

Application Interface Management Method and Apparatus

Technical Field

The present application relates to the technical field of terminals, particularly to an application interface management method and apparatus.

5 Background

An APP (Application) is a kind of software that can be installed in smart terminal devices, such as mobile phones and tablet computers. Users may implement various services through related APPS, such as: payment, shopping, and financial management. In related art, the layout of an APP interface typically is set by a developer in advance. However, as
10 different users have different usage requirements for an APP, an application interface set in advance usually is unable to meet the usage requirements of all users.

Summary

In light of the above, the present application provides an application interface management method and apparatus. The present application is achieved through the
15 following technical solutions.

An application interface management method, comprising:

requesting interface layout parameters from an interface layout server in an application start process, the interface layout parameters comprising layout parameters of one or more functional areas and layout parameters of one or more functional icons;

20 generating an application interface according to obtained layout parameters of functional areas and obtained layout parameters of functional icons, each functional area comprising one or more functional icons;

in response to a change operation by a user on the application interface, changing an interface layout of the application interface; and

25 storing layout parameters of one or more functional areas and layout parameters of one or more functional icons corresponding to the changed application interface, to the interface layout server.

An application interface management apparatus, comprising:

a parameter requesting unit, configured to request interface layout parameters from an
30 interface layout server in an application start process, the interface layout parameters comprising layout parameters of one or more functional areas and layout parameters of one or more functional icons;

an interface generating unit, configured to generate an application interface according

to obtained layout parameters of functional areas and obtained layout parameters of functional icons, each functional area comprising one or more functional icons;

an interface changing unit, in response to a change operation by a user on the application interface, configured to change an interface layout of the application interface;

5 and

a parameter storing unit, configured to store layout parameters of one or more functional areas and layout parameters of one or more functional icons corresponding to the changed application interface, to the interface layout server.

As shown in the above description, the present specification sets an interface management solution based on actual user requirements, and the user can customize the layout of an entire application interface according to their own requirements, thereby improving user experience and user stickiness.

Brief Description of the Drawings

FIG. 1 is a schematic diagram of an application interface shown in an exemplary embodiment of the present application.

FIG. 2 is a flow chart of an application interface management method shown in an exemplary embodiment of the present application.

FIG. 3 is a schematic diagram of an alternative application interface shown in an exemplary embodiment of the present application.

FIG. 4 is a schematic diagram of an alternative application interface shown in an exemplary embodiment of the present application.

FIG. 5 is a schematic diagram of an alternative application interface shown in an exemplary embodiment of the present application.

FIG. 6 is a schematic diagram of an alternative application interface shown in an exemplary embodiment of the present application.

FIG. 7 is a schematic diagram of an alternative application interface shown in an exemplary embodiment of the present application.

FIG. 8 is a structure diagram of a management apparatus used in an application interface shown in an exemplary embodiment of the present application.

FIG. 9 is a block diagram of an application interface management apparatus shown in an exemplary embodiment of the present application.

Detailed Description

Here, exemplary embodiments will be described in detail, with examples shown in the accompanying drawings. When the description below involves the accompanying drawings,

unless otherwise indicated, the same numeral in different accompanying drawings stands for the same or similar element. The implementation manners described in the following exemplary embodiments do not represent all the implementation manners consistent with the present application. Conversely, they are only examples of the apparatus and method
5 described in detail in the attached claims and consistent with some aspects of the present application.

The terms used in the present application are only for the purpose of describing exemplary embodiments and not intended to limit the present application. The singular forms “one”, “the”, and “this” used in the present application and in the attached claims also are
10 intended to cover plural forms unless their meanings are otherwise clearly indicated in the context. It should also be understood that the term “and/or” used in the text refers to any or all possible combinations containing one or more of the associated listed items.

It should be understood that although the present application may use terms such as first, second, and third to describe various kinds of information, the information should not be
15 limited to these terms. These terms are only intended to differentiate information of the same type. For example, without departing from the scope of the present application, first information may also be referred to as second information, and similarly, second information may also be referred to as first information. Subject to the context, the term “if” used here may be interpreted as “at the time of ...”, “when ...”, or “in response to a determination.”

20 In related art, an application interface includes a plurality of page areas, and each page area may include one or more buttons in a form of icons. Please refer to the schematic diagram of an application interface shown in FIG. 1. The area where “Button A”, “Button B”, and “Button C” are located is a page area of the application interface. “Button A”, “Button B”, and “Button C” are icon buttons in this area, and by clicking on these icon buttons,
25 corresponding functions may be enabled, such as: displaying a payment code, displaying an existing coupon, and the like. The area where “Button D”, “Button E”, ..., “Button H” are located is another page area of the application interface. “Button D”, “Button E”, ..., “Button H” are icon buttons in this area, and by clicking on these icon buttons, corresponding functions may also be enabled, such as: enabling an account transfer function, enabling a
30 payment function, and the like.

In the present application, to facilitate description, these buttons that are used to enable corresponding functions/operations may be called functional icons, and the page areas may be called functional areas. The foregoing functional areas and functional icons are set by developers. In general, the developers often set functional icons of the same function type in

the same functional area according to service providers requirements. For example, "Button A", "Button B" and "Button C" shown in FIG. 1 are located in functional area 1, and "Button D", "Button E", ..., "Button H" are located in functional area 2. At present, some applications allow users to customize the functional icons in a certain functional area. For example, a user
5 may set an order of "Button D", "Button E", ..., "Button H" in functional area 2 according to his own usage preferences. However, this only allows a personalized setting in a certain functional area, settings in other areas still mainly adopt functional icons or functional areas closely related to application profit points, and thus, the actual requirements of all users are not met.

10 FIG. 2 is a flow chart of an application interface management method shown in an exemplary embodiment of the present application.

Referring to FIG. 2, the application interface management method may be used in smart terminal devices, such as mobile phones and tablet computers. The method may include the following steps.

15 In Step 201, interface layout parameters may be requested from an interface layout server in an application start process. The interface layout parameters may include layout parameters of one or more functional areas and layout parameters of one or more functional icons.

In this embodiment, after a user starts an application, the application may request the
20 interface layout parameters of this application from an interface layout server. The interface layout server may be a server or a server cluster deployed by a service provider, and it may be used to store interface layout parameters of an interface personalized by a user. Of course, the interface layout server may further be used to provide related business services for users. The present application places no special limitations on this.

25 In this embodiment, the interface layout parameters may include: layout parameters of one or more functional areas in each interface of this application, and layout parameters of one or more functional icons. The layout parameters of the functional areas may include: IDs of functional areas and location parameters of the functional areas in the interface. The layout parameters of the functional icons include: IDs of functional icons, IDs of functional areas
30 where the functional icons are located, and location parameters of the functional icons in the functional areas where the functional icons are located.

Still referring to the application interface shown in FIG. 1 as an example, the interface layout parameters obtained by the application may include: the ID of functional area 1 and the location parameter of functional area 1 in this interface, the ID of functional area 2 and

the location parameter of functional area 2 in this interface, the IDs of functional icon A to functional icon H, the functional area IDs in which functional icon A to functional icon H are located, and the location parameters in the functional areas in which each functional icon is located. For example, the ID of functional area 1 is ID1, and the location parameter of functional area 1 is 1, indicating that the location of functional area 1 is the uppermost part of this interface; the ID of functional area 2 is ID2, and the location parameter of functional area 2 is 2, indicating that the location of functional area 2 is below functional area 1 and is adjacent to functional area 1.

In Step 202, an application interface is generated according to the obtained layout parameters of functional areas and the obtained layout parameters of functional icons. Each functional area may include one or more functional icons.

Based on Step 201, after the interface layout parameters are obtained, an application interface may be generated according to the interface layout parameters.

In this embodiment, a corresponding functional area may be displayed in an application interface according to the ID of the functional area and the location parameter of the functional area in the interface. Corresponding functional icons may be obtained from a locally prestored functional icon database according to the IDs of the functional icons, and they may be displayed in the corresponding functional areas according to the IDs of the functional areas where the functional icons are located and the location parameters of the functional icons in the functional areas where the functional icons are located. The functional icon database is obtained from an interface layout server and stored locally when a user installs or upgrades this application. Subsequently, when displaying the interface, the corresponding application icons may be directly obtained from the functional icon database according to the IDs of the functional icons, without making a request to the server again, thereby speeding up the interface display and saving user data traffic.

Still referring to the application interface shown in FIG. 1 as an example, in this step, functional area 1 may be displayed in the uppermost part of the application interface according to functional area ID1, and functional area 2 may be displayed adjacent to and under functional area 1 in the application interface according to functional area ID2.

Assuming the background color of functional area 1 is light blue and the background color of functional area 2 is white, a light blue functional area 1 and a white functional area 2 may be displayed. Further, functional icon A to functional icon H may be obtained from a locally prestored functional icon database according to the IDs of functional icon A to functional icon H, and then functional icon A is displayed in light blue functional area 1 according to

functional area ID1 in which functional icon A is located and the location parameter of functional icon A in functional area 1. Functional icon B to functional icon H may be displayed in the same way.

The foregoing interface display process is not visible to users. After a user starts an application, the user may see the displayed application interface, whereas the foregoing steps 201 and 202 describe the process of the application arranging the interface in backend. Related art may be referenced for the processing and implementation of this part, and it will not be described in detail in the present specification.

In Step 203, in response to a change operation by a user on the current interface, the interface layout of the current interface may be changed.

In this embodiment, a user may customize the layout of an application interface. For example, the user may adjust the locations of every functional icon and every functional area and may also delete functional icons and functional areas the user does not use. The user may trigger a change in an interface layout through a specific operation. The user may also press and hold or double click on any functional icon to trigger a change in an interface layout.

In this embodiment, when a user triggers a change in an interface layout, a close button of a shape “×” may be displayed in the upper right corner of each functional area and each functional icon. When the user clicks on the “×” button of a functional area or a functional icon, the user may delete the corresponding functional area or functional icon in the current application interface. When a user deletes a functional area, all functional icons in the functional area may be deleted all together, and the displayed layout of the application interface is changed by moving adjacent functional areas closer together. Referring to FIG. 3, on the basis of the application interface shown in FIG. 1, the application interface shown in FIG. 3 may further include functional area 3. Assuming functional area 3 is a functional area for displaying advertisements and does not include any functional icons, when a user clicks on the “×” button in the upper right corner of functional area 3, functional area 3 may be deleted from the application interface, resulting in the application interface shown in FIG. 1. Further, assuming the user has deleted all functional icons in functional area 2, functional area 3 will be automatically adjusted to the location of the former functional area 2, forming the application interface shown in FIG. 4.

In this embodiment, a user may further adjust the locations of functional areas or functional icons. In the application interface shown in FIG. 3, after pressing and holding functional icon F, the user may drag functional icon F to the location where functional icon A is located to achieve a cross-area location adjustment of functional icons and form the

application interface shown in FIG. 5. In another example, the user may press and hold functional area 3 and drag it to the location of functional area 2 to switch the locations of functional area 3 and functional area 2 and form the application interface shown in FIG. 6.

5 In one example, the dimensions of the functional areas may be fixed. As shown in FIG. 1 and FIG. 5, the dimensions of functional area 1 are fixed. When a user moves functional icon H to the location of functional icon A, the locations of functional icon A and functional icon B are moved accordingly to make room for functional icon H. As the dimensions of functional area 1 are not changeable, and consequently, functional icon C may be deleted.

10 In another example, the dimensions of the functional areas may be scalable. Based on the application interface shown in FIG. 3, assuming the user has deleted functional icon F to functional icon H in functional area 2, the dimensions of functional area 2 will decrease accordingly, and functional area 3 will be moved upwards accordingly, forming the application interface shown in FIG. 7.

15 In Step 204, layout parameters of one or more functional areas and layout parameters of one or more functional icons corresponding to the changed application interface may be stored, to the interface layout server.

Based on the foregoing step 203, when a user exits from an interface change, the application may store to the interface layout server the layout parameters of one or more functional areas and the layout parameters of one or more functional icons corresponding to the current application interface. Here, the interface layout server may store interface layout parameters of the user according to a user account number. The present specification places no special limitations on this.

25 As can be seen from the above description, the present specification sets an interface management solution based on actual user requirements, and users can customize the layout of an entire application interface according to their own requirements, thereby improving user experience and user stickiness.

Optionally, in another example, behavioral analysis of a user may be conducted based on the interface layout parameters, so as to perform targeted information push to the user. For instance, assuming user Xiao Bai arranges a fund icon in the most prominent and convenient location in an application interface, it may imply that Xiao Bai buys and sells funds at a high frequency, so that fund-related information and advertisements may be pushed to Xiao Bai.

Corresponding to the foregoing application interface management method, the present application further provides embodiments of an application interface management apparatus.

The embodiments of the application interface management apparatus provided by the

present application may be used on a terminal device. The apparatus may be implemented through software, hardware, or a combination of hardware and software. Taking implementation through software as an example, an apparatus in a logical sense is formed through the processor of the terminal device, which reads the corresponding computer
5 program commands to memory and runs them. In terms of hardware, FIG. 8 is a hardware structure diagram of the terminal device where the application interface management apparatus provided by the present application is located. In addition to a processor, memory, network interface, and NVM (nonvolatile memory) as shown in FIG. 8, the terminal device where the apparatus of the embodiment is located may further include other hardware
10 according to the actual functions of the terminal device. They are not described in detail here.

Referring to FIG. 9, the management apparatus 800 of the application interface may be used in the terminal device shown in FIG. 8, and include: a parameter requesting unit 801, an interface generating unit 802, an interface changing unit 803, and a parameter storing unit 804.

15 Here, the parameter requesting unit is configured to request interface layout parameters from an interface layout server in an application start process. The interface layout parameters may include layout parameters of one or more functional areas and layout parameters of one or more functional icons.

The interface generating unit 802 is configured to generate an application interface
20 according to obtained layout parameters of functional areas and obtained layout parameters of functional icons. Each functional area may include one or more functional icons.

The interface changing unit 803 is configured to change the interface layout of the current interface in response to a change operation by a user on the current application interface.

25 The parameter storing unit 804 is configured to store layout parameters of one or more functional areas and layout parameters of one or more functional icons corresponding to the changed application interface, to the interface layout server.

Optionally, layout parameters of the functional areas may include: IDs of functional areas and location parameters of functional areas in the interface.

30 Layout parameters of the functional icons may include: IDs of functional icons, IDs of the functional areas where the functional icons are located, and location parameters of the functional icons in the functional areas where the functional icons are located.

The interface generating unit 802 may display corresponding functional areas in an application interface according to the IDs of the functional areas and the location parameters

of the functional areas in the interface, obtain corresponding functional icons from a locally prestored functional icon database according to IDs of functional icons, and display the functional icons in a corresponding functional areas according to the IDs of the functional areas where the functional icons are located and the location parameters of the functional icons in the functional areas where the functional icons are located.

Optionally, the change operation may include: a functional icon deletion operation.

When all functional icons in a functional area are deleted, the interface changing unit 803 will delete the functional area and change the display layout of the application interface by moving adjacent functional areas closer together.

Optionally, dimensions of the functional areas are fixed or scalable.

Optionally, the interface layout parameters can be a basis of user behavior analysis and information push.

The implementation processes of the functions and effects of every unit in the foregoing apparatus are detailed in the implementation processes of corresponding steps in the foregoing method, and will not be repeated herein.

The apparatus embodiments essentially correspond to the method embodiments, so for relevant parts of the apparatus embodiments, please refer to the corresponding parts of the method embodiments. The apparatus embodiments described above are exemplary only, their units described as separate components may or may not be physically separate, and the components displayed as units may or may not be physical units, i.e., they may be located in the same place or distributed on a plurality of network units. Some or all of the modules may be selected according to the actual need to achieve the objectives of the solution of the present application. Those of ordinary skill in the art can understand and implement it without creative effort.

The foregoing descriptions are preferred embodiments of the present application and are not intended to limit the present application. All modifications, identical replacements, and improvements made without departing from the spirit and principle of the present application shall fall within the scope of the present application.

CLAIMS

1. An application interface management method, wherein the method comprises:
 - requesting interface layout parameters from an interface layout server in an application start process, the interface layout parameters comprising layout parameters of one or more functional areas and layout parameters of one or more functional icons;
 - generating an application interface according to the obtained layout parameters of functional areas and obtained layout parameters of functional icons, each functional area comprising one or more functional icons;
 - in response to a change operation by a user on the application interface, changing an interface layout of the application interface, wherein the changing an interface layout of the application interface comprises:
 - in response to the user deleting a target function icon in a first functional area, changing one or more location parameters of the first functional area to decrease dimensions of the first functional area by scaling the first functional area, and
 - following the scaling of the first functional area, changing one or more location parameters of a second functional area to move the second functional area such that the second functional area occupies at least part of a region in the application interface that was occupied by the first functional area prior to the scaling and not occupied by the first functional area subsequent to the scaling; and
 - storing layout parameters of one or more functional areas and layout parameters of one or more functional icons corresponding to the changed application interface, to the interface layout server.
2. The method of claim 1, wherein
 - the layout parameters of functional areas comprise: IDs of the functional areas and location parameters of the functional areas in the application interface;
 - the layout parameters of functional icons comprise: IDs of the functional icons, IDs of functional areas where the functional icons are located, and location parameters of the functional icons in the functional areas where the functional icons are located; and
 - generating an application interface according to the obtained layout parameters of functional areas and obtained layout parameters of functional icons, comprises:
 - displaying corresponding functional areas in the application interface according to the IDs of the functional areas and the location parameters of the functional areas in the interface;
 - obtaining corresponding functional icons from a locally prestored functional icon

database according to the IDs of the functional icons; and

displaying the functional icons in the corresponding functional areas according to the IDs of the functional areas where the functional icons are located and the location parameters of the functional icons in the functional areas where the functional icons are located.

3. The method of claim 2, wherein the change operation comprises: a functional icon deletion operation; and the method further comprises:

when all functional icons in a functional area are deleted, deleting the functional area and changing the interface layout of the application interface by moving adjacent functional areas closer together.

4. The method of claim 1, wherein dimensions of the functional areas are fixed or scalable.

5. The method of claim 1, wherein the interface layout parameters are a basis of user behavior analysis and information push.

6. An application interface management apparatus, wherein the apparatus comprises:

a parameter requesting unit, configured to request interface layout parameters from an interface layout server in an application start process, the interface layout parameters comprising layout parameters of one or more functional areas and layout parameters of one or more functional icons;

an interface generating unit, configured to generate an application interface according to obtained layout parameters of functional areas and obtained layout parameters of functional icons, each functional area comprising one or more functional icons;

an interface changing unit, in response to a change operation by a user on the application interface, configured to change an interface layout of the application interface comprising:

in response to the user deleting a target function icon in a first functional area, changing one or more location parameters of the first functional area to decrease dimensions of the first functional area by scaling the first functional area, and

following the scaling of the first functional area, changing one or more location parameters of a second functional area to move the second functional area such that the second functional area occupies at least part of a region in the application interface that was occupied by the first functional area prior to the scaling and not occupied by the first

functional area subsequent to the scaling; and

a parameter storing unit, configured to store layout parameters of one or more functional areas and layout parameters of one or more functional icons corresponding to the changed application interface, to the interface layout server.

5 7. The apparatus of claim 6, wherein

the layout parameters of functional areas comprise: IDs of the functional areas and location parameters of the functional areas in the interface,

the layout parameters of the functional icons comprise: IDs of the functional icons, IDs of functional areas where the functional icons are located, and location parameters of the functional icons in the functional areas where the functional icons are located; and

10 the interface generating unit is configured to display corresponding functional areas in the application interface according to the IDs of the functional areas and the location parameters of the functional areas in the interface, obtain corresponding functional icons from a locally prestored functional icon database according to the IDs of the functional icons, and display the functional icons in the corresponding functional areas according to the IDs of the functional areas where the functional icons are located and the location parameters of the functional icons in the functional areas where the functional icons are located.

8. The apparatus of claim 7, wherein

the change operation comprises: a functional icon deletion operation; and

20 when all functional icons in a functional area are deleted, the interface changing unit is configured to delete the functional area and change the interface layout of the application interface by moving adjacent functional areas closer together.

9. The apparatus of claim 6, wherein dimensions of the functional areas are fixed or scalable.

25 10. The apparatus of claim 6, wherein the interface layout parameters is a basis of user behavior analysis and information push.