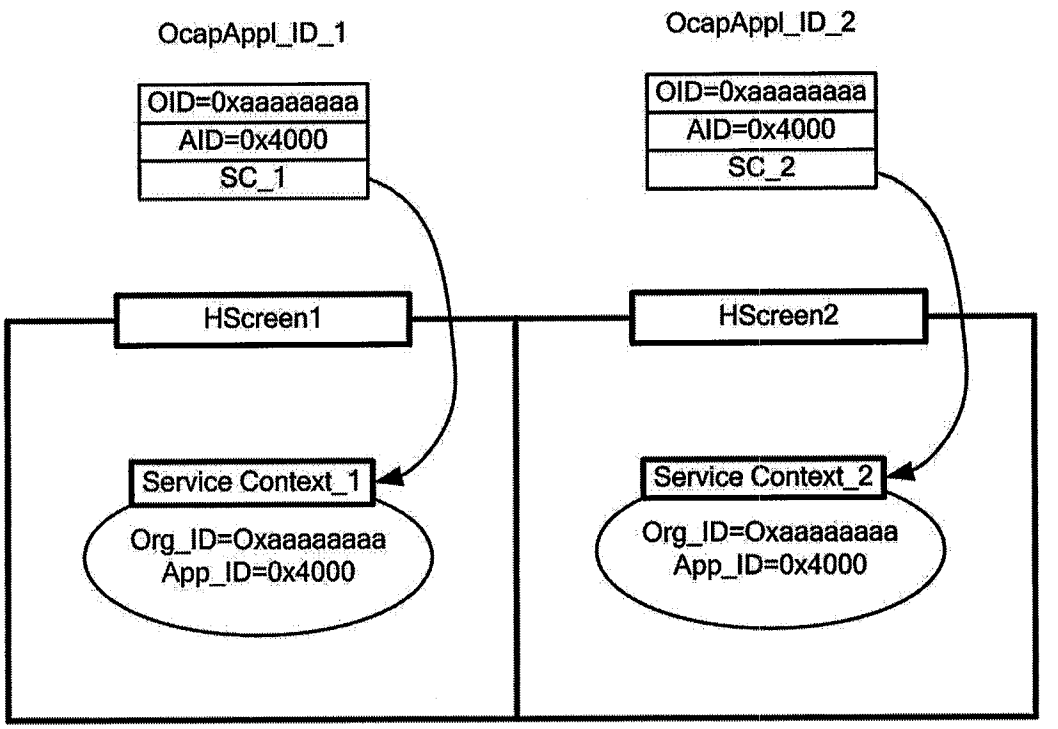

AppID
AID: int
OID: int
getAID(): int
getOID(): int
Inheritance
OcapAppID
sc: ServiceContext
getServiceContext(): ServiceContext

FIG. 24


OcapAppl_ID_1
OID=0xaaaaaaaa
AID=0x4000
SC_1
OcapAppl_ID_2
OID=0xaaaaaaaa
AID=0x4000
SC_2
HScreen1
HScreen2
Service Context_1
Org_ID=0xaaaaaaaa
App_ID=0x4000
Service Context_2
Org_ID=0xaaaaaaaa
App_ID=0x4000

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2007/001675

A. CLASSIFICATION OF SUBJECT MATTER

H04N 5/45(2006.01)i, H04N 5/00(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 8: H04N 5/44, H04N 5/445, H04N5/45

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean Utility models and applications for Utility Models since 1975
Japanese Utility Models and application for Utility Models since 1975

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKIPASS (KIPO internal) "multiple screen", "display screen", "logical screen"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2004-047433 A1 (PHILIPS ELECTRONICS N.V.) 03 Jun. 2004 See Abstract; Fig. 1;	1-12
A	WO 03-077540 A1 (PHILIPS ELECTRONICS N.V.) 18 Sep. 2003 See Abstract; Fig. 2;	1-12
A	WO 01-04868 A1 (SUN MICRO SYSTEMS, INC.) 18 Jan. 2001 See Abstract; Fig. 3;	1-12

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 JUNE 2007 (25.06.2007)	**25 JUNE 2007 (25.06.2007)**

Name and mailing address of the ISA/KR
Korean Intellectual Property Office
920 Dunsan-dong, Seo-gu, Daejeon 302-701,
Republic of Korea
Facsimile No. 82-42-472-7140

Authorized officer
NAM, Ock Woo
Telephone No. 82-42-481-5713

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/KR2007/001675

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO2004047433A1	03.06.2004	AU 2003272026A1	15.06.2004
		CN 1711759A	21.12.2005
		EP 1566055A1	24.08.2005
		JP 18506876	23.02.2006
		KR 1020050086696A	30.08.2005
		US 20060010476A1	12.01.2006
WO03077540A1	18.09.2003	AU 2003252809A1	22.09.2003
		CN 1643902A	20.07.2005
		EP 1486059A1	15.12.2004
		JP 17520407	07.07.2005
		KR 1020040093126A	04.11.2004
		US 20050120379A1	02.06.2005
WO0104868A1	18.01.2001	AT 289091E	15.02.2005
		AU 200059345A1	30.01.2001
		AU 200060973A1	30.01.2001
		AU 200060992A1	30.01.2001
		AU 200063495A1	30.01.2001
		AU 776813B2	23.09.2004
		AU 781357B2	19.05.2005
		CA 2373036A1	18.01.2001
		CA 2378588A1	18.01.2001
		CA 2378844A1	18.01.2001
		CN 1156753C	07.07.2004
		CN 1227588C	16.11.2005
		CN 1264354C	12.07.2006
		CN 1360694A	24.07.2002
		CN 1360695A	24.07.2002
		CN 1373967A	09.10.2002
		EP 01192617A1	03.04.2002
		EP 01194838A2	10.04.2002
		EP 01194840A2	10.04.2002
		EP 01197090A2	17.04.2002
		JP 2003504753T2	04.02.2003
		JP 2003504754T2	04.02.2003
		JP 2003504923T2	04.02.2003
		JP 2003522442T2	22.07.2003
		KR 1020020030784A	25.04.2002
		KR 1020020022085A	23.03.2002
		KR 1020020023414A	28.03.2002
		KR 1020020035558A	11.05.2002
		US 6701334B1	02.03.2004
		US 6762798B1	13.07.2004
		US 6874145B1	29.03.2005
		US 6986155BA	10.01.2006
		WO 200104743A2	18.01.2001
		WO 200104744A2	18.01.2001
		WO 200104868A1	18.01.2001
		WO 200105158A1	18.01.2001