Window Manager Designed for Cloud Services

Shizuki Yoshino, Tetsuo Tanaka, and Kazunori Matsumoto

Department of Information and Computer Sciences, Kanagawa Institute of Technology, Atsugi, Japan
yoshinoshizuki@gmail.com,
{t-tanaka,matumoto}@ic.kanagawa-it.ac.jp

Abstract. Cloud services like web-based e-mail or hosted office suites are becoming widespread. With these services, PC users are likely to use several services and to visit several sites at once. As a result, several windows appear on the desktop, and their overlapping complicates access to hidden windows. In this study, the authors propose a window manager running on the browser. The proposed window manager employs a tiling style in order to improve the usability of multiple cloud services at the same time. It also employs a window placement method, implemented by drawing frame edges, and a window replacement method using drag and drop. It is user-friendly, even for unskilled PC-users. An experiment showed that the proposed window manager was effective in reducing the number of operations for window placement or replacement.

Keywords: window manager, cloud service, drawing frame edges, tiling style, window placement, usability.

1 Introduction

Cloud services, like web-based e-mail or hosted office suites, are becoming widespread. With such services, PC users are more likely to use several services and to visit several sites at once, e.g. document preparation while consulting a net dictionary, or Internet surfing while watching a movie. As a result, several windows appear on the desktop, and their overlapping complicates access to a hidden window. Reducing the cost of window operations is an important challenge. According to an analysis by Shibata [1], workers at an intellectual property management department spent 7.4–9.1% of time in window operations such as activating, moving, and resizing windows.

A window manager may be used to operate multiple windows efficiently. Window managers are classified into two categories, overlapping style and tiling style [2][3].

Most current window managers follow the independent overlapping window style. In this style the user manages a window's location and size in any way desired. When the location and size of a window changes, other windows may be obscured. When a window is hidden behind the front window, the user needs to move the window to the front in order to see the contents.

S. Yamamoto (Ed.): HIMI/HCII 2013, Part I, LNCS 8016, pp. 157–166, 2013.

The window manager of MS Windows 7 has a feature called Aero Snap[4]. Using this function, users can automatically resize the window to half the size of the display screen by dragging and dropping to the edge of the screen. But the size of the window is fixed at half of the screen. If there are three or more windows, windows overlap and rear windows are hidden. Moreover, users have to turn the function on or off in the control panel, and it is troublesome to re-enable the function if users have disabled it.

With the tiling style, any window is always fully visible, because windows are not allowed to overlap. It is easy to figure out where each window is. However in existing window managers like dwn[5] and so on, a user has to enter a numeric value in pixels to set the position and size of a window. It is not easy to use for unskilled users.

Shibata et al.[6][7] proposed a system that made it possible to construct workspaces consisting of multiple windows through a simple user interface of window docking. The system makes it possible to switch workspaces easily, and reduces the overhead of window operations within workspaces by use of a tiled window approach. Their experiment of a task-switching task showed that the system is 14% to 21% faster than a traditional window system. However, when using the system, the user builds a workspace by docking, after starting multiple windows for the task and so time is spent on the placement of the windows.

Meanwhile, there has been a wide introduction of cloud services, such as Google Drive[8], which lets you store all your files including documents, photos, videos and Google Docs online and access them from anywhere, Microsoft Office Web Apps[9], which is a web-based version of the Microsoft Office productivity suite, Facebook[10], which is a social networking service, and so on. As a result, a window manager for the cloud services is required. But existing window managers are designed for managing desktop applications. There are few designed for cloud services.

CEITON technology Inc. offers a window manager WinLIKE[11] running on a browser. With WinLIKE, web applications, websites and portals can contain real windows just like a normal desktop application. However, it adopts an overlapping window style, and so it suffers from the above-mentioned problems of overlapping windows. Moreover, it takes time to get used to using it, because its user interface is a CEITON proprietary one, and it requires users to enter a URL to launch cloud services.

In this study, the authors propose a window manager that is developed as a cloud service running on a HTML5 enabled browser, such as Chrome or Firefox. The proposed window manager employs a tiling style in order to improve the usability of multiple cloud services at the same time. It also employs a window placement method, based on drawing boundary lines, and a window replacement method, based on drag and drop. It is user-friendly, even for unskilled PC-users. In addition, it can save and restore a workspace consisting of a collection of multiple windows. Users can switch tasks smoothly.

2 Approach

2.1 Tiling Style Window Manager Running on a Browser

Cloud services such as Web-based office software and social networking services have become widespread. In these cloud services, users don't have to download the application program; data can be stored on the server. Users can use them at any time, anywhere, on any PC or mobile device in the Internet environment. Leading companies like Microsoft and Google provide cloud services. These results suggest that the trend toward cloud services will continue to accelerate in the future.

In order to efficiently use multiple cloud services at the same time, we propose a window manager running on a browser. WinLIKE, described above, is a window manager for the cloud service, but because it uses an overlapping style, it requires a lot of operations for switching, moving, and resizing a window. There is a way to spread multiple browser windows over the desktop using the tiling window manager for the desktop, but with this approach, the proportion of the screen taken up with toolbars or menu bars at the top of each browser window increases, and the area available for the application body is reduced. In the proposed window manger, we adopt a tiled window manager that spreads multiple cloud service windows over a single browser window.

2.2 Intuitive Window Operation

In WinLIKE and dwn, because users must set the window size and position by means of numeric input, unskilled PC users do not really know where the window will be or how large it will be. For example, a user may need to modify the input number several times before succeeding in resizing the window to half the size of the desktop screen. It also takes time to move the cursor to the edge of the window to resize it by dragging the edge. If a user wants to swap a window, he or she must resize and reposition both of the two windows.

In order to overcome the difficulty of using the resize function by means of numeric input or dragging edges, this window manager provides three functions, namely adding and removing a window frame by drawing frame edges, automatic placement of the window, and moving windows using drag and drop. To add or to remove a window frame, the user draws frame edges in the window manager. The position and size of a window frame are determined by the frame edges. When a cloud service is invoked, the window is automatically opened at the size of window frame that has been set. If a user wants to swap the windows in the right half and the left half, this can be achieved by dragging the left half of the window, which the user wants to move, and dropping it into the right half of the window.

To use a cloud services, a user normally needs to enter the appropriate URL. To eliminate this operation, the window manager provides a menu bar at the top. The user clicks the cloud service menu item he or she wants to use, and the cloud service window will be displayed in the browser. The menu displays the cloud services that are commonly used. This makes it possible to launch a cloud service that the user

wants to use in an instant, without entering the URL. At present these services are pre-determined in the window manager, so a user cannot change them, but this option may be possible in the future.

Fig.1 shows an example of a screen including windows of a word processing service, a dictionary service, and a video site. Items (a) are frame edges that have been drawn. Each window is resized to fit into the frame enclosed by the drawn lines. Items (b) are the drag and drop bars for moving or swapping windows. Item (c) is the menu bar; using this menu bar, users can start a cloud service with a single click, without having to enter configuration details in a dialog window.

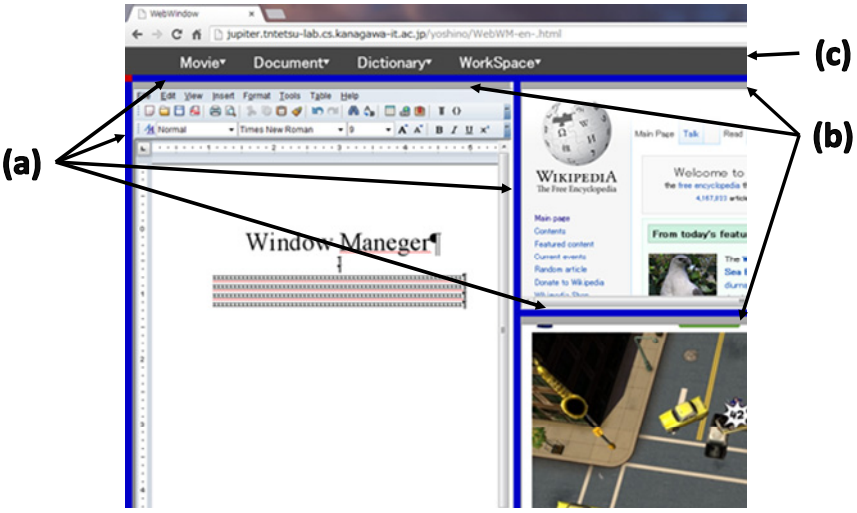

Fig. 1. Appearance of Window Manager

2.3 Operating Multiple Windows Collectively as a Workspace

When undertaking work, a user typically opens multiple application programs, and positions each application window in a place where it is easy for them to work. In other words, each user builds his/her own workspace specific to the job in hand. The applications which need to be open are different for each item of work. Therefore, users have to build a workspace to suit each job from scratch. Before users can start work on another job, it takes time if it is necessary to build a workspace from scratch. When a user suspends and then resumes his/her work, or when a user wants to use the same workspace many times, it is convenient if he or she can save and restore the workspace. This makes it possible to reduce the time spent on window operations.

This window manager is able to manipulate multiple windows collectively as a workspace in two ways: an operation on the browser window and a function of saving and restoring a workspace. Since the window manager runs on a browser, it is possible for users to manage a complete workspace containing several applications by a single operation on the browser window (resize, move, minimize, maximize). For example, for a brief interruption of work, the user only needs to minimize the browser

window. In addition, to meet requirements of a longer suspension of work, the window manager provides functions to save and restore the workspace. This is done from the menu item “Workspace” on the menu bar. Using this function, the user can avoid having to rebuild a workspace from scratch.

3 Functions of the Window Manager

This window manager is written in JavaScript and HTML5, and has the following five functions: (1) cloud service launch, (2) adding or removing a window frame by drawing, (3) automatic placement of a window, (4) moving a window, (5) and saving and restoring a workspace. Details are described below.

3.1 Cloud Service Launch

Using the menu bar, as shown in Fig.2, users can invoke cloud services. In this window manager, users can invoke three types of services, i.e., office software, video sites, and dictionaries. In the current implementation, we use Hanscom’s ThinkFree, which is compatible with Microsoft Office, as office software, Dailymotion as the video site, and Wikipedia, FreeTranslation and goo as dictionaries. Because the menu bar is similar to sites that have many users such as Google and Facebook, users can quickly grasp how to handle it. The menu has a hierarchical structure.

Fig. 2. Menu Bar

3.2 Adding and Removing a Window Frame by Drawing Frame Edges

In order to determine the position and size of the window, instead of a method of entering a numeric value in a dialog box, we have adopted a method of drawing frame edges. Users can determine the position and size of a window frame by drawing horizontal and vertical lines on the screen with a mouse click.

As shown in Figure 3, clicking point (a) on the frame edge (i), which is drawn in advance, a new frame edge line (iii) starting from point (a) is drawn. The frame edge

line (iii) is deleted when a user clicks on the point (a) again. A newly drawn frame edge (iii) can be manipulated in the same way, so clicking on point (b) will produce a new frame edge (iv). With this function, users can grasp the size of the window intuitively and visually. As a result, users can reduce the number of operations involved in resizing a window.

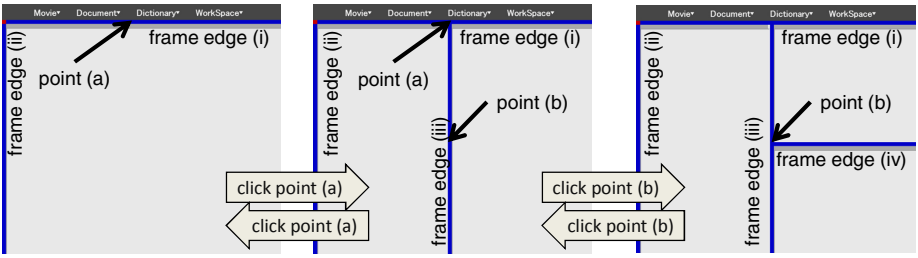

Fig. 3. Drawing frame edge lines

3.3 Automatic Placement of a Window

This function places the cloud service window to fit the drawn frame edge lines automatically. As shown in Fig.4, if the border lines have already been drawn, that is, if there is more than one empty window frame, the window manager displays the cloud service window in an empty frame in order starting from the top left frame. If there are no empty frames to display the window, the user can creates empty frames by adding frame edge lines, or a new window will overwrite the oldest one. When the user deletes a frame edge line, the frame on the extreme right or the bottom is removed, and the upper or left frame is expanded to cover the entire available area.

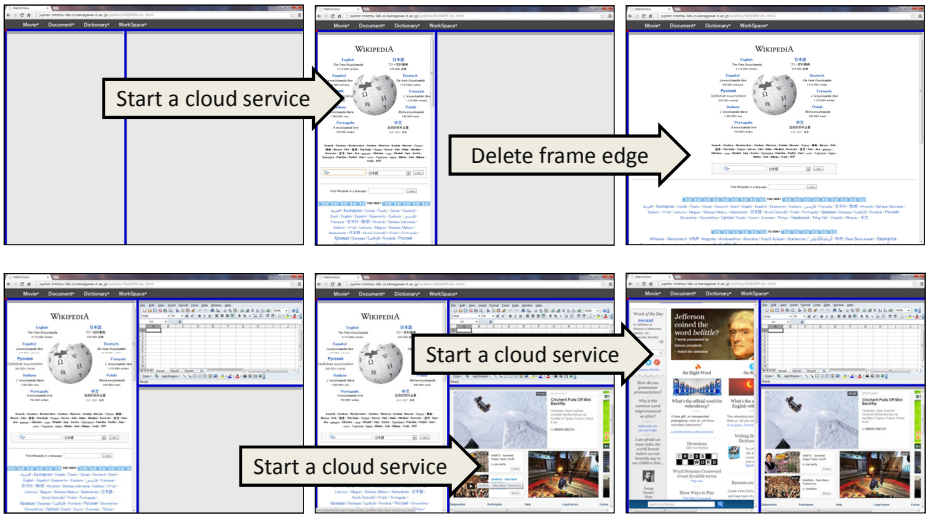

Fig. 4. Automatic Placement of Windows

3.4 Moving a Window

A window may be moved by the user dragging and dropping the window of a cloud service into another window frame. As shown in Fig.5, if a user wants to move the window in the frame on the left to the frame on the right, it is only necessary to drag the gray “drag and drop bar” at the top of the window and drop it onto the drag and drop bar on the right hand side. If another window is already displayed on the right, the positions of the windows are swapped with each other.

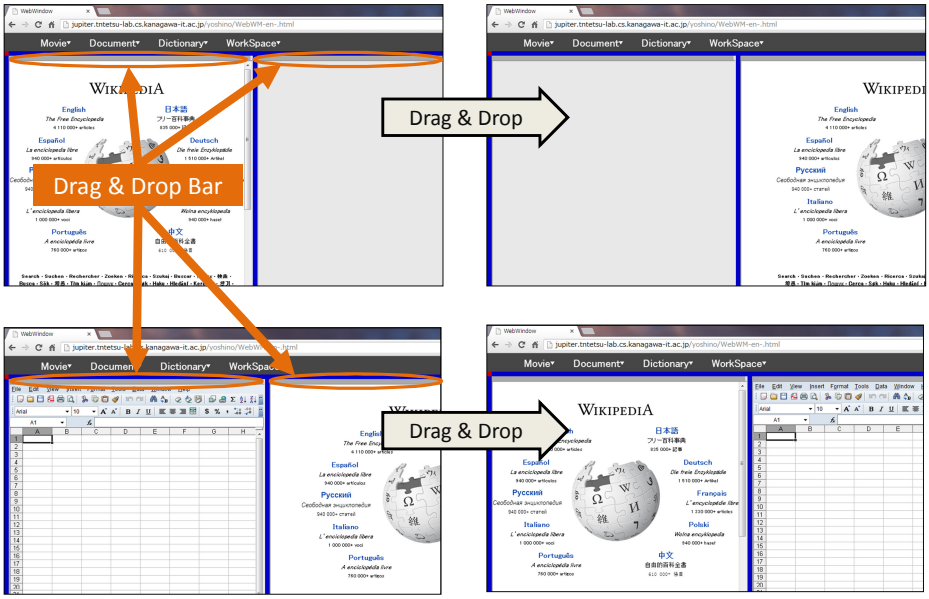

Fig. 5. Moving a Window

3.5 Saving and Restoring a Workspace

The list of window states that include position, size and the cloud service running on each window will be referred to as a “workspace”. Workspaces can be saved or restored using the menu item “Workspace” on the menu bar. Using this function, the user can avoid having to rebuild a workspace from scratch.

4 Implementation and Evaluation

4.1 Implementation

In order to run the window manager on a browser, it was developed as a web application (as a cloud service) using HTML5, JavaScript as shown in Fig.6. The frame edge line drawing function is implemented using the Canvas feature of HTML5. The workspace store function is implemented by the Local storage feature of HTML5.

In addition, the moving window function is implemented by the drag and drop feature of HTML5. A window of a cloud service is embedded in the window manager by using the inline frame element of HTML.

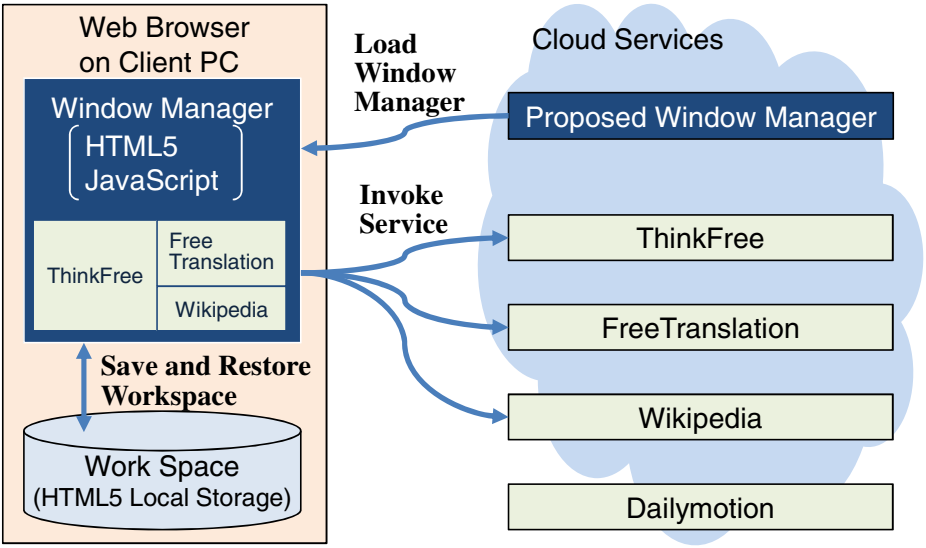

Fig. 6. Structure of the Window Manager

4.2 Evaluation

The effectiveness of this window manager was evaluated by comparison with Windows (with Aero Snap enabled), Windows (with Aero Snap disabled), and Win LIKE. In this experiment, we compared the number of mouse clicks and drags based on a scenario that consisted of the following seven steps:

1. Start the window manager
2. Launch three cloud services (i.e. a word processor, a presentation software, and a dictionary). Set the word processor window to be placed in the left half of the screen, arrange the presentation software window to be placed in the lower right quarter of the screen, and arrange the dictionary window to be placed in the upper right quarter.
3. Swap the word processor window with the presentation software window.
4. Save a document and a presentation, and save the workspace that has the current state of windows if possible.
5. Close the word processor and the dictionary, open a video site, set the video site window so as to be placed in the right half of the screen.
6. Exit the window manager and all cloud services.
7. Start the window manager again, start each cloud service and reconstruct window arrangement as it was stored in step 4.

Fig.7 shows the results of the evaluation. Thirty operations were required in Windows with Aero Snap disabled. When changing the window size by dragging, it was not possible to select the desired size with a single drag; it was necessary to make additional "tweaks" to the position when placing a tile, and in practice it takes thirty operations or more. In Windows, when Aero Snap is either enabled or disabled, the window manager does not remember the size and position of the window. Therefore, in order to reconstruct the window arrangement in step 7, the same operation as step 2 is required. WinLIKE requires a lot of operations to launch a new cloud service and to resize a window. This is due to the use of a dialog window to set the screen parameters when starting a new cloud service.

The experiment showed that the proposed window manager was effective, that is to say, it reduces by about half the number of operations for window placement or replacement compared with other window managers.

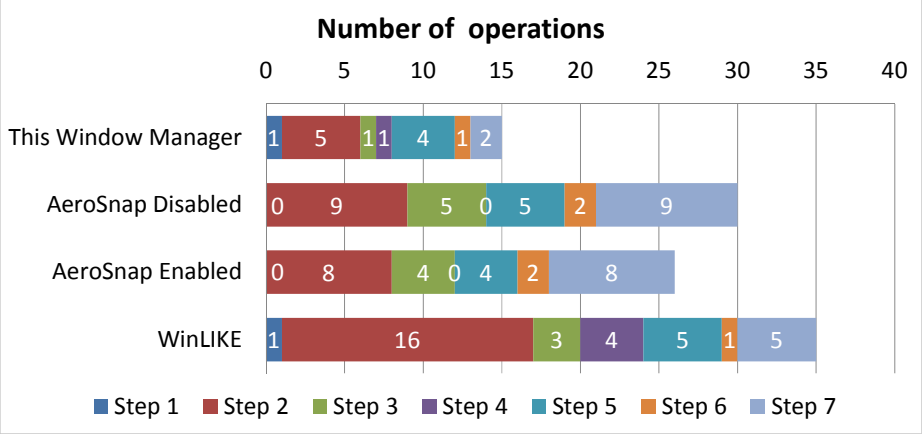

Fig. 7. Comparison of number of operations for different methods of managing windows

5 Conclusions

In this paper we have proposed a window manager designed for cloud services. The proposed window manager employs a tiling style in order to improve the usability of multiple cloud services at the same time. The main feature of the window manager is a window placement function based on drawing frame edge lines and a window replacement method by drag and drop. An experiment has shown that the proposed window manager reduces the number of operations for window placement or replacement.

References

1. Shibata, H.: Measuring the Efficiency of Introducing Large Displays and Multiple Displays. IPSJ Journal 50(3), 1204–1213 (2009) (in Japanese)
2. Bly, B., Rosenberg, J.: A Comparison of Tiled and Overlapping Windows. In: Human Factors in Computer Systems: CHI 1986 Conference Proceedings, pp. 101–106 (1986)

3. Myers, B.A.: Window Interfaces: A Taxonomy of Window Manager User Interfaces. IEEE Computer Graphics & Applications 8(5), 5–84 (1988)
4. Aero Snap, http://windows.microsoft.com/en-US/windows7/products/features/snap
5. dwn, http://dwm.suckless.org/
6. Shibata, H., Omura, K.: The support of multitasking through window docking. In: Proc. of IPSJ Interaction 2011 Conference Proceedings, pp. 391–394 (2011) (in Japanese)
7. Shibata, H., Omura, K.: Docking Window Framework: Supporting multitasking by docking windows. In: Proc. of the 10th Asia Pacific Conference on Computer Human Interaction, APCHI 2012 (2012)
8. Google Drive, https://drive.google.com/
9. Facebook, http://www.facebook.com/
10. Microsoft Office Web Apps, http://office.microsoft.com/en-us/web-apps/
11. CEITON technologies, The World Wide Web's Next Step Ahead: Web Window Manager WinLIKE – Empowering Websites and Web Applications (2003), http://www.ceiton.com/CMS/EN/press/WinLIKE_pr_001_information.pdf