Desktop Environments Paper for UNIX Lab 2500

Jacob Isbell
Isbell Media
209 Montsweag Ct
Kingsport, TN 37664
+1-423-408-5559
Jacob@Isbellmedia.com

ABSTRACT

This paper provides a short overview of what a desktop environment, or DE, is the history of how it came to be, and various elements which are involved in modern day desktop environments.

Topics that will be covered include various desktop environments over the years, common applications that are distributed with most desktop environments and also several tools that can assist a system programmer when working inside a desktop environment such as debuggers, profilers, text editors and IDEs.

Categories and Subject Descriptors
H.5.2 [**Information Systems**]: User Interfaces – *graphical user interfaces, screen design, standardization, user interface management systems, windowing systems.*

General Terms
Design

Keywords
GUI, window systems

1. INTRODUCTION

The proceedings are the records of the research done by Jacob Isbell into the world of desktop environments and the history of them, the tools that build them, and the tools they facilitate in modern day computing.

2. HISTORY OF DESKTOP ENVIRONMENTS

2.1 The first desktop environment

The first modern desktop environment is credited to being invented by Xerox. Before this time, there were only a few concepts of interacting with a computer in a graphical, responsive way that did not require the knowledge of keyboard-inputted commands. In those early days of computing, to manage files and documents, users had to know a variety of many times obscure commands in order to accomplish simple tasks. Now with recent GUI technology, people can use a document management tool which can assist the user in organizing her documents and maintaining a personal knowledge base[3]. In 1970 Xerox created the Palo Alto Research Center, otherwise known as PARC [1]. Here, Xerox hired top computer scientists and gave them near complete freedom to work on whatever they desire and in the end of three years, the researchers had completed what is credited as the first desktop environment.

2.2 Why Xerox

It may seem strange that a company known even to this day for their copiers was the pioneer in arguably one of the greatest advancements in computing the world has seen to date. There is a logical reason. In the 1970s when the world was talking about a "paperless office" and pushing for a greener environment, Xerox feared their business of building paper copy machines would soon be burning a hole in the pockets. Instead of sitting idly by and watching their market disappear as Kodak has recently done, Xerox decided not to fight this new computer trend, but rather work with it and use it to their advantage. It was this desire of self-preservation that propelled Xerox into recruiting the best computer scientists in the country to work on keeping them ahead of the market[1].

2.3 The hardware

The new graphical interface that the team at PARC had created was excellent, but Xerox needed hardware to run this new software. To solve this problem, Xerox built Alto. Alto was small enough to fit under a standard desk and touted a screen that was the exact same size and orientation as a normal piece of papers everybody was familiar with already [1]. User interaction with the computer was mainly exhibited with a keyboard and a rather new invention at the time, the mouse. With these combinations of controls, a user could both point and click on the screen and navigate with the keyboard – a feature we all take for granted today.

3. DESKTOP ENVIRONMENTS

3.1 Role of the desktop environment

The role of the desktop environment is to provide an easy template or framework which applications can access and use to display visual information to a user.

A desktop environment provides several benefits to both developers and users. Firstly, it creates a standard set of interface behaviors and appearances. It defines the look and feel for the entire operating system which will create a uniform, standard experience for the user. One application will usually not have a different looking button than another, unless specifically coded as such.

The developers for platforms also benefit from having a standardized desktop environment. The desktop environment handles all of the details that determine how material is drawn to the screen. Without this, developers would have to specifically write code to manipulate pixels on the screen to display different interfaces, buttons, colors and whatever they wanted in their application. This would make graphical software development nearly impractical.

3.2 Popular desktop environments

There are many different desktop environments available now, but a few stand out in particular. The most popular desktop environment is Windows. It is the basis of most people's computing experience. However, Windows is by no means the only desktop environment in the field. There is also the Mac OS X desktop environment, which, while it is similar to the Windows environment in many ways, and vice versa, has major differences which can increase productivity for some. Other popular desktop environments include GNOME and KDE. These last two are extremely popular options for the most commonly used version of Linux and BSD distributions. Some distributions are even offered in both of these environments, Ubuntu being a good example. Ubuntu is the GNOME version of a Linux kernel while Kubuntu is the KDE version of a specific Linux kernel[2]. While all of these options offer the same basic functionality, there are many differences between them making each of them special and better at one task than the other. In some cases though, a desktop environment is just a matter of what the particular user enjoys interacting with from a stylistic and functionality standpoint.

3.3 Desktop environment applications

In most desktop environments, there are a commons set of applications included with the environment to try to jump start the user into a more productive environment. The applications we will discuss today are not present in all desktop environments. However, they are present in the majority of consumer friendly desktop environments that you will see.

Some of the most basic applications that you can find in most desktop environments are internet, text editing and instant messaging applications. Internet applications are applications such as internet web browsers, mail clients and FTP clients. Text editing applications do exactly what they sound like they do – edit text. This text can be plain txt text or can be a file for a C++ program. Either way, most desktop environments ship with at least one text editor. The last application mentioned, while not as common as the other applications in some desktop environments, is an FTP client. FTP, or File Transfer Protocol, is a standard that has been in existence since the first creation of the internet nearly and is the standard form to transfer data files between a client and a server. The most common application of FTP usage would be in a web server to edit the file directory.

3.4 Programming tools

While not as commonly included with most desktop environments as the previously listed applications, developer tools are included with some operating system distributions.

There are a variety of developer tools available on many platforms. In general though, most of the tools fall into a few categories. Some of the most popular developer tools are IDEs, or Integrated Development Environment, text editors and command prompts.

Integrated Development Environments are applications that have all, or close to all, of the tools a programmer would need access to in order to develop a certain type of program, script or application. This generally includes a text editor, compiler, debugger, file manager and in some cases, a visual interface designer.

Though some may think of a text editor as a basic component of any operating system which has little value, it can be of great value to a programmer. Text editors for programming have evolved greatly over the past 20 years and now feature context sensitive code coloring, text prediction, text completion and automatic debugging and warning features.

And lastly there are command prompts. These are the basis of all computing and are usually accessible on any desktop environment or operating system in one form or another. Some operating systems such as Windows call this application "Command Prompt" while other operating systems, such as Mac OS X, call this application "Terminal". Terminal is usually the name used by most Linux and BSD distributions as well. Command prompts are the quickest way to get a lot of work done on a computer. A user can simply type in a command and hit enter which in many cases, takes much less time than navigating around a screen with a cursor. From a developer's perspective, a terminal or command prompt offers not only a whole new world of productivity over a graphical user interface, but it also offers more functionality and features. For example, using scripting, a developer can interact with a multitude of formats and input/output streams to manipulate data and output meaningful files, graphics and statistics. Using a few commands, a developer can also perform mass file searches or operations – things that are not possible to perform in a graphical environment.

4. ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing me to modify templates they had developed.

Also thanks to soon to be Dr. Eric Brown for requiring me to write this paper so I can fully learn to appreciate the aspects of modern day graphical interfaces.

This exercise has also opened my eyes to the vast wealth of information available on the ACM archives. I now know what I will be reading when I retire.

5. REFERENCES

[1] Reimer, Jeremy. 2005. *A History of the GUI.* http://arstechnica.com/features/2005/05/gui/3/

[2] Kubuntu | Friendly Computing. http://www.kubuntu.org/

[3] Schwarzkopf, Eric. 2004. *Enhancing the Interaction with Information Portals.* *http://delivery.acm.org/10.1145/970000/964519/p322-schwarzkopf.pdf?ip=149.149.181.77&acc=ACTIVE SERVICE&CFID=171146480&CFTOKEN=17251502&__acm__=1349752023_be2b04f19a65103e4c4032fafff9ebf3*