January 08, 2015

DYNAMICALLY SWITCHING APPLICATION INTERACTIONS BETWEEN DEVICES

Kenneth Mixter

DYNAMICALLY SWITCHING APPLICATION INTERACTIONS BETWEEN DEVICES

ABSTRACT

A user interacts with different devices, for example, mobile device, smartphones, laptops, and tablets, for accessing various applications. The applications have different device requirements. The device requirements can include preferred device specifications to run the application, e.g., processor speed, display screen size/resolution, operating system, and internet connectivity. A device switching system is used for dynamically switching application interactions from a device to another based on application's device requirements. The system detects that an application is running at a first device and determines the application's device requirements. The system compares the device requirements of the application with the device specifications of a second device. The system causes the application to be displayed at the output device of the second device and coordinates user interactions on the second device.

PROBLEM

Consumer applications are predominantly being developed for small form factor computing devices, while large form factor computing devices remain ideal for deep research and content creation. Examples of small form factor computing devices include mobile phones, smartphones, personal digital assistants (PDA), and handheld computing devices with display sizes typically 4″ diagonal or smaller. Applications developed for these devices include music, messaging, chatting, lower resolution games, etc. Examples of large form factor computing devices include notebook computers, desktop computers, and television displays with display

sizes typically 11″ diagonal or larger. Applications developed for large form factor devices include document editors, electronic mails, high resolution games, etc. However, the two form factor ecosystems remain largely disconnected. For example, a user of a mobile device comes across a long magazine article on a mobile device application, which the user will be more comfortable viewing on the user's laptop device. The user can view this article at the laptop device but the user needs to take additional steps to pull up the article on the separate device. The steps may be manually opening the article on a web browser application by memorizing the Uniform Resource Locator (URL) of the article webpage, sending a saved copy of the article via e-mail, or transferring the article file via a hardware storage device, e.g., a thumb drive. A system for interfacing the two form factor ecosystems is described.

DEVICE SWITCHING SYSTEM

The systems and techniques described in this disclosure relate to a device switching system that dynamically switches an application running on one device to another device based on application's device requirements. The device switching system can be implemented for use in an Internet, an intranet, or another client and server environment. The system can be implemented locally on a client device(s) or implemented across a client device and server environment. The client device can be any communication device, for example, laptop, mobile phone, computer, tablet, etc.

FIG. 1 illustrates an example method 100 for switching the display of an application running on a first device to a second device based on application's device requirements. The

method 100 can be performed by a system that dynamically switches the display of an application from one device to another device, for example, the device switching system.

As shown in FIG. 1, the system facilitates interaction between a first device and a second device (Block 102). The first device and the second device can have different device characteristics. For example, the first device and the second device can have different form factors, display screen sizes, processing power, synchronization, and bandwidth. Examples of first and second device include mobile phones, laptops, desktops, and tablets. The system may facilitate interaction between the first device and the second device using either wired input/output interfaces, such as USB, Firewire, or wireless input/output interfaces, such as WiFi, Bluetooth. The system uses appropriate protocols, for example, link management protocol and media transfer protocol for the interaction. Further, the interaction between the first device and the second device includes sending and/or receiving information between the first and the second device. Information includes data files like video, audio, etc. In an example, the first device and the second device are in a master and slave relationship. In this example, interactions also include issuing control commands by the master device for configuring certain slave device parameters like that of display, ports, etc.

The system detects an application running at the first device (Block 104). The application is presented at an output device associated with the first device. The system determines device requirements for the application running at the first device. The device requirements can include preferred device specifications to run the application, e.g., processor speed, display screen size/resolution, operating system, and internet connectivity. For example, a high-resolution gaming application requires powerful processor and large screen, high screen resolution device

requirements. The system can determine the device requirements from the application itself, or can determine the device requirements from a database that stores applications and their respective device requirements. The database may be stored at the server or at any memory of one or more of the devices.

The system causes the application to be displayed at an output device associated with the second device (Block 106). The system identifies the device specifications of the second device, e.g., processor speed, display screen size/resolution, operating system, and internet connectivity. The system can determine the device specifications from the device itself, or can determine the device specifications from a database that stores device specifications for specific devices. The system then compares the device requirements for the application with the device specifications of the second device. For example, the system compares the application display screen size/resolution requirement with the display screen size/resolution of the second device, the application processor speed requirement with the processor speed of the second device, etc. If the system determines that the device specifications satisfy the application's device requirements, then the system causes the application to be displayed and/or run at the output device associated with the second device. For example, if the application running on the first device requires a device that has optimal display greater than 14 inches, then the system determines if the second device has a display screen that fulfills the requirement. If the display screen fulfills the requirement, the system causes the application to be displayed at the output device associated with the second device. Additionally or alternatively, if the application running on the first device requires a device that has a greater processor speed, then the system determines if the second device has a processor speed that fulfills the requirement. If the processor fulfills the

requirement, the system causes the application to be run on the second device, and displayed at the output device of the second device.

Further, the system receives user interactions with the application at the second device (Block 108). The second device has an input device (such as keyboard or mouse) and an output device (such as display or audio speakers) for facilitating user interactions. The interactions with the application are coordinated by the system between the first device and the second device. For example, when the application is still running on the first device while it is displayed at the output device of the second device then the inputs received at the second device are transmitted to the first device. Suppose the application is a web browser application where the user is browsing a long magazine article and the user highlights a paragraph of the article at the second device. The inputs pertaining to highlighting are transmitted to the first device for synchronization. However, when the system causes the application to be run and displayed at the second device itself, the system receives and executes interactions with the application at the second device itself without needing to transfer the inputs received at the second device to the first device. Also, the user may navigate back to the first device after the desired interaction has been completed with the application at the second device or the system automatically redirects the interaction to the first device.

FIG. 2 illustrates example Graphical User Interfaces (GUI) for displaying an application on a first device and switching the application interaction to a second device. The GUI can be displayed on display screen or other output device associated with each of the two devices. The first and second devices have different device characteristics.

FIG. 2A illustrates a messenger application running at the first device, i.e., a mobile phone. The user interacts with the application and receives messages from a contact user. The user further interacts with the application by clicking on a uniform resource locator link sent by the contact user. The link opens up a long magazine article from a website. The system determines that the application interaction requires a device having a higher form factor than the mobile phone. The system based on the determination causes the application to be displayed at the laptop as shown in FIG. 2B. In an example, the system may display a pop-up user interface question taking permission from the user to switch the interaction to the laptop. Any other application's device requirement apart from the form factor can also be the reason for switching the application to a suitable device. Thus, the system makes it easier for the user to switch between connected devices for rich and obstruction free interactions and reduce cumbersome steps needed for doing the same manually.

The subject matter described in this disclosure can be implemented in software and/or hardware (for example, computers, circuits, or processors). The subject matter can be implemented on a single device or across multiple devices (for example, a client device and a server device). Devices implementing the subject matter can be connected through a wired and/or wireless network. Such devices can receive inputs from a user (for example, from a mouse, keyboard, or touchscreen) and produce an output to a user (for example, through a display). Specific examples disclosed are provided for illustrative purposes and do not limit the scope of the disclosure.

DRAWINGS

Figure 1

100

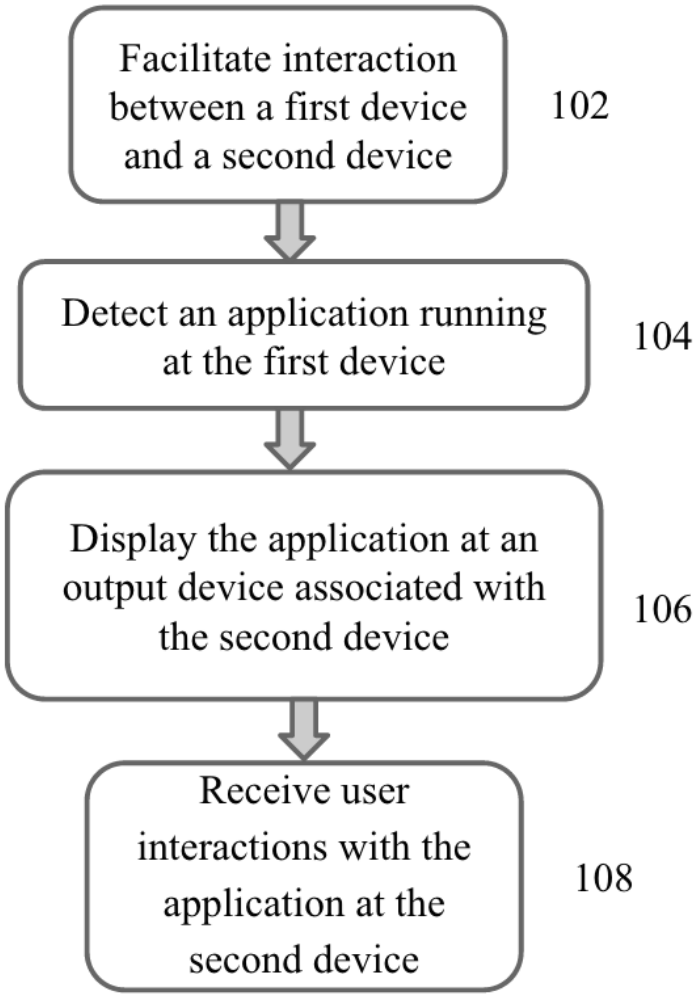

Figure 2

Figure 2A

Figure 2B