Community Help: Discovering Tools and Locating Experts in a Dynamic Environment

Carlos Maltzahn
Department of Computer Science
University of Colorado at Boulder, Campus Box 430
Boulder, CO 80309-0430
+1-303-492-8709
CarlosM@cs.colorado.edu

KEYWORDS
group work, help systems, group memory, Unix[1], research environments

ABSTRACT[2]
In a research community each researcher knows only a small fraction of the vast number of tools offered in the continually changing environment of local computer networks. Since the on-line or off-line documentation for these tools poorly support people in finding the best tool for a given task, users prefer to ask colleagues. However, finding the right person to ask can be time consuming and asking questions can reveal incompetence. In this paper we present an architecture to a *community sensitive* help system which actively collects information about Unix tools by tapping into accounting information generated by the operating system and by interviewing users that are selected on the basis of collected information. The result is a help system that continually seeks to update itself, that contains information that is entirely based on the community's perspective on tools, and that consequently grows with the community and its dynamic environment.

INTRODUCTION

networked computers in research environments offer a vast number of tools. These are accompanied by many volumes of off-line or on-line manuals. However, if computer users need help they prefer to ask their colleagues instead of using on-line or off-line manuals [3][5]. Traditional help systems (i) give little support for mapping user goals to tools since their access is solely based on keyword search [6], (ii) describe a given tool without domain-specific context [2][8], and (iii) are often outdated because of dynamic environments which acquire new tools on a regular basis (e.g. Unix).

Finding the right person to ask can be time consuming and can interrupt the work of multiple persons. A particular question is only answered to a single person, i.e. another person with the same question has to interrupt again. Furthermore, there is always the risk that a question can reveal incompetence with negative consequences for the person asking the question.

In this paper we report about on-going work on a prototype of a *community sensitive* help system called **ToolBox**. ToolBox actively collects information about Unix tools that are used on a collection of networked Sun 4 workstations at the author's address. Information is collected by tapping into accounting information generated by the operating system and by interviewing users that are selected on the basis of collected information. More specifically, ToolBox collects information via three processes: (i) tool discovery and user selection, (ii) user soliciting, and (iii) interviewing. The goal of the tool discovery and user selection process is to discover new tools and locate people that are most knowledgable about a particular tool. The goal of the soliciting process is to convince users to give an interview, and the goal of the interviewing process is to *exchange* valuable information with the user. The result is a help system that continually seeks to update itself, that contains information that is entirely based on the community's perspective on tools, and that consequently grows with the community and its dynamic environment.

RELATED WORK

The idea of a community sensitive help system is not new. The system described in [4] also tries to integrate users into a help systems for Unix tools. However, the author assumes that users will voluntarily contribute knowledge in absence of any occasion. Furthermore, in order to retrieve information the system requires the user to pose a query. It is often difficult to come up with effective queries without having sufficient knowledge about the domain structure [6].

Answer Garden [1] relieves the user from asking questions by presenting a branching network of diagnostic questions from which the user can choose the best match to his/her problem. If no satisfactory match exists the user can press an "Unhappy" button and post the new question which is automatically routed to the appropriate expert. The system is explicitly configured in order to provide this routing service, i.e., experts are known a priori. In the advent of new

1. Unix is a trademark of Novell, Inc.
2. An extended version of this paper is available as Technical Report CU-CS-747-94 from the author's address.

CHI' Companion 95, Denver, Colorado, USA
© 1995 ACM 0-89791-755-3/95/0005...$3.50

tools, Answer Garden does not support the discovery of new experts. The authors assume formally assigned experts whose job is to maintain Answer Garden.

DYK ("Did You Know") [7] allows any member of a community to add new pieces of information to a directory service with pre-defined categories. The idea is to disseminate information that people consider important and which is hard to find in documentation. DYK randomly presents these pieces of information to users who invoke the tool and select one of the categories. In order to motivate this information exchange the author designed DYK such that users like to use it as some form of procrastination. ToolBox does not rely on randomly collecting and disseminating information. Instead, it interviews selected users who construct and modify an information space according to their community specific understanding. ToolBox provides a very structured and customizable information space that helps users to retrieve information in the situation of a particular problem. Information exchange is motivated by providing occasions for interviews.

TOOLBOX

The architecture of ToolBox consists of a client and a server. The server consists of four processes that share a database: an extended World Wide Web (WWW) server, the Picker process, the Solicitor process, and the Newspaper process.

The *WWW server* (e.g., httpd) is extended by programs that dynamically generate Hypertext Markup Language (HTML) documents. The *Picker* process discovers new tools and locates their users by periodically scanning and analyzing process tables on a collection of hosts. The *Solicitor* process sends email interview invitations to selected users. The goal of the user selection process is to find users that are knowledgable and most likely to give interviews. The goal of the interview invitation process is to make it easy for the user to start an interview and convey a clear and very compact scope of the suggested interview to the user. The *Newspaper* process periodically advertises the ToolBox services to the community by sending out weekly email newsletters which (i) contain statistics about tool usage, (ii) acknowledge people who gave interviews to the ToolBox, (iii) announce newly discovered tools, and (iv) give general information about ToolBox.

The *client* can be any WWW client that supports HTML form processing. Together with the extended WWW server the client serves either as a viewer of the ToolBox information space or as the front end of the *Interviewer* by guiding a user through a set of interactive hypertext documents. These documents allow the user (i) to freely restructure the categorization of tools by moving tools and categories and by adding new categories, (ii) to categorize new tools, and (iii) to add other information that s/he considers relevant. The Interviewer facilitates a goal directed exploration of the ToolBox information space, the goal being the categorization of a small collection of tools. We assume that during this process the user is likely to discover new tools that relevant to his/her work. The argument is: since the user tries to categorize a collection of tools s/he frequently uses, s/he is likely to explore the part of the information space that seems to be most related to this tool collection and, thus, to his/her work. Hence, the ToolBox and the user are *exchanging* information during the interview

STATUS AND FUTURE RESEARCH

So far, we have tested ToolBox for a month within the Department of Computer Science of the University of Colorado. Before the test we seeded ToolBox with 150 Unix tools. During the test, ToolBox monitored seven main hosts and collected information about another 249 tools (the author contributed information about 23 tools). It solicited 235 different users of which 17 successfully. Two of those were related to the ToolBox project so the real success rate is ~6%. Due to feedback from users we believe that this low success rate is caused (i) by the currently very simple user selection strategy of the Solicitor, and (ii) by user interface problems with the Interviewer.

Although every community member could freely modify the information space of ToolBox it stabilized quickly after a few days of intense restructuring and growth: 63 categories were added within the first few days. At the end of the month only six more categories were added. The categories were redundant in only two cases.

Work on improving the ToolBox prototype continues. Users have provided very positive feedback about the ToolBox concept. On-going research consists of (i) finding new ways to improve the user selection strategy of the Solicitor, (ii) improving the interviewing process, (iii) exploring feedback mechanisms in order to distinguish the quality of contributions, and (iv) validating the ToolBox concept by measuring the average number of tools known to a community member before and after a test run.

REFERENCES

1. M.S. Ackerman and T.W. Malone. Answer garden: A tool for growing organizational memory. In *Conference on Office Information Systems*, pages 31–39, Cambridge, MA, 1990. ACM.
2. L. Bannon. Helping users help each other. In D. Norman and S. Draper, editors, *User Centered System Design: New Perspectives on Human-Computer Interaction*, pages 399–410. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.
3. S.R. Hiltz. *Online communities: A case study of the office of the future*. Ablex, Norwood, NJ, 1984.
4. M. Kurisaki. Help systems: An information-sharing approach. In D. Diaper, editor, *INTERACT'90*, pages 529–534. Elsevier Science Publishers B.V. (North Holland), New York, 1990.
5. K.N. Lang, R. Auld, and T. Lang. The goals and methods of computer users. *International Journal of Man-Machine Studies*, 17:375–399, 1982.
6. N. Miyake and D.A. Norman. To ask a question, one must know enough to know what is not known. *Journal of Verbal Learning and Verbal Behavior*, 18:357–364, 1979.
7. D. Owen. Answers first, then questions. In D. Norman and S. Draper, editors, *User Centered System Design: New Perspectives on Human-Computer Interaction*, pages 361–375. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.
8. L.L. Scharer. User training: Less is more. *Datamation*, pages 175–182, July 1983.