

SERVICE PROCESSING METHOD AND DEVICE

TECHNICAL FIELD

[0001] The present application relates to the field of Internet technologies, and in particular, to a service processing method and device.

5

BACKGROUND

[0002] With rapid development of Internet technologies and electronic device technologies, there are more and more types of electronic devices such as VR devices, personal computers, tablet computers, or smartphones. Different types of electronic devices can run different operating systems, and use different programming languages to write interaction instructions. Therefore, a service provider that provides certain services usually needs to develop different application programs for the different types of electronic devices.

[0003] For example, by using a service provider that provides a K-line share graph as an example, a smartphone, a personal computer, and a VR device can all access a website provided by the service provider or install an application program provided by the service provider, to view the K-line share graph provided by the service provider. In a process of viewing the K-line share graph, the device can further initiate an interaction instruction such as shifting to the left, shifting to the right, pinching and stretching, or sliding up, and the service provider performs corresponding processing on the K-line share graph based on the interaction instruction, and provides processed data for the device for rendering display. For interaction instructions initiated by the smartphone, the personal computer, and the virtual reality (VR) device, the service provider needs to develop different service processing modules for different devices, to detect and identify interaction instructions of the different devices. It can be learned that in the service processing mode in the related technology, the service provider needs to invest a relatively large quantity of development resources and costs.

SUMMARY

[0004] To alleviate a problem in a related technology, the present application provides a service processing method and device.

[0005] According to a first aspect of implementations of the present application, a service processing method is provided, where an instruction processing rule corresponding to each device type is configured according to the method; and the method includes the following: detecting a device type of an electronic device that needs to display interaction data; obtaining an instruction processing rule corresponding to the detected device type; obtaining, based on the instruction processing rule, a device interaction instruction initiated by the electronic device, and converting the device interaction instruction into a corresponding unified interaction instruction; and processing the interaction data by invoking a service processing mode corresponding to the unified interaction instruction obtained through conversion, and providing processed interaction data for the electronic device for display.

[0006] In an optional implementation, the detecting a device type of an electronic device that needs to display interaction data includes the following: initiating a device type query request to the electronic device; and receiving device type description information fed back by the electronic device based on the query request, and determining the device type based on the device type description information.

[0007] In an optional implementation, the instruction processing rule includes an instruction acquisition rule used to obtain the device interaction instruction initiated by the electronic device, and an instruction set conversion rule used to convert the device interaction instruction into the corresponding unified interaction instruction.

[0008] In an optional implementation, the instruction processing rule is preconfigured in the following way: receiving a correspondence among a device type, a device interaction instruction, and a unified interaction instruction that are sent by a development side device, and configuring the instruction processing rule based on the correspondence.

[0009] In an optional implementation, a coding scheme for the unified interaction instruction is a coding scheme of one of different device types for a device interaction instruction.

[0010] According to a second aspect of implementations of the present application, a service processing device is provided, where an instruction processing rule

corresponding to each device type is configured according to the device; and the device includes the following: a type detection module, configured to detect a device type of an electronic device that needs to display interaction data; a rule acquisition module, configured to obtain an instruction processing rule corresponding to the
 5 detected device type; an instruction conversion module, configured to obtain, based on the instruction processing rule, a device interaction instruction initiated by the electronic device, and convert the device interaction instruction into a corresponding unified interaction instruction; and a service processing module, configured to process the interaction data by invoking a service processing mode corresponding to the
 10 unified interaction instruction obtained through conversion, and provide processed interaction data for the electronic device for display.

[0011] In an optional implementation, the type detection module is further configured to: initiate a device type query request to the electronic device; and receive device type description information fed back by the electronic device based on the
 15 query request, and determine the device type based on the device type description information.

[0012] In an optional implementation, the instruction processing rule includes an instruction acquisition rule used to obtain the device interaction instruction initiated by the electronic device, and an instruction set conversion rule used to convert the
 20 device interaction instruction into the corresponding unified interaction instruction.

[0013] In an optional implementation, the instruction processing rule is preconfigured in the following way: receiving a correspondence among a device type, a device interaction instruction, and a unified interaction instruction that are sent by a development side device, and configuring the instruction processing rule based on the
 25 correspondence.

[0014] In an optional implementation, a coding scheme for the unified interaction instruction is a coding scheme of one of different device types for a device interaction instruction.

[0015] According to a third aspect of implementations of the present application,
 30 a service processing device is provided, and includes a processor and a memory configured to store a processor executable instruction, where the processor is configured to: detect a device type of an electronic device that needs to display interaction data; obtain an instruction processing rule corresponding to the detected device type; obtain, based on the instruction processing rule, a device interaction

instruction initiated by the electronic device, and convert the device interaction instruction into a corresponding unified interaction instruction; and process the interaction data by invoking a service processing mode corresponding to the unified interaction instruction obtained through conversion, and provide processed interaction data for the electronic device for display.

[0016] The technical solutions provided in the implementations of the present application can include the following beneficial effects:

[0017] In the implementations of the present application, the instruction processing rule corresponding to each device type is configured. Therefore, device interaction instructions of different types of electronic devices can be converted into the unified interaction instruction. Because the service processing mode corresponding to the unified interaction instruction is preconfigured, the interaction data can be processed, and the processed interaction data can be provided for the electronic device for display. In the implementations of the present application, a same interaction effect can be achieved based on different end-user devices and support capabilities, and developers do need to develop different interaction modules for different devices. Therefore, development resources and costs can be significantly reduced, and development efficiency can be improved.

[0018] It should be understood that the previously described general descriptions and the following detailed descriptions are merely examples for explanation, and do not limit the present application.

BRIEF DESCRIPTION OF DRAWINGS

[0019] The accompanying drawings here are incorporated in the specification and become a part of the specification, show implementations that are in accordance with the present application, and are used together with the specification to explain a principle of the present application.

[0020] FIG. 1A is a schematic diagram illustrating a service processing scenario, according to an example implementation of the present application;

[0021] FIG. 1B is a flowchart illustrating a service processing method, according to an example implementation of the present application;

[0022] FIG. 1C is a schematic diagram illustrating an interaction page displayed by an electronic device, according to an example implementation of the present

application;

[0023] FIG. 1D is a schematic diagram illustrating processing of a device interaction instruction and a unified interaction instruction, according to an example implementation of the present application;

5 [0024] FIG. 2A is a diagram illustrating an application scenario of a service processing method, according to an example implementation;

[0025] FIG. 2B is a schematic architectural diagram according to an example implementation of the present application;

10 [0026] FIG. 2C is a schematic diagram illustrating service processing of three electronic devices, according to an example implementation;

[0027] FIG. 2D is a flowchart illustrating another service processing method, according to an example implementation;

[0028] FIG. 3 is a diagram illustrating a hardware structure of an electronic device that a service processing device is located in, according to an example implementation
15 of the present application; and

[0029] FIG. 4 is a block diagram illustrating a service processing device, according to an example implementation of the present application.

DESCRIPTION OF IMPLEMENTATIONS

[0030] Example implementations are described in detail here, and examples of the
20 example implementations are presented in the accompanying drawings. When the following description relates to the accompanying drawings, unless specified otherwise, same numbers in different accompanying drawings represent a same or similar element. Implementations described in the following example implementations do not represent all implementations consistent with the present
25 application. On the contrary, the implementations are merely examples of devices and methods that are described in detail in the appended claims and that are consistent with some aspects of the present application.

[0031] The terms used in the present application are merely for the purpose of illustrating specific implementations, and are not intended to limit the present
30 application. The terms "a", "said" and "the" in singular forms used in the present application and the appended claims are also intended to include plural forms, unless otherwise specified in the context clearly. It should be further understood that the term

"and/or" used in the specification indicates and includes any or all possible combinations of one or more associated listed items.

[0032] It should be understood that although the terms such as "first", "second", "third", etc. can be used in the present application to describe various information, the information is not limited by the terms. These terms are merely used to distinguish between information of a same type. For example, without departing from the scope of the present application, first information can also be referred to as second information, and similarly the second information can also be referred to as the first information, which depends on the context. For example, the term "if" used here can be explained as "while", "when", or "in response to determining".

[0033] With rapid development of Internet technologies and electronic device technologies, there are more and more types of electronic devices such as VR devices, personal computers, tablet computers, or smartphones. Different types of electronic devices can run different operating systems, and use different programming languages to write interaction instructions. Therefore, a service provider that provides some services usually needs to develop different application programs for the different types of electronic devices.

[0034] In a related technology, many services relate to service processing. In an example, a service provider in a financial graph can provide a user with a service of viewing a K-line share graph. When an electronic device displays the K-line share graph, the user can need to perform an interaction operation such as sliding up, sliding down, or pinching and stretching on the displayed graph. The electronic device can obtain graph data obtained after the service processing based on the interaction operation performed by the user, and display a new graph. In another example, a service provider that provides a map service can provide a service of viewing a map. When an electronic device displays a map, a user can need to perform an interaction operation such as sliding up, sliding down, or pinching and stretching on the displayed map. The electronic device can obtain map data obtained after the service processing based on the interaction operation performed by the user, and display a new map.

[0035] A service provider usually needs to develop different application programs for different types of electronic devices, and therefore the service provider needs to invest a relatively large quantity of development resources and costs. The implementations of the present application provide a service processing solution, so as

to reduce development costs and development resources of the service provider, and improve development efficiency of an interaction system. The following describes the implementations of the present application in detail.

[0036] FIG. 1A is a schematic diagram illustrating a service processing scenario, according to an example implementation of the present application. A service provider and four electronic devices of different types are included in FIG. 1A, and a user can install an application program provided by the service provider on the electronic devices.

[0037] FIG. 1B is a flowchart illustrating a service processing method, according to an example implementation of the present application. An instruction processing rule corresponding to each device type is configured according to the method, and the method can include the following steps 101 to 104:

[0038] Step 101: Detect a device type of an electronic device that needs to display interaction data.

[0039] Step 102: Obtain an instruction processing rule corresponding to the detected device type.

[0040] Step 103: Obtain, based on the instruction processing rule, a device interaction instruction initiated by the electronic device, and convert the device interaction instruction into a corresponding unified interaction instruction.

[0041] Step 104: Process the interaction data by invoking a service processing mode corresponding to the unified interaction instruction obtained through conversion, and provide processed interaction data for the electronic device for display.

[0042] In this implementation, although the service processing process is for different devices, interaction operation objectives that a user needs to achieve for the different devices are consistent. For example, FIG. 1C is a schematic diagram illustrating an interaction page displayed by an electronic device. Because there is a relatively large amount of data on the interaction page, the electronic device displays only a part of the data, and data on the right of the diagram is not displayed on a device screen. Assume that the user needs to view the data on the right of the diagram, if the electronic device is a device that supports a touch function, the user can slide to the left on a touchscreen, and the electronic device detects an event of sliding to the left, and determines that the data on the right of the diagram needs to be displayed. If the electronic device is a personal computer, the user can drag to the left on a page by using a mouse, perform an operation on a sliding control displayed on the page by

using the mouse, shift to the left by using a direction key on a keyboard, etc. The personal computer detects one of the previously described events, and determines that the data on the right of the diagram needs to be displayed. If the electronic device is a VR device, the user can turn the head to the right. The VR device detects the event,
 5 and determines that the data on the right of the diagram needs to be displayed.

[0043] Based on the previously described analysis, it can be determined that although different electronic devices are used for a same piece of interaction data, interaction operation objectives that the user needs to achieve are consistent. Therefore, in this implementation, a unified interaction instruction is preconfigured
 10 for a same interaction data processing objective, an instruction processing rule for different device types is predetermined for the unified interaction instruction, and a corresponding service processing mode is configured.

[0044] For example, FIG. 1D is a schematic diagram illustrating processing of a device interaction instruction and a unified interaction instruction. FIG. 1D shows
 15 interaction instructions of sliding to the left that are initiated by four electronic devices. According to an instruction processing rule corresponding to each type, a corresponding instruction processing rule can be followed by a same unified interaction instruction, and a corresponding service processing mode is preconfigured for the unified interaction instruction. Therefore, the service processing mode can be
 20 invoked to process interaction data.

[0045] A device type is used to distinguish between devices that support different interaction trigger modes, and the device type can include an electronic device that supports a gravity sensing function, a VR device, a personal computer, an electronic device that supports a gesture recognition function, etc. In practice, the device type
 25 can be flexibly configured as needed. This is not limited in this implementation. An objective of distinguishing between device types is that because different types of electronic devices can run different operating systems and use different programming languages to code interaction instructions, corresponding instruction processing rules are preconfigured for the different device types, to invoke the corresponding
 30 instruction processing rules to process the interaction instructions initiated by the electronic devices.

[0046] After an application program is installed on the electronic device and started, a device type can be first detected. For example, it can be successively detected whether the device is a VR device, a device that supports a gravity sensing

function, a device that supports a gesture recognition function, a device that accesses a mouse, etc. In some examples, some application programs can have permission to directly read related device information of the electronic device, for example, a MAC address, a device name, or a device model, so as to determine a device type of the electronic device. In some other examples, an operating system of the electronic device has a device information read interface. Therefore, the detecting a device type of an electronic device that needs to display interaction data further includes the following: initiating a device type query request to the electronic device; and receiving device type description information fed back by the electronic device based on the query request, and determining the device type based on the device type description information.

[0047] In this implementation, the device type query request can be initiated to the electronic device by using an interface that the electronic device has. The electronic device sends the device type description information to an application program as requested by the application program. The application program receives the device type description information, and determines the device type based on the device type description information. The device type description information can be device type information, a MAC address, a device name, a device model, etc.

[0048] After the device type of the electronic device is determined, the instruction processing rule corresponding to the device type can be invoked, to obtain the device interaction instruction initiated by the electronic device, and convert the device interaction instruction into the unified interaction instruction. For how to obtain the device interaction instruction of the electronic device, in some examples, the operating system of the electronic device can capture an interaction operation initiated by the user. The operating system has a related interface of the interaction instruction, and the application program of the electronic device can obtain the device interaction instruction by using the interface. How to obtain the device interaction instruction by using the interface can be flexibly configured based on different operating system needs.

[0049] The unified interaction instruction is predetermined to one or more instructions and a combination of the instructions based on an actual interaction need. For example, the unified interaction instruction can include one or a combination of tapping, touching and holding, sliding to the left, sliding to the right, sliding up, sliding down, pinching and stretching, and rotating. A coding scheme for the unified

interaction instruction can be flexibly configured in practice, for example, the unified interaction instruction can be identified by using a digit, a character, or a character string. For example, a unified interaction instruction that represents "tapping" is represented as abc, a unified interaction instruction that represents "touching and holding" is represented as xyz, etc. This is not limited in this implementation.

[0050] To reduce development costs and improve development efficiency, in an optional implementation, a coding scheme for the unified interaction instruction is a coding scheme of one of different device types for a device interaction instruction. In this implementation, because corresponding coding schemes of all device types have been configured for device interaction instructions, one of the coding schemes can be selected as the coding scheme for the unified interaction instruction. For example, assume that there are four device types, coding schemes of the four device types for device interaction instructions can be obtained, and one of the coding schemes can be selected as the coding scheme for the unified interaction instruction, in other words, a device interaction instruction of one of the device types is used as the unified interaction instruction. Therefore, development costs can be reduced, and no instruction conversion needs to be performed on the device interaction instruction of the selected device type, so as to significantly improve service processing efficiency.

[0051] It can be learned from the previously described analysis that device interaction instructions of different operating systems can be obtained in different ways, and the instruction processing rule can be used to obtain the device interaction instruction, and convert the device interaction instruction into the unified interaction instruction. In some examples, device types can be different, but the electronic devices all run in a same operating system. For a same operating system, an interface for obtaining a device interaction instruction can be the same. However, because the device types are different, different trigger modes can be supported. For example, both a device that supports a gravity sensor and a device that supports a gesture event can run in an Android operating system although they are two different types of devices. In this case, the instruction processing rule includes an instruction acquisition rule used to obtain the device interaction instruction initiated by the electronic device, and an instruction set conversion rule used to convert the device interaction instruction into the corresponding unified interaction instruction.

[0052] In this implementation, developers can flexibly configure the instruction set conversion rule as needed. In an optional implementation, the instruction set

conversion rule is preconfigured in the following way: receiving a correspondence among a device type, a device interaction instruction, and a unified interaction instruction that are sent by a development side device, and configuring the instruction processing rule based on the correspondence.

5 **[0053]** In this implementation, the developers can be further provided with an instruction extension function, and the developers can extend the unified interaction instruction as needed. Specifically, the developers can configure the correspondence among a device type, a device interaction instruction, and a unified interaction instruction as needed, so as to preconfigure the instruction processing rule based on
10 the correspondence.

[0054] Because configuring the unified interaction instruction, the developers do not need to consider adaptation between different devices when performing service processing development. Therefore, development costs can be significantly reduced, and workload can be improved. In practice, a unified interaction instruction SDK can
15 be developed for steps 101 to 103 shown in FIG. 1B, and the SDK (which can be understood as an independent module) can be provided for the developers that perform service processing (corresponding to step 104). In addition, when an interaction component such as a webview component that can be embedded in a plurality of different devices is used, development efficiency can be more
20 significantly improved. For example, processing of the unified interaction instruction can be used as an independent SDK, and the SDK can be based on a webview (browser kernel) technology. Therefore, according to the previously described processing, the developers can develop only a unified module based on the unified interaction instruction in the service processing process, and the unified module can
25 be applied to service processing processes of different electronic devices. A webview component is embedded in all corresponding systems of a personal computer, an intelligent end-user device (such as Android or iOS), and a VR device. Therefore, development performed based on webview (including a native application that a webview kernel is embedded in or a web application that is directly based on a
30 browser) is almost applied to all end-user devices. Therefore, by introducing the SDK provided in this implementation based on webview by using a web development technology (javascript, html, css, etc.), the developers can perform deployment on various end-user devices and achieve a unified service processing effect by developing only one set of application programs. By using the solution in this

implementation, multi-terminal deployment through one set of code can be implemented, automatic adaptation supports a plurality of device interaction modes to achieve a same interaction effect, development costs can be reduced, and reusability is relatively high.

5 **[0055]** The following describes the service processing solution in the implementations of the present application in detail by using an implementation. In this implementation, a service provider can be a graph service provider, and can provide visual and interactive graph data for a user. An operation performed by the user on the graph data usually includes: sliding to the left and right, sliding up and
10 down, pinching and stretching, etc. In this implementation, unified graph interaction can be implemented for different interaction operations of an end-user device such as a personal computer, a smartphone, a tablet computer, or a VR device based on a web technology (such as webview or webVR API), so as to achieve an operation effect that the user needs to achieve.

15 **[0056]** FIG. 2A is a diagram illustrating an application scenario of a service processing method, according to an example implementation. Two groups of developers serving as service providers are included in FIG. 2A, a developer group A is responsible for interaction instruction development, and a developer group B is responsible for service processing development.

20 **[0057]** The developer group A can pre-determine the device type, and configure an instruction acquisition rule used to obtain the device interaction instruction initiated by the electronic device, and configure an instruction set conversion rule used to convert the device interaction instruction into the corresponding unified interaction instruction, for different device types.

25 **[0058]** The unified interaction instruction includes one or a combination of basic tapping, touching and holding, sliding to the left, sliding to the right, sliding up, sliding down, pinching and stretching, and rotating. If the developer group B has more needs, the developer group B can submit an instruction need to the developer group A, which can be a correspondence between a device type, a device interaction instruction,
30 and a unified interaction instruction. The developer group A further configures the instruction processing rule as needed by the developer group B.

[0059] The developer group A provides a unified interaction instruction SDK for the developer group B, and application developers can directly perform development by using the SDK, or can extend a customized instruction set by using an extension

module of the SDK. The SDK can be accessed by directly introducing an online resource (which is sent by using a javascript file) to webview, or can be directly packaged to an application as an offline resource for release in an application packaging and release phase. Therefore, an application range supported by the SDK covers a native application that is based on a specific operating system, and can also support a browser-based online web application, etc. An application range of the SDK can be relatively wide.

[0060] The developer group B does not need to consider adaptation between different devices, but only needs to configure the service processing mode needed by the unified interaction instruction. Development achievements of the developer group A and the developer group B are separately used as submodules of the application program. In practice, the service provider can develop more submodules having other functions as needed. Finally, the application program of the service provider can be provided for different users and installed on different electronic devices. The application program includes modules developed by the developer group A and the developer group B. FIG. 2B is a schematic architectural diagram illustrating an application program, according to an example implementation of the present application. FIG. 2C is a schematic diagram illustrating service processing of three electronic devices, according to an example implementation. A personal computer, a VR device, and a smartphone are included in FIG. 2C. Different electronic devices can perform the method in the implementation shown in FIG. 1 or FIG. 2D. Therefore, multi-terminal deployment through one set of code can be implemented, automatic adaptation supports a plurality of device interaction modes to achieve a same interaction effect, development costs can be reduced, and reusability is relatively high.

[0061] Specifically, with reference to FIG. 2B and FIG. 2D, after the electronic device is started, a service processing process is as follows:

[0062] End-user device detection: It is detected whether the electronic device is a VR device, an end-user device that supports a gravity sensor, an end-user device that supports a gesture event, or a mouse device.

[0063] Instruction receiving module: A VR instruction receiving module, a gravity sensing instruction receiving module, a gesture event instruction receiving module, or a mouse operation instruction receiving module is prepared in advance based on different device types, and a corresponding instruction receiving module is loaded based on a type of an end-user device.

[0064] Unified instruction set: A unified interaction instruction set is prepared in advance based on the different device types.

[0065] Instruction conversion module: A received instruction is converted into a unified interaction instruction based on the previously described loaded instruction receiving module.

[0066] Graph system: The unified interaction instruction is received, and an icon component is drawn based on the unified interaction instruction, to output the icon component to the electronic device for user check.

[0067] Corresponding to the previously described implementation of the service processing method, the present application further provides implementations of a service processing device and an electronic device that the service processing device is applied to.

[0068] The implementation of the service processing device in the present application can be applied to the electronic device. The device implementation can be implemented by using software, or can be implemented by using hardware or a combination of hardware and software. Using software implementation as an example, a device in a logical sense is obtained after a processor that performs service processing reads a corresponding computer program instruction in a non-volatile memory to a memory to run. From a perspective of hardware, FIG. 3 is a diagram illustrating a hardware structure of an electronic device that a service processing device is located in, according to the present application. In addition to a processor 310, a memory 330, a network interface 320, and a non-volatile memory 340 shown in FIG. 3, the electronic device that the device 331 in this implementation is located in usually can further include other hardware based on actual functions of the electronic device. Details are omitted here for simplicity.

[0069] FIG. 4 is a block diagram illustrating a service processing device, according to an example implementation of the present application. An instruction processing rule corresponding to each device type is configured according to the device, and the device includes the following: a type detection module 41, configured to detect a device type of an electronic device that needs to display interaction data; a rule acquisition module 42, configured to obtain an instruction processing rule corresponding to the detected device type; an instruction conversion module 43, configured to obtain, based on the instruction processing rule, a device interaction instruction initiated by the electronic device, and convert the device interaction

instruction into a corresponding unified interaction instruction; and a service processing module 44, configured to process the interaction data by invoking a service processing mode corresponding to the unified interaction instruction obtained through conversion, and provide processed interaction data for the electronic device for display.

[0070] In an optional implementation, the type detection module 41 is further configured to: initiate a device type query request to the electronic device; and receive device type description information fed back by the electronic device based on the query request, and determine the device type based on the device type description information.

[0071] In an optional implementation, the instruction processing rule includes an instruction acquisition rule used to obtain the device interaction instruction initiated by the electronic device, and an instruction set conversion rule used to convert the device interaction instruction into the corresponding unified interaction instruction.

[0072] In an optional implementation, the instruction processing rule is preconfigured in the following way: receiving a correspondence among a device type, a device interaction instruction, and a unified interaction instruction that are sent by a development side device, and configuring the instruction processing rule based on the correspondence.

[0073] In an optional implementation, a coding scheme for the unified interaction instruction is a coding scheme of one of different device types for a device interaction instruction.

[0074] Correspondingly, the present application further provides a service processing device, including a processor and a memory configured to store a processor executable instruction, where the processor is configured to: detect a device type of an electronic device that needs to display interaction data; obtain an instruction processing rule corresponding to the detected device type; obtain, based on the instruction processing rule, a device interaction instruction initiated by the electronic device, and convert the device interaction instruction into a corresponding unified interaction instruction; and process the interaction data by invoking a service processing mode corresponding to the unified interaction instruction obtained through conversion, and provide processed interaction data for the electronic device for display.

[0075] For specific implementation processes of functions and roles of the

modules in the service processing device, reference is made to implementation processes of corresponding steps in the service processing method. Details are omitted here for simplicity.

5 [0076] Because a device implementation basically corresponds to a method implementation, for related parts, reference can be made to partial descriptions in the method implementation. The described device implementation is merely an example. The modules described as separate parts can or cannot be physically separate, and parts displayed as modules can or cannot be physical modules, in other words, can be located in one location, or can be distributed in a plurality of network modules. Some
10 or all of the modules can be selected based on actual needs, to achieve the objectives of the solutions of the present application. A person of ordinary skill in the art can understand and implement the implementations of the present specification without creative efforts.

15 [0077] A person skilled in the art can easily figure out other implementation solutions of the present application after considering the specification and practicing the present application disclosed here. The present application is intended to cover any variations, functions, or adaptive changes of the present application. These variations, functions, or adaptive changes comply with general principles of the present application, and include common knowledge or a commonly used technical
20 means in the technical field that is not disclosed in the present application. The specification and the implementations are merely considered as examples, and the actual scope and the spirit of the present application are pointed out by the following claims.

25 [0078] It should be understood that the present application is not limited to the accurate structures that are described above and that are shown in the accompanying drawings, and modifications and changes can be made without departing from the scope of the present application. The scope of the present application is limited only by the appended claims.

30 [0079] The previously described descriptions are merely examples of implementations of the present application, but are not intended to limit the present application. Any modification, equivalent replacement, or improvement made without departing from the spirit and principle of the present application should fall within the protection scope of the present application.

CLAIMS

What is claimed is:

1. A service processing method, wherein an instruction processing rule corresponding to each device type is configured according to the method, and wherein
5 the method comprises the following:

detecting a device type of an electronic device that needs to display interaction data;

obtaining an instruction processing rule corresponding to the detected device type;

10 obtaining, based on the instruction processing rule, a device interaction instruction initiated by the electronic device;

converting the device interaction instruction into a corresponding unified interaction instruction;

15 processing the interaction data by invoking a service processing mode corresponding to the unified interaction instruction obtained through conversion; and providing processed interaction data for the electronic device for display.

2. The method according to claim 1, wherein the detecting a device type of an electronic device that needs to display interaction data comprises the following:

20 initiating a device type query request to the electronic device;

receiving device type description information fed back by the electronic device based on the query request; and

determining the device type based on the device type description information.

25 3. The method according to claim 1, wherein the instruction processing rule comprises an instruction acquisition rule used to obtain the device interaction instruction initiated by the electronic device, and an instruction set conversion rule used to convert the device interaction instruction into the corresponding unified interaction instruction.

30

4. The method according to claim 1, wherein the instruction processing rule is preconfigured in the following way:

receiving a correspondence among a device type, a device interaction instruction, and a unified interaction instruction that are sent by a development side device, and configuring the instruction processing rule based on the correspondence.

5 5. The method according to any one of claims 1 to 4, wherein a coding scheme for the unified interaction instruction is a coding scheme of one of different device types for a device interaction instruction.

10 6. A service processing device, wherein an instruction processing rule corresponding to each device type is configured according to the device; and wherein the device comprises the following:

 a type detection module, configured to detect a device type of an electronic device that needs to display interaction data;

15 a rule acquisition module, configured to obtain an instruction processing rule corresponding to the detected device type;

 an instruction conversion module, configured to:

 obtain, based on the instruction processing rule, a device interaction instruction initiated by the electronic device; and

20 convert the device interaction instruction into a corresponding unified interaction instruction; and

 a service processing module, configured to:

 process the interaction data by invoking a service processing mode corresponding to the unified interaction instruction obtained through conversion; and

 provide processed interaction data for the electronic device for display.

25

 7. The device according to claim 6, wherein the type detection module is further configured to:

 initiate a device type query request to the electronic device;

 receive device type description information fed back by the electronic device

30 based on the query request; and

 determine the device type based on the device type description information.

 8. The device according to claim 6, wherein the instruction processing rule comprises an instruction acquisition rule used to obtain the device interaction

instruction initiated by the electronic device, and an instruction set conversion rule used to convert the device interaction instruction into the corresponding unified interaction instruction.

5 9. The device according to claim 8, wherein the instruction processing rule is preconfigured in the following way:

receiving a correspondence among a device type, a device interaction instruction, and a unified interaction instruction that are sent by a development side device, and configuring the instruction processing rule based on the correspondence.

10

10. The device according to any one of claims 6 to 9, wherein a coding scheme for the unified interaction instruction is a coding scheme of one of different device types for a device interaction instruction.

15 11. A service processing device, comprising the following:

a processor; and

a memory configured to store a processor executable instruction;

wherein the processor is configured to:

20 detect a device type of an electronic device that needs to display interaction data;

obtain an instruction processing rule corresponding to the detected device type;

obtain, based on the instruction processing rule, a device interaction instruction initiated by the electronic device;

25 convert the device interaction instruction into a corresponding unified interaction instruction;

process the interaction data by invoking a service processing mode corresponding to the unified interaction instruction obtained through conversion; and provide processed interaction data for the electronic device for display.