October 02, 2019

Sharing Application States Between Computing Devices

Jeff Pierce

Sharing Application States Between Computing Devices

Abstract:

This publication describes methods of sharing application states (tasks) between computing devices. The methods include steps for invoking options of sharing, determining if applications support sharing, acquiring tasks, presenting saved-task notifications, and synchronizing saved tasks across devices. Through the utilization of these methods, users may access applications at a saved state across multiple devices.

Keywords:

Uniform resource identifier, URI, uniform resource locator, URL, mobile deep linking, worldwide web resource, application software, application states, task states, share, transfer, save, encapsulation

Background:

Computing device users frequently coordinate activities with others, defer completion of activities until a more convenient time, and transfer activities to a more suitable computing device. These activities are often application based. For example, a computing device user (John) may desire to purchase ballgame tickets but needs to verify preferred seat locations with a friend. So, while interacting with a ticket purchasing application on his device, John needs to ask his friend to select where she wants to sit. When John reviews the options for sharing content in the application, he discovers that his device's share functionality is essentially limited to sharing content or links to content with his friend (*e.g.*, sharing the uniform resource locator (URL) of the

ticket order page, sharing a URL scheme link, sharing a screenshot of the ticket order page) and that he doesn't have a way to share the application state with his friend such that his friend can access and view the same application state on her device. In other words, John realizes that sharing content from the application does not afford his friend the ability to view the same application at the same state from which John shared it.

If John could share the application state with his friend, then she could easily access the application and select seats. As a result, it is desirable to integrate alternative methods of sharing application states that may afford users a more convenient and consistent experience on their computing devices.

Description:

This publication describes methods of sharing application states between computing devices (*e.g.*, mobile devices, tablets, laptops). Figure 1 illustrates an example computing device and elements of the computing device that support the methods described in this publication.

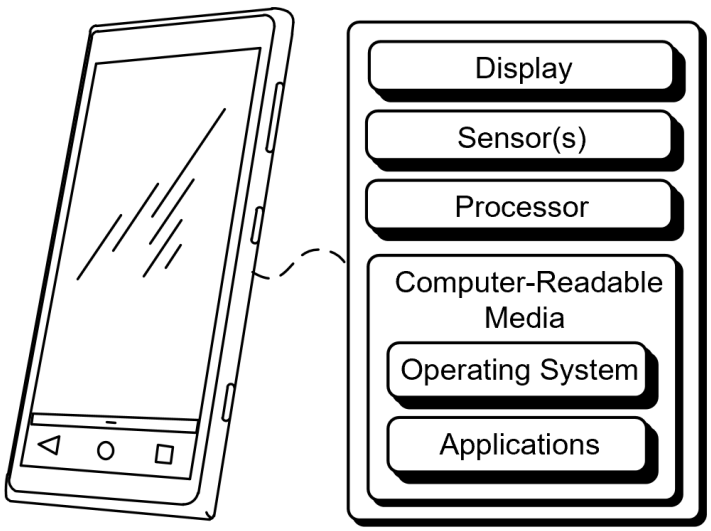

Figure 1

As illustrated in Figure 1, the computing device includes a display, sensors, at least one processor, and a computer-readable medium (CRM). The CRM may include the operating system (OS) of the computing device, applications, and executable instructions for sharing application states.

An application state (task), or more simply the state of an application, includes stored data relating to the status or progress in an application and is managed by the OS. The methods described in this publication permit users to share tasks between computing devices. One or more steps, executed in any order, may be utilized in the methods.

In a first step, a user of the computing device may invoke (*e.g.*, input commands by means of voice commands, touchpad gestures, or button sequences) options to share a task. For instance, while navigating in an application, a user may invoke options to share, save, or transfer a task by swiping fingers across the screen. Incorporating such invocation at the system-level can provide a consistent experience for users and encourage applications to implement broad support for sharing application states.

In a second step, the OS determines if an application supports application state sharing. Applications may indicate support by registering or declaring general support in their manifest (*e.g.*, files that contain information about the applications) or by proactively notifying the OS when a new task commences. An additional step of determining support includes, the OS dynamically querying an application's encapsulation (*e.g.*, the collection of data or methods under a single unit such as a class) state.

In a third step, the OS obtains an application state. The acquisition of an application state entails providing a hook (*e.g.*, code to intercept function calls) for applications to encapsulate tasks when a user selects an option to share, save, or transfer a task. Data acquired by the hook may

contain a deep link (*e.g.*, a uniform resource identifier (URI)) that encapsulates enough information for the application to reconstitute the task, a description of the task, an image of the current state, and a timestamp. Alternatively, a hook can be provided to applications so that they can save and pass state data. Finally, the hook can return a Boolean value to indicate whether the application generated the state.

In a fourth step, the OS can display saved-task reminders. Saved tasks can be displayed in a saved-task user interface, as a notification in the status bar, as a notification bubble superimposed on an application, or as a persistent indication of saved tasks presented on the home screen. Additionally, the new tab view of a web browser can present saved tasks.

In a fifth step, saved tasks can synchronize across multiple devices. For instance, a user may be navigating in an application between two devices. As the user works in the application on one device, the application progress can be saved using cloud computing and resumed on another device. Saved tasks can be deleted automatically after the user resumes the task. Alternatively, the OS can delete the saved task after a few days or another time period.

One or more of these steps implemented on a computing device to enhance sharing of tasks can be appreciated in the following example. A user (John) desires to purchase tickets to an upcoming ballgame, so he searches for tickets in an application on his mobile device. Upon finding available tickets, he invokes sharing options in the application and shares the application state with his friend. The friend can then view the application state on her device in an application or in a web browser. If the friend decides to view the content in an application, the same state from which John shared the task can be immediately viewed by the friend so she can easily select seats. In the interim, John can then save the task so that he can receive reminders to complete the transaction. After receiving a reply from his friend, he can then open the application via the saved task or by

selecting the shared task in his messaging application. Upon opening the task, John then transfers it to a more suitable computing device to finalize the transaction.

The implementation of these steps on John's mobile device is illustrated in Figure 2.

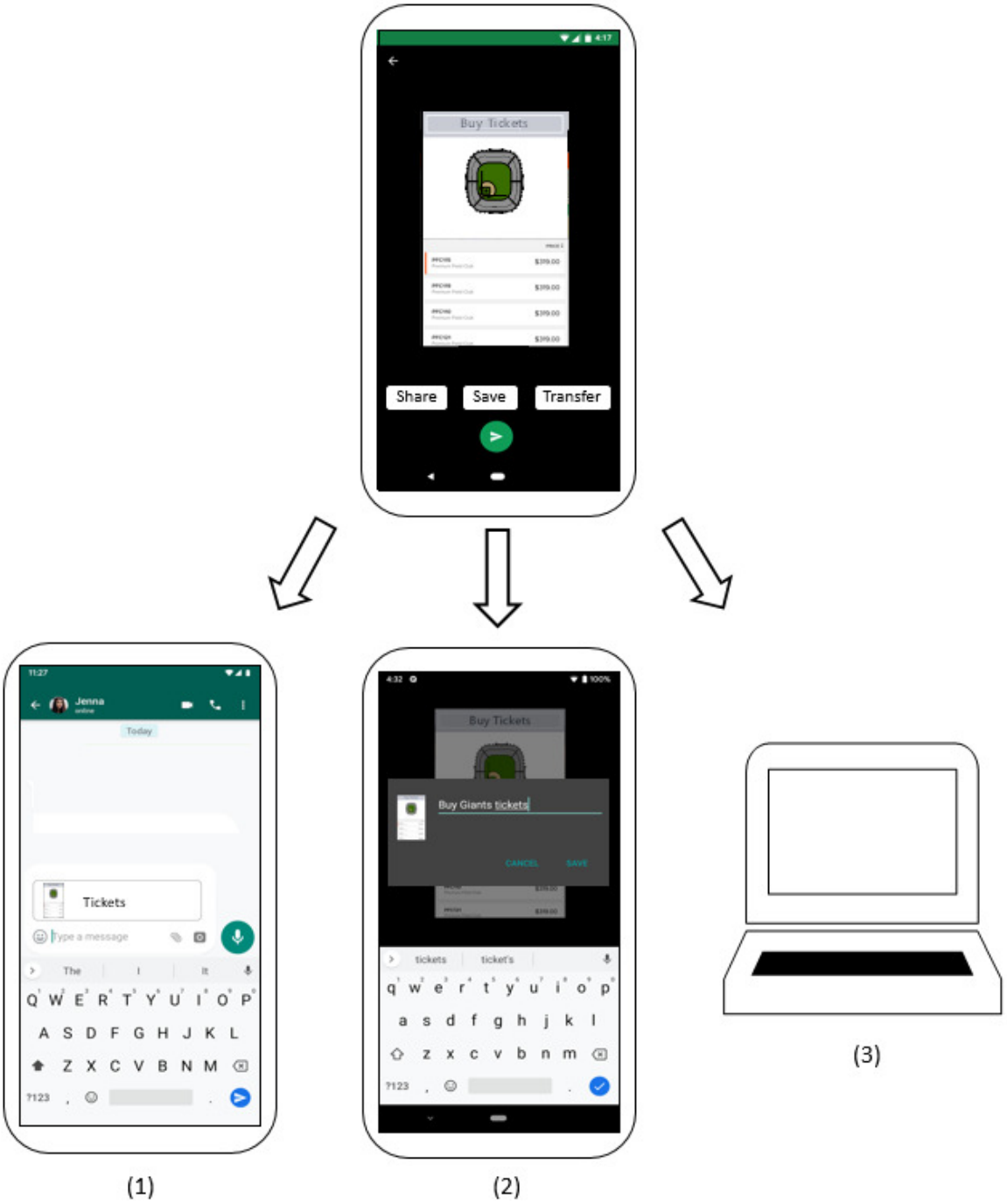

Figure 2

As Figure 2 illustrates, presented on John's mobile device are options to share an application state. John has the option to (1) share a task through a messaging app, (2) save the task for another time, or (3) transfer the task to a more suitable device, such as a computer. In all instances, the application state can be retained such that John can return to the state either in the application or a web browser.

An additional benefit of the disclosed methods includes greater potential for notification management. For example, notifications are often utilized as reminders or notices. Currently, users can interact with notifications in a variety of ways, such as address the notification immediately, dismiss the notification, ignore the notification, or snooze the notification. In many instances, however, users may desire to address the notification, but at a more convenient time. Thus, snoozing, dismissing, ignoring, or addressing the notification may not fully suit their intentions. The disclosed methods afford users the ability to save the notification as a task such that they can return to the saved task under more convenient circumstances.

References:

[1] Patent Publication: US20180001194A1. Method and system for sharing video game content. Filing Date: February 27, 2017.

[2] Patent Publication: US20160184712A1. Game State Save, Transfer and Resume for Cloud Gaming. Filing Date: January 22, 2015.

[3] "Everything we know about Google Stadia - so far; Google Stadia is the new streaming gaming platform from Google." Author(s): games.thisisalpha.com. Accessed: August 12, 2019. URL: https://games.thisisalpha.com/wp-content/uploads/2019/04/Everything-We-Know-About-Google-Stadia-So-Far.pdf.

[4] Polygon. Google Stadia is the new streaming gaming platform from Google. March 19, 2019. Accessed: August 15, 2019. https://www.polygon.com/2019/3/19/18272856/google-stadia-gdc-2019-announcement.

[5] "Create Deep Links to App Content." Author(s): Android Developers. Accessed: August 15, 2018. URL: https://developer.android.com/training/app-links/deep-linking.

[6] Pocket. Accessed: August 12, 2019. URL: https://getpocket.com/.

[7] Pushbullet. Accessed: August 12, 2019. URL: https://www.pushbullet.com/.

[8] "Use Handoff to Continue a Task on Your Other Devices." Apple Support. February 20, 2019. Accessed: August 12, 2019. https://support.apple.com/en-us/HT209455.