DETERMINATION OF A PRIMARY WINDOW

ABSTRACT

A primary window management system can be used to determine a primary window from a multitude of windows opened in one or more applications. Applications may include web browsers, text editors, raster graphics editors, word processors, etc. The system determines the primary window from the multitude of windows based on user input. Then, the system modifies user interfaces of the one or more applications to provide easy navigation to the primary window from other windows.

PROBLEM STATEMENT

Many applications today allow multiple windows to be opened in a single parent window. These applications may use a graphical control element such as tabs for navigation between sets of windows. Such applications may include web browsers, text editors, raster graphics editors, word processors, etc. The usage of tabs as a navigational widget has become dominant in devices particularly with wide display screen area such as desktop computers, laptops, etc.

However, in scenarios where a user focuses working in one particular window and uses other windows to help with the work on the particular window, it often becomes difficult to navigate to the particular window. For example, when writing a document in a particular window, the user may have a window opened with a search engine to search for information, a window opened with a related presentation, another window opened with a related email, and so on. Frequently navigating between the windows and then back to the particular window that has
the document opened is inconvenient for the user. Thus, a system for determining a primary (or central) window is described.

**PRIMARY WINDOW MANAGEMENT SYSTEM**

The systems and techniques described in this disclosure relate to a primary window management system that determines a primary window from a multitude of windows opened in one or more applications. The system provides easy navigation between the primary window and the multitude of windows by modifying user interfaces of the one or more applications. The primary window management 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 or implemented across a client device and server environment. The client device can be any electronic device, for example, laptop, mobile phone, computer, and tablet.

Fig. 1 illustrates an example method 100 that provides a modified user interface of an application for easy navigation between sets of windows opened in the application. The method 100 can also be used for multitude of windows opened at a plurality of applications. The method 100 can be performed by a system, for example, the primary window management system, that determines a primary window from a multitude of windows opened in one or more applications and modifies user interfaces of the one or more applications. The system can be implemented at an electronic device running the applications.

The primary window management system receives a user input at an application with a multitude of windows displayed as tabs (110). In one example, the application can be a web
browser application as shown in Fig. 2 with a multitude of windows opened and displayed as tabs “TAB 1,” “TAB 2,” “TAB 3,” and “TAB 4.”

Further, the user input may be of different types such as explicit user input and implicit user input. Explicit user input may include instructions from a user identifying a window from a multitude of windows as a primary window. Such instructions can be received via a selection or identification of the primary window in a user interface as the primary window from the user. Implicit user input may include receiving frequent navigations to a particular window from the multitude of windows by a user. The user may use an input device for navigation, for example, mouse, keyboard, and touchscreen of the electronic device. In the example of Fig. 2, the user may be drafting an email at “TAB 2” and using “TAB 3” and “TAB 4” for copying text and pasting it in the email at “TAB 2.” The system receives these navigation inputs from the user and accordingly determines that the user frequently reverts to “TAB 2.”

The system then determines a primary window from the multitude of windows based on the received user input (120). In the above example, the system uses the implicit user input to determine that the user is frequently reverting back to “TAB 2” from other windows. Thus, the system determines that the “TAB 2” is a primary window from all the windows displayed, as shown in Fig. 2. Additionally, or alternatively, the system may observe a time that the user has spent on each of the multitude of windows and determine the primary window based on the greatest amount of time that the user spends on a particular window.

Further, the system may determine the primary window from the multitude of windows based on receiving an explicit selection of one of the windows as the primary window. For example, the user may select a user interface element associated with one of the windows to
identify that windows as the primary window. In another example, the system may provide a
suggestion to the user for marking a window as the primary window based on the usage statistics
of the window, for example, amount of time spent on the window and number of navigation
instructions received back to the window from other windows.

Upon determining a primary window, the system modifies a user interface of the
application based on the determined primary window (130). In one example, the system
highlights the determined primary window by, for example, making it a different color from
other windows or tabs, increasing the size of the tab or window borders, and bolding the title of
the window. As shown in Fig. 2, the primary window, “TAB 2,” is highlighted and bolded.
Additionally, or alternatively, the system may dynamically rearrange the order of windows such
that the primary window is always immediately adjacent to the window the user is currently
active in, e.g., a new window the user has just opened.

In another example, the system provides a shortcut key to navigate back to the primary
window directly from the current window. This shortcut key can be defined by the user, a
manufacturer of a device running the application, the application provider, or a provider of the
primary window management system. In one example, this shortcut key can be “x+y” (where x
and y can be any key) defined by the user such that the system automatically navigates back to
the primary window whenever the command “x+y” is received at any window other than the
primary window. In another example, the user may select another existing command such as the
copy command as a shortcut key. After the system modifies the user interface of the application,
the user can easily and quickly navigate to the primary window.
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**

![Diagram](image_url)

**Fig. 1**
Fig. 2