The following paper was originally presented at the
Seventh System Administration Conference (LISA ’93)
Monterey, California, November, 1993

Customization in a UNIX Computing Environment

Craig E. Wills, Kirstin Cadwell, & William Marrs
Worcester Polytechnic Institute

Customization in a UNIX Computing Environment

Craig E. Wills, Kirstin Cadwell, & William Marrs – Worcester Polytechnic Institute

ABSTRACT

This work studies the use of customization in a campus UNIX computing environment where a large computing culture exists, but one that has many diverse interests rather than focusing on a cohesive project. We found there to be a gap between the core system users who use customization facilities and periphery users who show an interest in customization, but are unaware of how to pursue it. These results point to the need for better means for connecting novice users with useful customizations.

Introduction

Customization of computing environments is increasingly available to computer users, but its use often does not match its availability. This work looks at the use and sharing of customization in a campus environment. The setting is the campus of Worcester Polytechnic Institute, which is predominately an engineering school with approximately 2500 undergraduate and 1000 graduate students. Most of the students and faculty have computer accounts on the central computing facilities of the campus, which provides access to a UNIX [9] computing environment and many workstations running the X Window System [10].

In this environment there is much collaboration and sharing by students. Electronic communication and bulletin boards are frequently used. Common assignments often find students in communication either via or around computers. Students work on small-team group projects as part of their courses and explicit project requirements. The result is a ‘‘computing culture,’’ which promotes interaction and sharing of information, particularly among students. Part of this interaction involves how users design their computing environment. By the nature of the environment, students can customize aspects of the command interpreter, window environments, and specific applications such as editors and mailers. Each of these customizations is influenced both by individual preferences and the computing culture as a whole.

Previous work on customization has observed that customization of a user's computing environment is not always a solitary task, but is also influenced by the particular computing culture [4,6]. These studies have identified small groups of users within the computing environment who are responsible for translating and communicating customizations among the community as a whole. In [6], this person is termed a ‘‘handyman’’ referring to his or her ability to bridge the gap between workers and computer professionals. Mackay [4] concludes from her work that the design of customizable software should provide:

- the ability to browse through others' useful ideas,
- better mechanisms for sharing customizations,
- methods of finding out which customizations are used and effective, and
- methods of identifying customizations that are ineffective.

Our work explores how users customize their computing environment and how customization techniques can be made more visible to, and therefore used by, more users in a campus environment [12]. This work is motivated by two factors. First, which we emphasize in this paper, to study the use and sharing of customization in a campus setting where a large and diverse computing culture exists. Second, to explore the development of a tool to facilitate sharing and make customization features more accessible and understandable to novice users. Not only should this tool provide for current customizations, but more importantly be extensible so new customizations can easily be added. This tool directly addresses the needs identified by Mackay; serving as a database of useful customizations that can easily be incorporated into the user's environment.

This work is important because it examines a larger and less cohesive computing environment than previous studies. On the order of 3000 users are studied in some depth concerning their use of customization features. 13% of the users had never logged in to their accounts, while 57% had logged in during the last two weeks and 72% in the past two months. Specific customization uses are collected from 224 users and 13 are interviewed in detail. The result of this project is to learn about the culture of customization in a campus UNIX computing environment.

This paper presents the results of our work and its implications. It begins with a look at related work and the computing domain here at WPI. We go on to discuss the study of customization that was done and the results that were obtained. Based on these results, a customization tool, *ctool*, was constructed as an initial attempt to facilitate sharing of customization. We conclude with our experience with it, a discussion of future directions of this work and a summary of our findings.

Related Work

The UNIX system is powerful, but often not a friendly environment for new users. Early on, Norman [7] discussed many faults of the UNIX human interface, but it has still become a common environment in academic and industrial settings, providing many opportunities for customization.

MacLean, et al. [6], emphasize the idea that tailorability of a system requires two pieces: ‘‘a system that can be tailored’’ and ‘‘a culture within which users feel in control of the system and in which tailoring is the norm.’’ They define a ‘‘Tailorability Mountain and its Inhabitants’’ model where the worker lives on the plains, the tinkerer on the foothills and the programmer on the peaks. The handyman is a person who lives on both the peaks and the foothills to bridge the gap between workers and programmers.

Related to this work is a paper by Hesketh that also uses a button-based interface called Perly [3]. His work with the UNIX interface involves the use of ‘‘Perly’’ buttons, on-screen graphical buttons that connect to UNIX command scripts and can be shared between users.

Mackay concentrates on the influence of the community in the customization of individual systems. In [4] and [5] she presents results from interviews of 51 people concerning customization in a project environment. In looking at triggers and barriers to customization she found two of the primary barriers were that the system seemed too hard to modify and a lack of time.

Anothering interesting customization approach is using one application, a *customizer*, to dynamically change the appearance of another [8]. The customizer application provides users a common mechanism to tailor other applications.

Computing Domain

WPI is predominately an engineering school with approximately 2500 undergraduate, 1000 graduate students and 300 faculty. At the undergraduate level there are approximately 200 Computer Science majors with the remaining students in engineering, science, math and management fields. All students, faculty and staff on campus may obtain user accounts on machines of the College Computer Center (CCC). The computer center provides over 3000 accounts. Approximately 75% of the accounts belong to undergraduate students, with the remainder split between graduate students, faculty and staff. Nearly all undergraduates and faculty have accounts on the central computing system.

At the time of the study, the principal CCC machine for computing was an Encore Multimax running the UNIX operating system, with many DEC color workstations, running the UNIX system and the X Window system, also widely used. Other machines exist within specific departments on campus, but most computing needs are supplied by the CCC, particularly for undergraduate students. The large coverage of WPI users by the CCC machines makes it an attractive environment to study the characteristics of a campus computing environment.

As a basis for studying the use and sharing of customization techniques in our campus environment we identified specific applications that allow customization and are widely used. The most visible application to UNIX system users is the command interpreter or shell [1]. The predominate shells used on campus are *csh* and *tcsh*. Applications in the UNIX environments are customized primarily through ‘‘.files’’ (dot files), which are stored in a user’s home directory and normally not seen when a user lists files. Customization of the shell is controlled by the files created. A default copy of each file is placed in a new user’s directory as part of account creation.

The X Window System is run on each of the DEC workstations. No default customization files exist for the window system, but two files can be used to tailor the environment. The .Xdefaults file controls such window aspects as the screen layout, colors, window size, and fonts. This file is automatically created if the user invokes the DECstation session manager to control the X Window environment. The .X11Startup file is executed each time the user logs into a workstation. It is primarily used to initially create windows for running applications such as a clock, load monitors and remote logins.

Electronic mail is an important application used by the majority of users. Many mailers are used with each having their own customization files for setting up features such as mail aliases. The most common mailer is the UNIX *mail* program, which is controlled by entries in a .mailrc file. One other application studied in our work is the editor. The most widely used editor is GNU *emacs* from the Free Software Foundation [2,11]. It is both customizable and extensible, with settings controlled by the file .emacs.

The Study

Given that these applications are widely used and can be customized, our primary interest was how they are being used in a campus environment. To determine student use of customization possibilities, we created two data collection programs. These programs – one voluntary, one run on every account

– collected data on specific aspects of customization. The first collected customizations that would be useful sharing with other users. It also allowed us to find heavy users of customization who could be subsequently interviewed to gain insight into their customization habits.

The second data collection program was run system wide on each user's account and used to obtain an overview of the customization level of the users on the WPI system. From these data we were able to determine percentages of users at specific customization levels. Based on our experience and previous studies we expected to find a low level of customization.

The first program was a shell script and was voluntarily run by 224 users on the system who responded to requests for help on the project. It gathered specific information from shell, X window, *mail*, and *emacs* dot files. The program maintained a file of all user shell aliases and collected a list of all user dot files. It also asked users if they would be willing to participate in a follow-up interview.

The second program was also a shell script, but it was run by the system administrator on each user account on the system. To protect user privacy, nothing was copied from user accounts and all data were stored without identifying the user. The script searched for counts of information such as aliases, whether a user's shell dot files were the defaults and a list of dot files for the user. Data were collected on 3216 users.

The Results

Given the large number of user accounts that were studied, the first task was to gauge the level of account activity. The system-wide data collection program collected the last login time of users revealing that 415 (13%) of the users had never logged in to their accounts. However, 1834 (57%) had logged in during the last two weeks, and half of the remaining 30% had logged in during the previous two months. These figures indicate a high degree of system activity with the number of inactive accounts constituting a minority.

Customization Files

To measure the customization levels of the users, the program analyzed dot files of applications under study. Table 1 shows the level of customization on the dot files for the shell, the UNIX *mail* program, *emacs* and the X Window System. It shows that approximately 45% of the users have not modified at least one of the default shell configuration files. For the other applications, which have no default files in a new user's account, 15-25% of the users had created customization files. However for the .Xdefaults file, half of the users had done so. Later investigation found that the use of color, provided by the DEC session manager and stored in the .Xdefaults file, was often the first level of customization.

The data collection program also gathered information about additional dot files shown in Table 2. The .newsrc file controls what network news a reader is interested in. As the number indicates, this application is also used frequently by our computing community. The .plan file is used to store personal information that can be accessed by other users. The .friends and .enemies files record other users the person is interested in highlighting when using a locally written piece of software to display where users are logged in. The table shows most users use the default DECstation window manager as opposed to alternative window managers such as *twm*.

File	Ave # of Lines	# found
.newsrc	731.2	1309 (40%)
.friends	24.7	532 (17%)
.plan	13.6	489 (15%)
.enemies	2.4	249 (8%)
.twmrc, et al.	211.5	173 (5%)

Table 2: Other Common Dot Files

Shell Customizations

The system data collection program collected many statistics about use of various shell constructs. The shell has its own programming language and use of these constructs indicate a higher level of customization sophistication. The data collection program collected usage for the following constructs, which

File (Default # of Lines)	Ave # of Lines	# Default	# not exist
.login (9)	12.4	1409 (44%)	8 (0%)
.cshrc (6)	12.6	1453 (45%)	17 (1%)
.mailrc	7.9	N/A	2489 (77%)
.emacs	20.0	N/A	2448 (76%)
.Xdefaults	35.6	N/A	1597 (50%)
.X11Startup	5.9	N/A	2668 (83%)

Table 1: Dot Files Analyzed

are used in shell dot files: aliases, set, setenv, stty, echo, comments, if, and back quotes. Rather than look at each of these features in detail we look at the use of back quotes, used in more sophisticated customization, and aliases, which are a more common use of customization.

A string enclosed in back quotes is executed as a command with the resulting output replacing the back quoted string. To use back quotes, a user generally has a higher level of sophistication. We found that 89% of all users have none or one back quoted string in a shell dot file. Since there is one back quoted string in the default files, it is apparent that the use of back quotes is not a common form of customization.

The system data collection program found that 81% of the users had between one and six aliases. The default shell dot files contained two aliases. The maximum number of aliases was 155 with an average number of six.

The data collected from the volunteer users allowed us to identify specific customizations done by users. From the 224 volunteer users we compiled a list of 4174 aliases. As we see the average number of aliases, and the level of customization, was higher for the volunteer users. We were able to automatically identify a number of categories of aliases.

- 114 of the aliases used the *telnet* command, which allows remote logins to other machines. In particular, many of these aliases were for connecting to interactive games available at other sites.
- 397 of the aliases were used for sending messages to other users. These one-line messages are sent directly to a logged in user's terminal screen.
- 61 aliases were for *finger*ing people across the Internet. This command retrieves information about a user.
- 580 aliases were set up to use locally created programs that were not in normally accessible public areas.
- 447 aliases were used to rename UNIX commands into the equivalent DOS command name, such as *del* as an alias for *rm* to delete a file.

Customization Level of Users

To examine the customization level of users, we divided the users into three groups depending on their use of dot files. A new user begins with only the two default files shown in Table 1. Users who have not modified these default files are grouped together as beginning users. The second group of users are those that have one or more of the other application default files shown in Table 1. Because the creation of these files requires a higher level of sophistication, these users are labeled as intermediate. The last group of users are those that have modified their default shell files and have one or more of the other application dot files. These users have the highest level of sophistication.

Figure 1 shows the number of users at each customization level after discarding users who never have logged into the system. This figure shows that 79% of users have modified at least one of their default files and 22% have also created one or more additional application dot files. These data indicate that most users are incorporating some amount of customization into their computing environment.

We were interested in determining whether all of the beginning users were actually at the same level of customization. Having identified colors as an important first step to customization, we examined the colors for beginning users by looking at the .Xdefaults file. This file manages colors and window set-up. It is interesting to note that of the 572 beginning users, only 13% have no .Xdefaults file, while 87% beginning users do have a .Xdefaults file. We used this information to divide the beginning

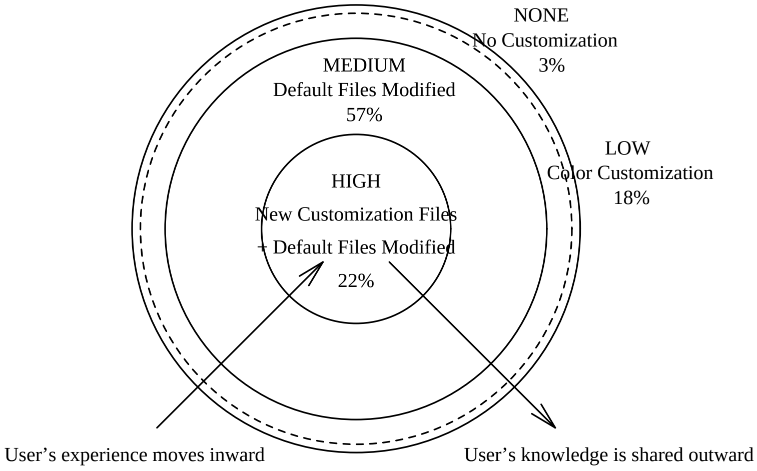

Figure 1: Level of User Sophistication

users into two groups: no and low customization levels. Low users have at least taken the first steps in the process of customization.

From our findings concerning the level of customization on the WPI system, we conclude that almost all users employ some level of customization. Only 3% of the users have done no customization of their computing environment. On the other hand we only have between 15 and 25% of the users with higher levels of customization. Thus, while users appear open to using customization there is a gap between the customization experience of core users shown in Figure 1 and the dissemination of this knowledge to users on the periphery. In addition, the level of customization does not indicate where the customizations came from. For example, some customization comes from invoking programs, such as the session manager or a network news application, that automatically create dot files for the user. Other customization comes from copying customization files of friends. To find answers concerning the origin of customizations users were contacted personally for follow-up interviews.

Interview Results

The volunteer data collection program solicited users for follow-up interviews with a large number of the respondents expressing an interest. To determine which users to interview, a rough measure of the level of customization was computed for each user based on the number and size of dot files, use of constructs such as back quotes and lack of system default files. Thirteen users with a high level of customization were identified. These ‘‘experts’’ were interviewed to determine specific customization techniques they employ that could be shared with others and to determine the triggers and barriers to customization, similar to the study by Mackay.

The interviews consisted of three types of questions: specific information about customizations that experts use; ideas suitable for sharing with other users; and information about the culture of customization at WPI.

Findings About Specific Customizations

Specific information was gathered about shell prompts, mailers, use of the *emacs* editor, what commands the user executes when logging into a terminal or workstation, and the most and least used customizations. In terms of mailers most expert users used UNIX *mail*, but *elm* and *emacs* were also used for handling mail.

Our expert users presented an enormous amount of customization for the personalized making the customizations harder to directly share with other users.

It was interesting to explore the use of the DEC session manager, which allows the user to customize certain aspects of the window environment. Only three of the users currently used the session manager for setting their environment, although seven of the experts admitted using it ‘‘in the beginning.’’ Three of these users specifically mentioned that they used the session manager to set the colors on their workstation display as their first act as new users.

One example of a specific customization that we explored in detail with our expert users was their choice of a shell prompt. The results are shown in Table 3. As shown, there is no clear preference for a specific prompt, but a number of users had a nonsense word for their prompt (for example ‘‘Zug zug’’ or ‘‘woogie’’).

history	3
partial directory	3
full directory	3
user name	2
host name	4
"DEC"	3
"WPI"	2
nonsense word	6

Table 3: Prompt Contents

Findings About Culture

The interviews were particularly useful for finding out about the culture of customization on campus – triggers and barriers to its use and how it is shared among users. Table 4 shows that 12 of our 13 expert users customized their system slowly over a period of time as they reacted to needs or worked in bursts. Only one user said that he did all of his customization in one day. As the interview progressed, the user volunteered that he had copied all of his customization from another of our expert users.

Slowly over time	12
All at once	1
Added as needed	6
In bursts	3

Table 4: Customization Timing

The two most common ways of acquiring customization, according to our expert users, were to personally write their own customization and to modify customization techniques given to them as shown in Table 5. Almost all of the users acquired some number of customizations from other users, which compares favorably to the results obtained by Mackay [4]. However, the proportion of expert users writing their own customizations compares to Mackay’s findings for system programmers in a corporate environment indicating that high customization users are playing a similar role in a campus environment.

From these interviews we were able to determine that time is a barrier for many users. Almost all of the expert users informed us that their customization was created slowly over a period of time, and most of their customizations were personally written. Before a user is able to implement personally written customization techniques, he or she must have an understanding of the technique. Another barrier which many users commented on throughout the interviews was that the man pages (the on-line system manual) were not written for beginners, so they were hard to learn from.

The expert users also identified triggers to the customization process. Many mentioned that workstation colors and bitmap backgrounds made the environment more appealing to work with, which increased the amount of time and energy they put into customizing the system. Many expert users also mentioned hanging around the computer center and absorbing the culture there as a trigger to their customization process.

Ctool

Our findings show that users are open to customization, but we need a better means to disseminate what and how it can be done. Users are willing to share, but often the sharing is limited to a small subset of friends and in the large, unstructured computing environment the sharing often takes the same form. Sharing entire customization files provides useful customizations for the recipient, but can also leave the user bewildered with many customizations that are not understood.

As a result of our findings *ctool* was developed to help beginning and intermediate users become familiar and comfortable with the computer environment. A key design feature of *ctool* was extensibility so that new customization modules can easily be added by more sophisticated users to aid novice users.

The tool allows users to select customizations from a tree-structured menu, which causes specific lines to be added to the appropriate dot file. When the program adds lines to a file, it also adds comments informing the user that these lines were added by the customization tool, and what the lines do. In this way, the user can begin to understand what happens in specific customization files and begin to experiment with the command options available. At this time, the tool does not replace or delete any customizations previously added by itself or the user.

Initial feedback on the tool has encouraged us that such a tool has a role to play in the community. Even though the customizations are not sophisticated they are useful for novice users to feel more comfortable and therefore in more control of the system. Providing users a means to select individual customization features allows them to understand how their environment is being changed. Because of its extensibility, the tool naturally allows others to add customization modules.

Future Directions

Our work suggests a number of directions for future research. Of immediate interest to us is obtaining more measurable feedback on the effect of tools to facilitate customization sharing. Although *ctool* was a good initial attempt at sharing customizations, it is relatively unsophisticated and needs to be upgraded in how it works and its interface. The operations available in the modules and the ability to recognize what the user already has for customizations would increase the power of the tool. The interface could be made more sophisticated than the initial command-line, menu-oriented one.

Another important question is how to measure effectiveness of customization in a computing environment. A place to start would be a study of tool usage over an extended period, although our experience and others' suggests that measuring the feeling of comfort and control is the real goal. Related to usage of the tool is whether real handymen need to also be available to encourage customization. While such people can be used in cohesive project groups, their effectiveness in a large and diverse community such as a campus appears to be less.

An interesting aspect for future work is to examine the customization habits of users over time. Plotting customization level versus time would help in understanding triggers and barriers by identifying times that customization did and did not occur. The use of tools such as *ctool* should break down barriers and increase the level of customization in a shorter time.

	All	Most	Some	Few	None	Then Modified
Personally Wrote	4	2	0	2	0	
Group Wrote	0	0	1	0	0	
Public Programs / Found	2	0	0	0	0	
Given to Them	3	1	1	1	0	4

Table 5: How Customizations Acquired

Such time/customization plots could also be used to examine characteristics of customizable applications or environments. For example, applications that show a low usage of customization capabilities over extended time indicate that customization is not needed or may be too difficult to use. Application plots that show immediate customization and then a leveling out by all users of the application indicate that the default settings for the application may be incorrect, forcing users to immediately customize.

Summary

In this work we examined the use of customization in a campus UNIX computing environment. This study provides information for customization on a large scale as compared to previous studies, which were centered around users on a specific project. As expected we found a variety of customization use in the system, but were interested to note that almost all users, who had logged in, showed evidence of customization. The least sophisticated users had at least used a session manager tool to customize colors on their workstation indicating an openness to customization. In terms of type of customizations, many users had a number of aliases for shortcuts to interaction with each other. The use of customization for aiding interaction is also seen with the number of users having .friends and .enemies files.

On the other hand, most users still showed a relatively low level of customization, indicating a lack of time or understanding for customizing their environment. We did find the existence of a computing culture where handymen disseminate customizations to other users, but in a large environment we find many users who are not well connected. This situation leaves a gap between the core system users who use customization facilities and periphery users who show an interest in customization, but are unaware of how to pursue it.

These results point to the need for better means for connecting novice users with useful customizations. Towards this aim we explored the introduction of a customization tool, which allows novice users to tap into the customization knowledge of others. Initial reaction of users suggests such a tool can be useful, but it needs more sophistication and more customization modules for continued use.

References

[1] S. R. Bourne. UNIX time-sharing system: The UNIX shell. The Bell System Technical Journal, 57(6):1971-1990, July-August 1978.

[2] Heikki Halme and Juha Heinanen. GNU emacs as a dynamically extensible programming environment. Software-Practice and Experience, 18(10):999-1009, October 1988.

[3] Richard Hesketh. Perly-UNIX with buttons. Software-Practice and Experience, 21(11):1165-1187, November 1991.

[4] Wendy E. Mackay. Patterns of sharing customizable software. In CSCW 90 Proceedings, pages 209-221, October 1990.

[5] Wendy E. Mackay. Triggers and barriers to customizing software. In ACM CHI'91 Proceedings, pages 153-160, April 1991.

[6] Allan MacLean, Kathleen Carter, Lennart Lovstrand, and Thomas Moran. User-tailorable systems: Pressing the issues with buttons. In ACM CHI'90 Proceedings, pages 153-160, April 1990.

[7] Donald F. Norman. The trouble with UNIX. Datamation, pages 139-150, November 1981.

[8] John K. Ousterhout. An X11 toolkit based on the Tcl language. In Proceedings of the Winter USENIX Conference, pages 105-115. USENIX Association, January 1991.

[9] D. M. Ritchie and K. Thompson. The UNIX time-sharing system. Communications of the ACM, 17(7):365-375, July 1974.

[10] Robert W. Scheifler and Jim Gettys. The X Window System. ACM Transactions on Graphics, 5(2):79-109, April 1986.

[11] Richard M. Stallman. EMACS: The extensible, customizable self-documenting display editor. SIGPLAN Notices, 16(6):147-156, June 1981.

[12] Craig E. Wills, Kirstin Cadwell, and William Marrs. Sharing customization in a campus computing environment. In HCI International '93, August 1993.

Author Information

Craig Wills has been an assistant professor at WPI since 1990. Previously he had worked at AT&T Bell Labs. His interests are distributed systems, networking and user interfaces. Reach him via U.S. mail at Computer Science Department, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609. Reach him electronically at cew@cs.wpi.edu.

Kirstin Cadwell graduated from WPI in 1992. She works at Digital Equipment Corporation, where she is an SNMP software engineer for Network Access Servers. She can be reached via U.S. mail at Digital Equipment Corporation, 550 King Street LKG1-3/A12, Littleton, MA 01460 or electronically at kirstin.cadwell@lkg.mts.dec.com.

William Marrs graduated from WPI in 1991. He is at Atria Software Inc. where he works on Configuration Management tools. Reach him via U.S. mail at Atria Software Inc., 24 Prime Park Way, Natick, MA 01760. Reach him electronically at bill@atria.com.