Desktop Environment as a Problem Solving Tool
What users should demand from their desktops and desktop developers should aspire to provide users.

Robert Pastel

Introduction

The computer desktop environment, since its introduction in Xerox’s Star computer and popularization by Apple’s Macintosh computer [1], is the favorite general-purpose computer interface. The desktop provides graphical access to common utilities and important rudimentary tools for organizing the user’s work in windows. Linux distributions recognize the importance of the desktop and provide multiple customizable desktops, but these desktops are not significantly different from the original 1980’s desktops. The desktop can and should offer environments for efficient working and assist in creative problem solving. Because of the traditions of openness and innovation in the Linux community, I believe that Linux systems are capable and primed to develop desktop environments that truly assist the user in problem solving. This article explores the role of the desktop as a problem-solving tool and provides design considerations for desktop developers.

What is the role of desktops in problem solving?

Using the computer typically involves writing documents, drawing graphics and other complex information management tasks. The desktop’s role in problem solving is not to perform these specific tasks but rather to organize the work at a higher level by providing access to the specific applications and data files, and also to manage the display of application windows. Nearly all desktops offer integrated tools for viewing file structures and icons for quick access to common applications. Also the windowing system organizes the display by framing the applications and providing window minimization. Although these tools are sufficient to organize the user’s work as testified by the ubiquitous use of the computer, we should expect the desktop to evolve, offering more assistance and more comprehensive tools.

What is the creative process?

We could learn about creative problem solving by reviewing software engineering process such as the waterfall and iterative processes [2]. The iterative process recognizes that solving problems requires multiple attempts and is seldom a linear procedure as suggested by the original waterfall process. Group design processes frequently organize a project by posting design issues and idea notes about the workshop walls. This forum fosters creativity by providing a global view of the project and facilitating associations between previously unrelated ideas. Fortunate juxtaposition of ideas is fertile ground for
initiating new associations between disjoint ideas. I also use other techniques to encourage new associations and promote creativity. Walking through the woods or sleeping on the problem allows the unconscious mind to sift through all the connections, discover associations and sort out the solution. Simultaneously working on more than one task also encourages associative thinking and breaks the blockage of unproductive attempts. My understanding of the creative process is the fluid and occasionally sporadic process of working and associating multiple ideas and tasks.

**How do we solve problems?**

We should observe the actual process of creativity or problem solving; hoping to learn more specifics about the process [3]. In the following two paragraphs I describe two scenarios of writing a large document, using a computer and not using the computer.

**Using the Computer:**

Creating a document using the computer I use a word processor as my primary application window, other users may use a terminal-based text editor. On my desktop panel is an icon to my favorite word processor for quick access. I search for the current document and previous documents using a file manager. Like most computer-users my file structure is both very broad and deep; so although I can always find my documents, sometimes locating them can take significant time. While writing, I open dictionaries and spread sheets. I search for reference material on the web, which generally generates a prolific number of windows. At the same time I have windows open for easy access to email and streaming audio. Quickly my desktop is cluttered with windows piled 6 layers deep! All of these windows are important to my work, and are usually generated in a flurry of creative activity. Reluctantly I must clean up my desktop to keep working. I must decide which windows to close and which to minimize. Typically this culling of windows occurs at the most inopportune moment when I have made a vague association, and am searching for verification. I do compartmentalize my work by locating related windows in the same page or workspace [4]. Separating windows into pages is only convenient for tasks that are truly compartmentalized; for example users frequently locate email windows in a distinct page. But if more than infrequent references are made to other windows, the process of paging becomes awkward and inhibits creativity. Even with tasks separated into different pages the desktop will eventually require cleaning and windows minimized. Once windows are minimized many potential associations are lost. Eventually exhausted by the creative task, I leave my computer for a walk or sleep, and enjoy returning to the computer with the same window configuration on the desktop.

**Without using the computer:**

Not long ago the personal computer existed in the office as just one appliance among many on the desk. Even now I use many resources not on the computer to compose my documents. My physical desktop and adjacent furniture contain frequently used tools such as pens, colored pencils, paper, scissors, stapler etc. I generate notes and locate them on the desk within easy reach and view. As I reference books from my shelves I leave the books open on the desk. As you can imagine the desk becomes cluttered, but
the capacity of my physical desk is much larger than my computer desktop. I automatically stack the books and notes in an orderly fashion; the process is more physical and visual than using the computer. The piles are of related work and the current tasks have smaller, more spread out piles. When I desire a more permanent association of my notes, I can staple them together or bind them into a notebook. Even using this orderly process, my desktop fills to capacity; but the floor is a readily available receptacle for my reference materials. Eventually my desktop expands to my whole office. As my work progresses, my piles migrate from the floor to the desktop and notes from the desk go on the floor. When I return to my office after a break, I instantly recognize the many tasks and their status from the visual clues of the piles. Do not get the wrong impression; I eventually do clean up. Only during the cleaning process do I decide to discard some notes.

**What are the implications from the problem solving scenarios?**

Evaluating the creative process using the computer we ascertain that the desktop provides much basic and essential functionality for managing multiple tasks. Clear and unobtrusive boarders separate applications and the desktops provide primitive mechanisms for switching between applications and directing the keyboard focus. Current comprehensive desktops have expanded the windowing system protocol for application communication, for example CORBA in gnome [5] and DCOP in KDE [6] desktops. These interfaces create strong functional links between tasks by allowing operations from one application in another. Locating related applications in the same page or workspace can create very weak proximity association between tasks. Filing documents in the same directory creates another association between documents. File managers graphically display the file structure, which can resemble the user’s associations between the documents. This hierarchical tree representation of document associations is a computer science imposition on society. The tree file structure is successful because of its close analogy to folders in filing cabinets. Although the tree structure is unambiguous, it is not the most flexible structure and inhibits many potential associations.

The most notable distinction between physical and computer desktops is the instantaneous multiple perspectives of the work offered by the physical desk. The computer desktop provides only a limited number of work views. Because new associations between ideas are fostered by visual juxtaposition, progressively more distant views of the work permit associating more disjoint ideas. In addition the physical desk offers multiple means for grouping and associating tasks ranging from tenuously stacking of notes to progressively permanent stapling and binding notes together. Problem solving is often a process of trial and error. The intermediate levels of associations offered by stapling and binding allows the creator to temporarily organize his thoughts and continue developing them. Because creativity often occurs in a flurry of activity, the mechanics of organizing ideas should be minimal and not hinder the inception of new ideas generated by the associations. Current desktops are successful
when the user’s tasks are sequential and documents can be neatly divided into workspaces or directories.

Even with the many features offered by the computer desktop, the interface is not as flexible as the physical desktop and does not maximize the potential associations between tasks. Desktops provide strong functional associations by embedding documents into each other and weak proximal associations by locating windows in pages. Current desktops do not offer immediate organizational associations that assist navigation through the desktop and positioning of windows on the display. Organizational associations can assist problem solving by freeing the user’s preoccupation with navigation between tasks and the tedious alignment of application windows on the display. These immediate associations, which can be permanent or temporary links between applications or documents, can also encourage creativity by the fortuitous associations perceived while traversing through the links.

Conspicuously absent is a virtual desktop that expands and provides multiple perspectives of the desktop by zooming, saving us from having a 10-foot wide display. Users should not be constrained to locate their work in discrete display size workspaces. Our work constantly expands beyond the borders of the display and current desktops provide at most two viewing levels, a primary working level view and a bird’s eye view of the entire desktop. Working across multiple applications should not be limited to a single magnification. Sometimes working between applications, for example comparing images, benefits from a more distant vantage point intermediate to the bird’s eye view. Humans are visual and can recognize work areas from the visual patterns of the windows, so a distant (zoomed out) perspective of the desktop enables quick navigation through the whole desktop.

The analysis of the creative or problem solving process suggests at least two desktop features that can assist the user in his creation; a structure for associating tasks and a fully functional zooming utility for viewing windows on the desktop. Associating windows assists the creative process by flexibly organizing tasks and providing additional access to windows on the desktop. A fully functional zooming utility provides rapid navigation across the desktop and allows convenient simultaneous viewing of multiple windows at the appropriate resolution. Zooming also implies that pages or workspaces in the desktop are not limited to the display dimensions. These features are especially worthy of note if they are easily implemented.

**And how do we implement these desktop features?**

Linux desktops are built on the X-windowing system [7]. The X-windowing designers tried not to restrict the desktop appearance and functionality and compartmentalized many of these desktop aspects in a client-application, the window manager [8]. The X-windowing system’s server behaves as the conduit of communication between the window manager and other client-applications. The X-windowing designers allowed for increased functionality through extensions. But extensions are problematic when all the components of the windowing system, the X-window server, window manager and client
applications, need to be aware of the extensions. Nevertheless the X-windowing system is partially successful in providing a layer of abstraction to the desktop, and the X-windowing system and window manager are natural targets for adding fundamental functionality to the desktops.

Because the X-windowing system does not provide for window manager interception of all drawing instructions, the X-windowing front end, Xlib, needs to be rewritten in order to provide zooming. The window manager can manage a single root window property, the virtual-size. All the drawing operations are then scaled by the virtual-size. The implementation is only problematic for font scaling, but restricting the fonts to scalable fonts alleviates most of these issues. Viewing levels near the normal magnification are limited by the discrete front sizes, and viewing beyond the resolvable font size naturally blurs characters into lines.

The window manager using a graph data structure can manage most of the functionality required by task associations. The window manager could draw additional buttons on the application’s frame providing access to all the task association commands, such as association transversal or association creation and deletion. Many task associations should be automatically created; for example an application spawning an additional top-level window should be associated with the newly created window. This automatic association necessitates notification to the window manager, which can be implemented by a simple extension to the X-protocol.

Conclusions

These desktop features are greater than superficially apparent. Work is naturally defined by the associated tasks. Because humans are very adept at visual abstractions, users can navigate the desktop and workspace through zoomed perspectives. Users can also navigate through the windows by following associations. While navigating, the windows can be automatically positioned on the screen and hints from the associations can assist in positioning adjacent windows. This efficient use of the display area would result in effectively larger screen area. As computer resources increase, the automatically expanding desktop would eliminate the need for filing many documents in a directory [9]. Users would only need to locate their work on the desktop and all the associated tasks would already be in place. Users would not need to use a file manager to locate their documents and activate all the related applications. New work or unrelated tasks can be initiated in a clean location on the expanding desktop. These new desktop features are not meant to replace current features, rather to supplement them. Users could still clean the desktop using traditional filing techniques, and the associations could be stored with the documents. The user could also organize large sections of the desktops into workspaces titled with huge fronts legible from very distant views, perhaps eliminating all need for traditional directory filing. Task associations, zooming and expanding desktops would enhance the desktop environment and foster creativity.
Notes


[2] Designing is a problem solving process. We review software engineering design processes because the literature is extensive, for example, Sommerville, Software Engineering, Addison-Wesley, ISBN 0-201-39815-X.

[3] In the Human-Computer Interface (HCI) community these observations are called task analysis. Rosson, Mary Beth and Carroll, John M., Usability Engineering, Morgan Kaufmann Publishers, p. 28, 2002, is a modern reference to HCI design.


[9] Eliminating the traditional tree file structure is not unique to me; Jef Raskin in The Humane Interface, Addison-Wesley, p 149, 2000, describes a similar interface, zooming interface paradigm (ZIP), implemented at Apricus for organizing hospital records. Jef Raskin, the creator of the Macintosh computer, is a premiere HCI designer; his book is insightful and original.