

Technical Disclosure Commons

Defensive Publications Series

January 08, 2015

BUFFERING AND INSERTING TEXT INPUTS

Winson Chung

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation

Chung, Winson, "BUFFERING AND INSERTING TEXT INPUTS", Technical Disclosure Commons, (January 08, 2015)
http://www.tdcommons.org/dpubs_series/3



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

BUFFERING AND INSERTING TEXT INPUTS

ABSTRACT

A text input buffering system stores text input in a buffer for a predetermined period of time. A user may input text in an application, for example, a browser, text editor, etc. If the system receives text input without an initial selection of a text field, the system stores the received text input in the buffer for a predetermined period of time. The system receives a selection of a text field. Further, the system displays a notification in the user interface for inserting the text input into the selected text field. The system inserts the text input into the text field on receiving an affirmative response from the user, else the system deletes the stored text input after the predetermined period of time.

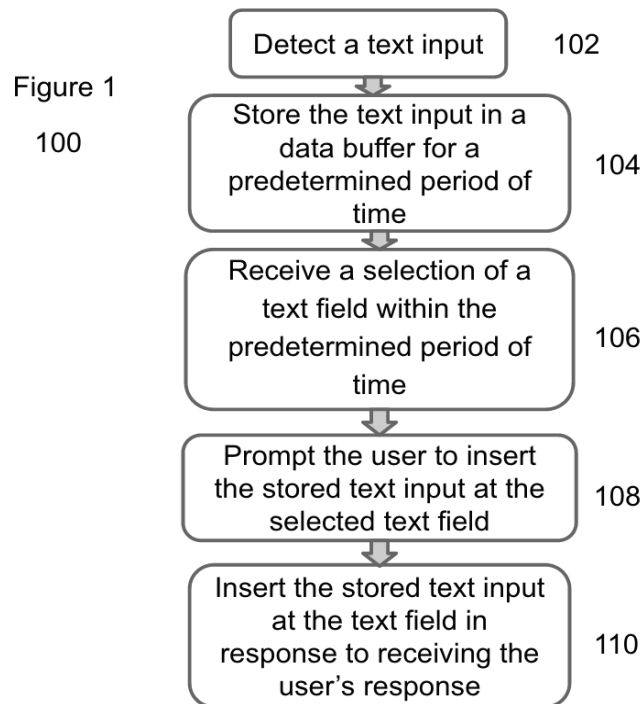
PROBLEM

Text input becomes problematic when users attempt to input text but a target text field is not identified. The text input is not piped through to the intended text field and is lost. This can result when the text field is not actually focused on or otherwise selected, or the UI finishes loading and changes the focus or selection. Users who focus their attention on the their keyboard rather than the display screen are particularly affected as they may be delayed in noticing their text inputs have not actually registered in their intended text field. The users will then have to select the text field and re-input the same text, resulting in repetition and lost time. A system is described that automatically stores the text input for a certain period of time.

TEXT INPUT BUFFERING SYSTEM

The systems and techniques described in this disclosure relate to a text input buffering system that stores text inputs in a buffer for a fixed interval of time in an application. The text input buffering system can be implemented for use in an Internet, an intranet, or another client and server environment, e.g., as part of software as a service on the cloud. The system can be program instructions implemented locally on a client device or implemented across a client device and server environment. The client device can be any electronic device, for example, laptop, mobile phone, computer, tablet, etc.

FIG. 1 illustrates an example method 100 for storing text inputs received from a user in a buffer and providing the text inputs at a later point in time. The method 100 can be performed by a system that is capable of storing text inputs and providing the text inputs within a predefined interval of time, for example, the text input buffering system.



The system detects a text input (Block 102). When a user is interacting with an application, e.g, a web browser or a web application on a web browser, on a client device, the user may input text. The text input can be received from any input device, e.g., a keyboard, touchpad, touchscreen, associated with the client device. The text input can be any combination of characters. For example, the text input is a string of characters that make up a partial or complete word or sentence. If a text field in the application is selected, the text input is input and displayed at the text field. Alternatively, the user may input text without an associated text field being first identified in the application, e.g., because a text field is not actually selected, or a UI finishes loading and deselects or changes the expected selection. On receiving the text input without an identified text field, the system stores the text input in a data buffer for a predetermined time period (Block 104). The predetermined time period may be a fixed amount of time, for example, 1 minute or 5 minutes. Alternatively, the predetermined time period may be based on the text input. For example, the predetermined time period may be proportional to the number of characters in the text input. Alternatively, or additionally, the predetermined time period may be algorithmically calculated by the system or set by the user. The system may provide a selection menu in a user interface for the user to set the predetermined time period via a settings menu. The user may also be able to disable the text buffering input functionality via the settings menu. The system can also display a countdown of the predetermined time period once the text input is received and stored. Additionally, or alternatively, the system can indicate that the text input has been stored, e.g., by providing a notification to the user.

The system then receives a selection of a text field within the predetermined period of time (Block 106). For example, the user, upon realizing that the text input was not input into a text field, selects the text field in the application where the user wishes to insert the text. For example, in a web browser application, the user selects the URL (uniform resource locator) bar to insert the text identifying a URL. In response to the selection of the text field, the system prompts the user to insert the buffered text input at the selected text field (Block 108). The system may display a pop-up notification in the user interface. Alternatively, or additionally, the system may produce an audible alert that notifies the user that there is stored text input available to insert in the selected text field. Alternatively, the system may automatically insert the stored text input into the text field as soon as the user selects the text field. If the system doesn't receive the selection of a text field within the predetermined time period, the stored text input in the data buffer is deleted. Alternatively, the text input can be stored in the buffer until the system receives a user's selection of a text field.

The system inserts the stored text input at the text field in response to receiving the user's response (Block 110). The system displays the inserted text input at the text field at the output device of the client device. Alternatively, if the user chooses to not insert the stored text into the text field, the stored text in the data buffer is deleted. In an implementation, if the system does not receive any response from the user, the system deletes the stored text input.

FIGURES 2A-2D illustrates an example Graphical User Interface (GUI) of a web browser application that receives text input, stores the text input in a buffer, and provides the text input when the user selects a text field, as facilitated by the text input buffering system.

FIGURES 2A-2D depicts a web browser application displayed at the display device, e.g., touch

sensitive screen, of the electronic device 200, e.g., a laptop. FIGURES 2A-2D illustrate the sequence of events that take place when the system receives text input to the time it is inserted into the text field.

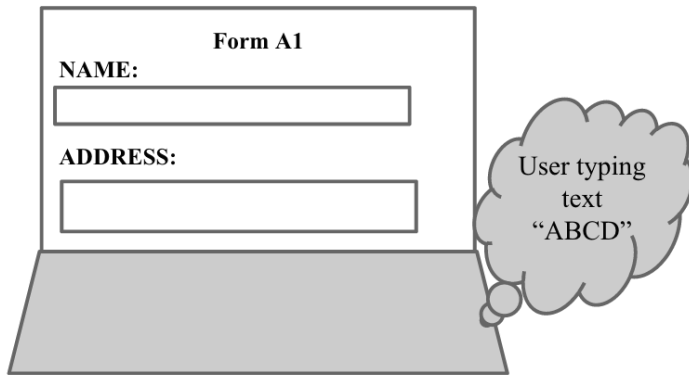


Figure 2A

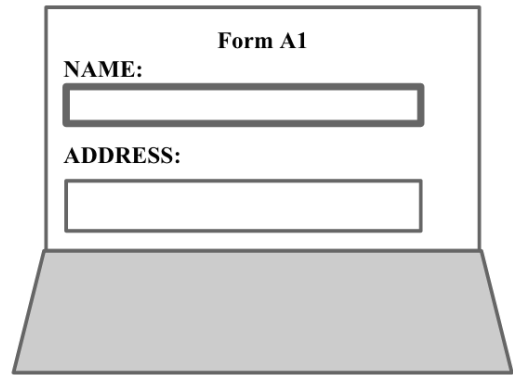


Figure 2B

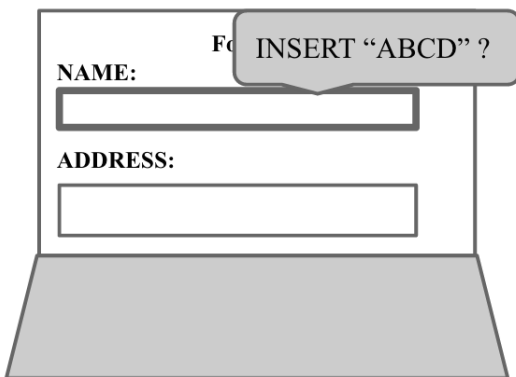


Figure 2C

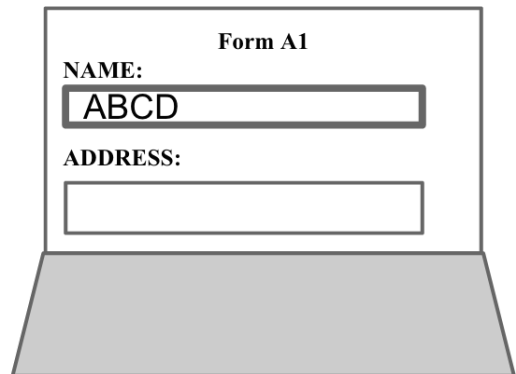


Figure 2D

The web browser application displays a website in the user interface. As shown in FIG. 2A, the website displays a form for the user to fill in. The user starts typing text to fill out the

form, but before a text field to input the typed text in is selected. The system receives the text input without the selection of the data field. The system stores the text input in the data buffer for a predetermined time period. As shown in FIG. 2B, the user then selects the text field, e.g., using a mouse or touching the appropriate location on the touch sensitive screen. The system provides the user with a pop-up notification “INSERT ABCD” in response to receiving the text field selection within the predetermined time period, as shown in FIG. 2C. If the system receives the user’s affirmative response to the pop-up notification, it inserts the stored text into the text field, as shown in FIG. 2D.

FIG. 3 is a block diagram of an exemplary environment that shows components of a system for implementing the techniques described in this disclosure. The environment includes client devices 310, servers 330, and network 340. Network 340 connects client devices 310 to servers 330. Client device 310 is an electronic device. Client device 310 may be capable of requesting and receiving data/communications over network 340. Example client devices 310 are personal computers (e.g., laptops), mobile communication devices, (e.g. smartphones, tablet computing devices), set-top boxes, game-consoles, embedded systems, and other devices 310’ that can send and receive data/communications over network 340. Client device 310 may execute an application, such as a web browser 312 or 314 or a native application 316. Web applications 313 and 315 may be displayed via a web browser 312 or 314. Server 330 may be a web server capable of sending, receiving and storing web pages 332. Web page(s) 332 may be stored on or accessible via server 330. Web page(s) 332 may be associated with web application 313 or 315 and accessed using a web browser, e.g., 312. When accessed, webpage(s) 332 may be transmitted and displayed on a client device, e.g., 310 or 310’. Resources 318 and 318’ are

resources available to the client device 310 and/or applications thereon, or server(s) 330 and/or web pages(s) accessible therefrom, respectively. Resources 318' may be, for example, memory or storage resources; a text, image, video, audio, JavaScript, CSS, or other file or object; or other relevant resources. Network 340 may be any network or combination of networks that can carry data communication.

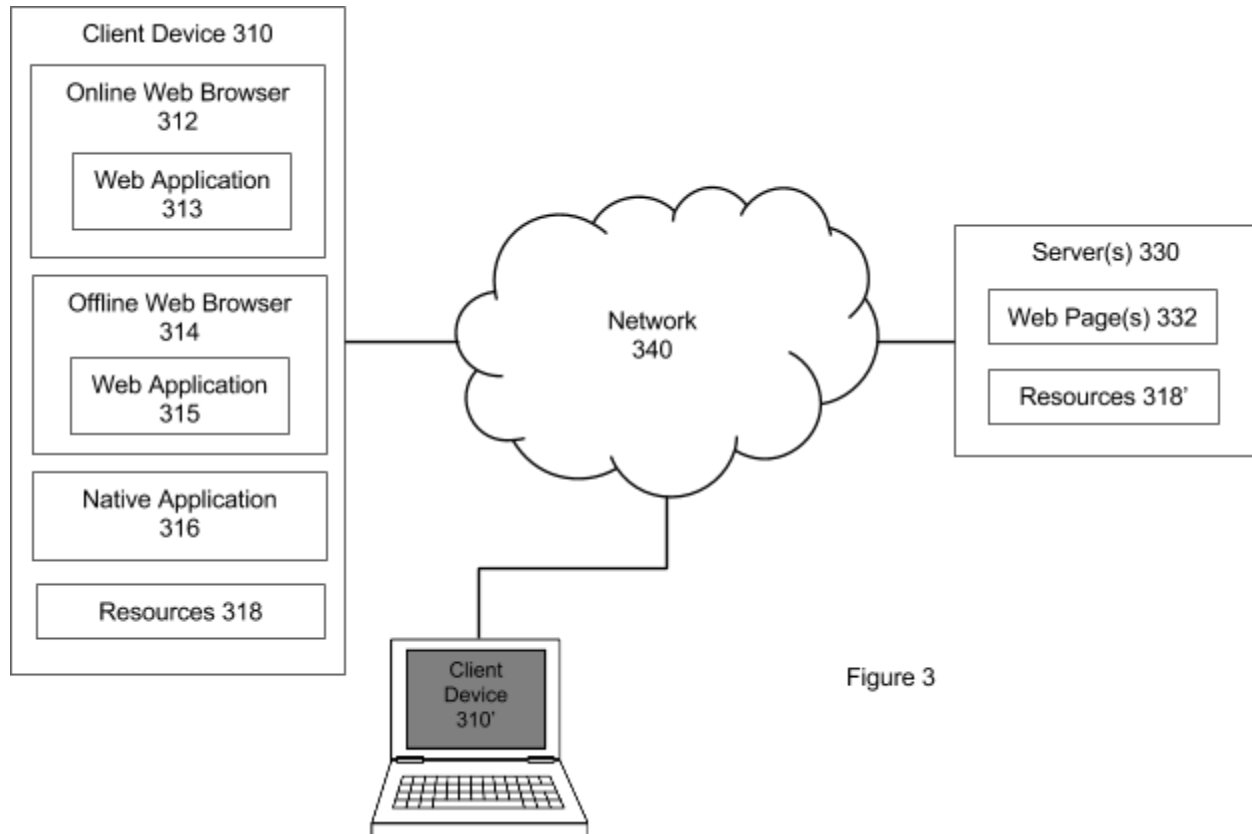


Figure 3

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.