1M−9

Cognitive Analysis of a UNIX Network Command

Yasushi Kambayashi
Shonan Junior College

1 Introduction

Shonan Junior College has just started a serious computer science education by using UNIX workstation under computer networks. One of the main problems is that UNIX operations are not very easy for new students to learn. This paper describes an example of cognitive analyses of ftp, UNIX file transfer protocol. The second section covers cognitive walkthrough and the third one covers dialog analysis of ftp. They will show cognitive difficulties of using an ordinary UNIX command user interface and provide base information for constructing graphical user interfaces.

2 Cognitive Walkthrough

When people face a user-interface, they are supposed to form a goal which have some subgoal structures, and to take a sequence of actions to achieve the goal. When he/she cannot form correct subgoal, goal mismatch occurs, and cannot find correct action then he or she cannot get through the subgoal, and consequently cannot achieve the main goal. Therefore if one measures how many examinee can form correct subgoals and how many of them can solve them, one can evaluate the complexity of the given user-interface quantitatively.

The interface which was examined was a normal UNIX user interface. The task was to get a file with unknown suffix from a remote network site. Examinees were freshmen, and they were supposed to know the following things; 1) general knowledge of computer, 2) how to use keyboard, 3) the basic knowledge of MS-DOS, 4) the basic knowledge of UNIX commands,
5) the necessity to read UNIX Programmer's Manual. The correct action sequence is; 1) login system, 2) invoke ftp, 3) open remote site, 4) check file name, 5) get the desired file, 6) close the remote site, 7) exit from ftp. Any knowledge of Computer Network was not assumed.

Since the target group of users had only very basic knowledge of UNIX, and not had the concept of networking, they were not be aware of opening and closing the remote site. The majority of users formed the following initial goal.

```
GET A FILE 'CS560.?'(the suffix is not known.)
           LOGIN SYSTEM
   and then INVOKE FTP
   and then CHECK THE FILE NAME
   and then GET THE DESIRED FILE
   and then EXIT FTP
   and then LOGOUT SYSTEM
```

There was no execution problem nor goal problem so far, and 100% of users could invoke ftp.

When the user tried to check the file name, however, the first problem occurred. Since current goal structure was as above, all users got a serious problem. The problem was that they couldn't do anything. 100% of user got stack in this problem. They had to modify the goal structure by checking the UNIX manual. They realized that they need to open remote site. Backtracking occurred and the new goal structure was formed by inserting a new subgoal OPEN REMOTE SITE after INVOKE FTP. However, about 50% of users gave up their tasks and quited. Some extra knowledge for networking might be required to understand this necessity. In addition, if a user mistyped for remote host name, the system responded as 'unknown host', about 75% of them found the misspelling of the host name and changed it to succeed, but 25% of them got panic and gave up because of not informative response.

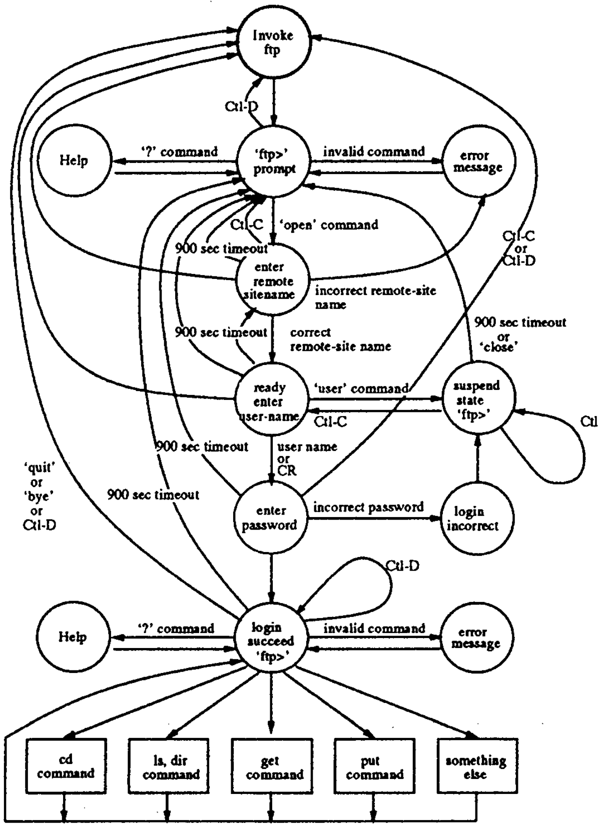
The next subgoal GET THE DESIRED FILE was then modified to GET THE FILE 'cs560.tex' and it was then active. There were two commands to get a file, namely 'get' and 'recv'. They were identical commands and just a synonym each other. There was a problem. To get a file, the user naturally chose command, 'get' or 'receive'. If the user chose 'get', it worked perfectly, but if he chose 'receive', he got response '?Invalid command' from the system. One could find the command name 'recv' without reading manual or get on-line help by typing '?', and guessed 'recv' meant receive. 50% of the users chose 'receive' and a half of them could not figure out it was actually 'recv'. Therefore 25% of users got stuck here. They gave up to proceed.

The next subgoal EXIT FTP was then active. The problem here was, again, the UNIX command name. To get out from ftp, user had to type 'quit' or 'bye'. They had same function. But many users tried to use 'logout' or 'exit' and get the response '?Invalid command' and then got panic.

There was, however, a magic command, Control-D. It worked, even though Control-C did not. Finally 75% of users found the way to exit from ftp. When a user typed in 'quit', 'bye' or Control-D, the ftp prompt disappeared. All the users could sense that they exit from ftp.

3 Dialog Analysis

The ftp system is constructed as a network. It is really task oriented, or task directed system. Since each task has relatively short dialogue, and relatively independent, it may be appropriate to reconstruct one or at most two levels of hierarchy. But the current ftp system does not have any menu to support this kind of structure. The help display can act as main menu, and actually a user can do his/hers task hierarchical. But one cannot say the system is hierarchical with such possibility. Indeed, there is no rigid structure as shown in the figure 1. This diagrams corresponds to opening remote site operation, and the most complex one.



This is rather complicated diagram. Since people were thinking ftp has very simple structure, this complication should be surprising discovery. This simple diagram shows the usefulness of the Dialogue Analysis.

Ftp does not have excellent feedback. When one type 'ftp', just the 'ftp>' prompt appears, and no further information is offered to the user. One has to learn '?' is the help command to find what commands are available in ftp. This help command is always available at the 'ftp>' prompt. When this command is known, the rest is rather easy. Because the necessary arguments can be input in step by step fashiOn with guidance.

Escape and undo are available by using 'Ctl-D' and 'Ctl-C'. At any state, one can undo his/hers work and get back to the 'ftp>' prompt by using 'Ctl-C' command, and one can exit from ftp system by using 'Ctl-D' command. On the other hand, these commands are cryptic and makes the diagrams complex.

In addition to 'Ctl-C' and 'Ctl-D' matter, there is an inconsistency in opening-remote-site operation. Generally, 'Ctl-C' transfer the current state to 'ftp>' prompt state and 'Ctl-D' transfer it to the Unix prompt state, but at the 'enter-password' state, both 'Ctl-C' and 'Ctl-D' make exit from ftp and transfer the state to Unix prompt one. Even though the password checking is very important, I believe 'Ctl-C' had better just transfer the state to 'ftp>' prompt and stay in ftp dialogue for consistency.

4 Conclusion

As mentioned in the second section, a certain number of users may got trouble and gave up their tasks. The result of the cognitive walkthrough can be summarized as follows. 100% of the users could invoke ftp, but 50% of them realized the necessity of opening the remote file and only 75% of them actually opened the remote site. Even though all of those try to check the file name to succeed, 75% of them could get the desired file and 25% of them got trouble to exit from ftp. Therefore, 75% of 75% of 75% of 50% of the all users could complete this task at the first attempt. About only 21% of them succeed!

Although, according to the dialog analysis, the ftp system can be said to be fairy consistent with other Unix commands, reconstruction this network structure into hierarchical one with main menu makes ftp system easier to use for general users.

The main problems on ftp are due to line-by-line conversation mode. The user cannot foresee what is expected in the future, and easily lost where he or she is. Also once the user open the remote site, he/she lost the way to see local files and directory. (Though this problem is solved by using multi-windows as now so common.)

We are now planning to build a new graphic base screen interface for ftp where the entire session is in one screen so that the user can see everything at a glance.

References

[1] P. Polson, C. Lewis, J. Rieman, and C. Wharton. Cognitive walkthroughs: a method for theory-based evaluation of user interface, *International Journal of Man-Machine Studies*, 36, 1992.

[2] Sime and Combs eds. Dialog Engineering, *Designing for Human-Computer Communication*, 1983.