

processing units (CPU), an input/output interface, a network interface, and a memory.

[0036] The memory can include a non-persistent memory, a random access memory (RAM), and/or a nonvolatile memory in a computer readable medium, for example, a read-only memory (ROM) or a flash memory (flash RAM). The memory is an example of the computer readable medium.

[0037] The computer readable medium includes persistent, non-persistent, movable, and unmovable media that can implement information storage by using any method or technology. Information can be a computer readable instruction, a data structure, a program module, or other data. An example of a computer storage medium includes but is not limited to a phase change memory (PRAM), a static random access memory (SRAM), a dynamic random access memory (DRAM), a random access memory (RAM) of another type, a read-only memory (ROM), an electrically erasable programmable read-only memory (EEPROM), a flash memory or another memory technology, a compact disc read-only memory (CD-ROM), a digital versatile disc (DVD) or other optical storage, a cassette magnetic tape, a tape and disk storage or another magnetic storage device, or any other non-transmission media. The computer storage medium can be configured to store information that can be accessed by a computing device. As described in the present specification, the computer readable medium does not include a transitory computer readable medium such as a modulated data signal and a carrier.

[0038] As shown in FIG. 2, according to an aspect of the present application, a method reusing a ListCell in a hybrid application is provided. The method includes the following steps.

[0039] Step S1: Define a prototype of a ListCell at a front end of the hybrid application.

[0040] Step S2: Obtain the ListCell through copying at a native end of the hybrid application based on the prototype of the ListCell, and reuse the ListCell. In the present application, the prototype (a parent) of the ListCell is defined at the front end. The prototype of the ListCell is copied (cloned) at the native end (Native). The native end reuses an invisible ListCell based on a current list presentation location. As such, the prototype of the ListCell can be defined at the front end, and the ListCell can be freely created, rendered, and reused at the native end. Therefore, not only the flexibility of defining the ListCell in the hybrid development can be reserved, but also the ListCell can be reused at the native end, thereby greatly improving the large-list

CLAIMS

1. A computer-implemented method, comprising:
 - defining, by a front end layer of a hybrid software application, a prototype of a ListCell;
 - creating, by a native layer of the hybrid software application, a ListCell template based on a created prototype of the ListCell defined by the front end layer;
 - obtaining, by the native layer, a ListCell by copying the ListCell template;
 - obtaining, by the native layer, first ListCell content transmitted from the front end layer;
 - filling, by the native layer, the ListCell with the obtained first ListCell content;
 - initiating, by the native layer, display of the ListCell having the first ListCell content; and
 - reusing, by the native layer, the ListCell, comprising:
 - in response to a user input, determining that the first ListCell content has been moved out of a display area of the native layer,
 - in response, removing the first ListCell content from the ListCell, obtaining second ListCell content transmitted from the front end layer, and
 - refilling the ListCell with the second ListCell content.
2. The computer-implemented method according to claim 1, wherein defining the prototype of the ListCell comprises defining a structure layout and a class of the prototype of the ListCell.
3. The computer-implemented method according to claim 1, wherein obtaining the ListCell comprises:
 - traversing and obtaining all sub-views of the ListCell template; and
 - creating a ListCell that comprises all the sub-views and attributes of the sub-views.
4. The computer-implemented method according to claim 3, wherein filling the ListCell comprises filling the obtained first ListCell content in each corresponding sub-view of the created ListCell, wherein the obtained first ListCell content is obtained from the front end of the hybrid software application.

5. The computer-implemented method according to claim 1, comprising, during obtaining the ListCell by copying the ListCell template, a number of copied ListCells is determined based on a number of ListCells that can be displayed in a computer display area.
6. A non-transitory, computer-readable medium storing one or more instructions executable by a computer system to perform operations according to the method of any one of claims 1 to 5.
7. A computer-implemented system, comprising:
 - one or more computers; and
 - one or more computer memory devices interoperably coupled with the one or more computers and having tangible, non-transitory, machine-readable media storing one or more instructions that, when executed by the one or more computers, perform one or more operations according to the method of any one of claims 1 to 5.