Building User and Expert Models by Long-Term Observation of Application Usage

Frank Linton, Deborah Joy, Hans-Peter Schaefer

The MITRE Corporation, Bedford MA, USA

Abstract. We describe a new kind of user model and a new kind of expert model and show how these models can be used to individualize the selection of instructional topics. The new user model is based on observing the individual's behavior in a natural environment over a long period of time, while the new expert model is based on pooling the knowledge of numerous individuals. Individualized instructional topics are selected by comparing an individual's knowledge to the pooled knowledge of her peers.

1 Keywords

OWL, recommender system, logging, organization-wide learning, learning recommendations.

2 Introduction

The goal of this research is to provide individualized instruction based on a new kind of user model and a new kind of expert model. This new user model is based on observing the individual's behavior in a natural environment over a long period of time, while the new expert model is based on pooling the knowledge of numerous individuals. Individualized instructional topics are selected by comparing an individual's knowledge to the pooled knowledge of her peers, which is expected to evolve over time.

This approach is quite distinct from that of other systems, such as Microsoft's Tip Wizard, which recommend new commands to users based on their logical equivalence to the less-efficient way a user may be performing a task, and it is much simpler than the more ambitious Office Assistant which uses Bayesian analysis to understand users' actions and questions and provide intelligent assistance (Horvitz, et.al., 1998). It is also distinct from that of Intelligent Tutoring Systems, which recommend learning actions and activities based on a user's capability to perform specific actions in a defined set of exercises.

Learning often takes place outside formal training or educational classes. Some of this informal learning is accomplished by the exchange of information among people with similar interests. More than ever before, information technology (IT) is the medium of work, and much informal learning in recent years has focused on IT skills.

The purpose of the research reported here is to study informal knowledge acquisition processes and, ultimately, to provide mechanisms that support them. The domain of interest is the use of information technology in the workplace, and the support mechanisms will be based on information technology as well.

As a domain of interest, information technology skills have the advantage of being observable. It is possible to observe text editing skills such as use of the outline tools. In contrast, other workplace skills such as writing skills, e.g., generating an outline, are primarily mental activities, and can only be inferred.

While some familiar workplace technologies, such as e-mail and intranets, support the processes of informal learning, other less familiar technologies such as recommender systems (Resnick and Varian, 1997), which enable the pooling and sharing of information, may be applied to support informal learning as well.

In the first section of this paper we describe the process of logging users' commands; we build models of the users of information technology as they go about their everyday tasks in the workplace. Next, we present an analysis of that data, characterize the users, provide views of the pooled data, and show how contrasting individual user models with the pooled expertise points to learning opportunities. Finally, we examine some of the other uses of this sort of user and expert modeling.

3 The Logging Process

A software application is a computer-based tool; thus details of how the application is used by individuals can be logged. The recent shift from standalone to networked PC computing has resulted in the capability of logging the actions of a large population of individuals performing a variety of tasks with a particular software application for a prolonged period of time. These logged observations can be analyzed and used for designing or refining training programs and for automated coaching. The data can be analyzed and synthesized to build models of current use, and models of expertise. Users can be individually coached by a module that compares individual performance to a synthesized expert's. Finally, the data can be analyzed to observe and promote the evolution of expertise over time.

From a practical standpoint, it is crucial that the logging process be reliable, unobtrusive, and frugal with resources, as observation takes place for extended periods of time (Kay and Thomas, 1995). The research reported here is based on logs of Microsoft Word users. The logger was written in the Word Basic macro language. In general, it is difficult to implement a logger without access to the application's source code, but Cheikes, et.al. (1998) make available a tool for instrumenting UNIX applications without modifying the application.

In our system, each time a user issues a Word command such as Cut or Paste, the command is written to the log, together with a time stamp, and then executed. The logger, called OWL for Organization-Wide Learning, comes up when the user opens Word; it creates a separate log for each file the user edits, and when the user quits Word, it sends the logs to a server where they are periodically loaded into a database for analysis. A toolbar button, Figure 1, labeled 'OWL is ON' (or OFF) informs users of OWL's state and gives them control.

Figure 1. The OWL toolbar button.

Figure 2 displays a sample OWL log. The first five lines record general information: the logger version, the date and time stamp, and the author, followed by the platform, processor, and version of Word. At this point detailed logging begins. Each time the user enters a Word command, the logger adds a line to the log file. Each line contains a time stamp, the command name, and possibly one or more arguments. For example, the line beginning 17:11:34 records these facts: at 5:11:34 p.m. the author used the FileOpen command to open the file entitled "Notes for UM'99." The author then performed some minor editing (copy, paste, etc.), then printed the file. The log does not record text a user enters; this omits some potentially useful information but preserves users' privacy and makes logging more acceptable.

Logging captures a detailed record of users' activities but the record may be sketchy for several reasons. First, an individual may also edit text on other systems without loggers, so some of their activity may not be captured. Second, a macro-based logger besides omitting text, does not capture certain other keyboard actions such as Tab and Return, nor does it capture certain mouse actions such as scrolling, nor does it distinguish *how* commands are entered (by menu, icon, or keyboard command). Finally, permitting user control over logging means that logging can be turned on and off at will, though the default is that OWL is on. To summarize then, the logged data is neither a census of the user's actions, nor a random sample, but rather an arbitrary selection of them.

```
Initiated OWL 4.4 Logging at 11/5/98 17:11:34
System Identfier/Author m300
Platform = Macintosh 8.1
Processor: 68040
Microsoft Word Version 6.0.1
17:11:34 FileOpen Frobnut:Conferences 99:UM'99:Notes for UM'99
17:11:36 Doc size: 4,790
17:12:05 EditCopy
17:12:15 EditPaste
17:12:40 EditClear
17:12:49 EditCut
17:12:55 FormatBold
17:13:12 FilePrint
17:13:34 FileDocClose
17:13:34 Doc size: 4,834
17:13:34 Filename: Notes for UM'99
17:13:34 Path: Frobnut:Conferences 99:UM'99:
```

Figure 2. Sample OWL log.

4 Analysis

This section presents an analysis of log data. Much of the data is publicly accessible (Linton, 1999). First we present summary statistics of the users and their log data. Next we describe the relative frequencies of the different types of commands. We then present a table showing

relative frequencies of each individual command, and give an equation characterizing their sequential relationship. Fourth, we show how the total volume of data logged for an individual influences their apparent level of expertise. We then show that the structure of individual user data is similar to the structure of the pooled data. Finally we show how we find learning opportunities by comparing individual user models to the expert model created by pooling the knowledge of the group.

The analysis presented here is exploratory in nature. The method we have used is Naturalistic Inquiry, which, to paraphrase Patton (1990, p. 40, 41) involves studying real-world situations as they unfold naturally in a non-manipulative, unobtrusive, and non-controlling manner, with openness to whatever emerges and a lack of predetermined constraints on outcomes. The point is to understand naturally occurring phenomena in their naturally occurring states. This data has been acquired from a set of users who were not randomly selected from the population, and the logged data is not a random sample of the users' actions. Therefore, all statistics presented are descriptive (of this data for this group), not predictive. Any generalizations inferred from this data must be treated cautiously until tested further.

4.1 The Subjects (Users)

The project obtained substantive logs from 16 users. The majority of them were members of one department of The MITRE Corporation's Advanced Information Systems Center. MITRE is a federally funded not-for-profit corporation performing research in the public interest. The users consisted of one group leader, ten Artificial Intelligence Engineers at four levels of responsibility, three technical staff, and two support staff. There were eight males and eight females. The users had been employed at MITRE from one to twenty-nine years with a median of eight years. The users worked on four different Apple Macintosh platforms, three versions of the Macintosh Operating System, and three versions of Word 6.0 for the Macintosh. The data presented here was obtained during 1997, the period of logging ranged from 3 to 11 months per person. The project acquired a total of 96 user-months of data.

During the time they were logged, the users -- as a group -- applied 152 of the 642 available Word commands a total of 39,321 times. The average person used 56 (SD = 25) different commands in the period they were logged (the average logging period was 6 months per person); applying 25 different commands 409 (SD = 545) times in an average month.

4.2 Pooled Knowledge: Overall

We now switch focus from the subjects to the commands they used, beginning with an overall, descriptive view. One of the most salient characteristics of the recorded data is the relative inequality in the use of each type of command. For example, as shown in Figure 3, the File commands, i.e., the commands under 'File' in Word's main menu, make up nearly 48% of all commands used, while the Help commands account for only 0.09 % of the commands logged.

Table 1 lists the 20 most frequently occurring Word commands sequenced by their frequency of occurrence, together with the percentage occurrence of each, and the cumulative percent of usage for all commands up to that point in the sequence. Command names are preceded by their main menu type, e.g., FileOpen is the Open command on the File menu. The

first two commands account for 25% of all use, the first 10 commands account for 80%, the first 20 commands account for 90%, etc.

The inequalities in command counts (for example, the log shows more FileOpen commands than FileClose commands) may be accounted for by recalling that there are multiple ways of accomplishing the same effect, that is, a file may be closed by FileClose, by FileQuit, by FileSaveAs, or by crashing the system; this last method is not logged.

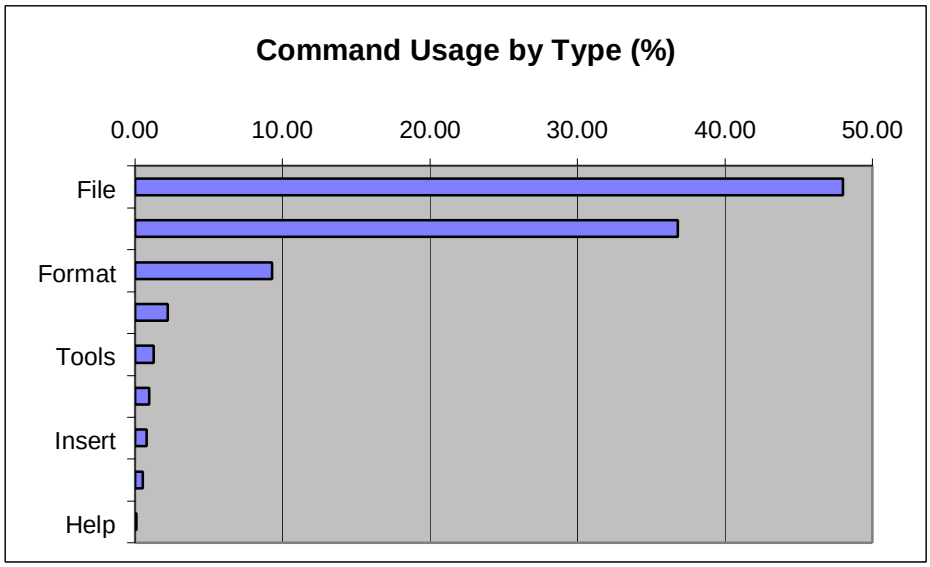

Figure 3. Command usage by type.

The chart in Figure 4 presents command usage data for the 100 most-frequently-used Word commands. The horizontal axis represents commands 1 through 100 (the names of the first 20 of these commands were itemized in Table 1). Each command's usage is indicated by the Percent line relating to the logarithmic scale on the left margin of the chart. Note that command usage (expressed in percent) varies by more than three orders of magnitude. The trendline plotted, which most-closely fits the observed data ($R^2 = 0.96$), is a power curve. An exponential curve also provides a close fit ($R^2 = 0.90$) but it badly mis-estimates the first 12 values (in contrast, the power curve mis-estimates only the first four values). The power curve equation describing Word command frequency of use is

$$y = 137x^{-1.88}$$

This equation is in contrast to the exponential distribution reported by Thomas (1996) for Sam editing commands, and the Zipf distribution reported by others (Thomas, 1996) for the UNIX domain.

The line formed by the light-colored triangles in the chart in Figure 4 plots the cumulative percent of data against the axis on the right margin of the chart. As mentioned, relatively few commands account for the bulk of the commands used.

Table 1. Command sequences and percentages.

Sequence	Command	Percent	Cumulative Percent
1	File Open	13.68	13.68
2	Edit Paste	12.50	26.18
3	File Save	11.03	37.22
4	File DocClose	10.25	47.47
5	Edit Clear	9.50	56.97
6	Edit Copy	7.86	64.83
7	Format Bold	4.22	69.05
8	File Print	4.12	73.16
9	Edit Cut	3.50	76.66
10	File Quit	2.73	79.40
11	File SaveAs	2.17	81.57
12	File PrintDefault	1.23	82.81
13	Edit Undo	1.16	83.97
14	Format Underline	0.94	84.90
15	File New	0.90	85.81
16	Edit Find	0.85	86.66
17	Format CenterPara	0.79	87.45
18	Tools Spelling	0.75	88.19
19	File PrintPreview	0.74	88.94
20	View Header	0.68	89.62

These figures and tables above are based on *all* the collected data. Since there are many short data samples and only a few long ones (Figure 5), and since the frequency of occurrence of commands is a function of the total length of an individual's data sample, it might be expected that frequently-occurring commands are somewhat over-represented, and rarely-occurring commands are somewhat under-represented. However fitting trendlines to selected subsets of data, such as the first 1000 data points of all the logs longer than 1K, and the first 4000 data points of all the logs longer than 4K revealed no systematic changes in the curve.

4.3 Pooled Knowledge: Details

One might hypothesize that it would be adequate to observe an individual for a relatively short period of time to determine their level of expertise, or hypothesize that a graph of an individual's use of distinct commands would rise over time to the user's level of knowledge and then plateau. However, contrary to what one might expect, the number of distinct commands ob-

served is highly correlated (R^2 = 0.83) with the total length of an individual's data sample. In other words, the longer an individual is observed the more knowledgeable she appears to be! The explanation for this phenomenon can be found in the command trendline in Figure 4. The less frequently a command is used in general, the longer an individual must be logged before the command will appear in their record. The chart in Figure 5 plots distinct commands vs the total logged data for each individual in our study.

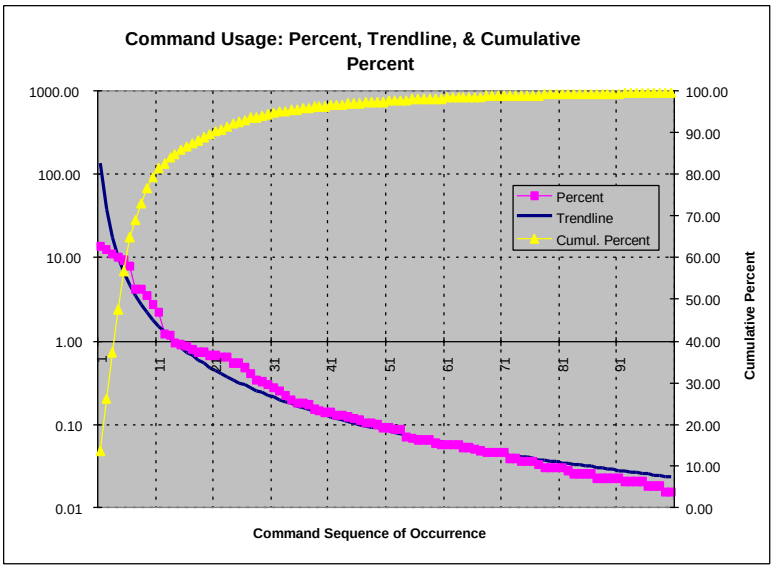

Figure 4. Command trendline.

The apparent differences in knowledge among the individuals that we observed can mostly - but not entirely - be accounted for by differences in the volume of logged data. In the following section we will examine the genuine individual differences and the learning opportunities they present.

The graph in Figure 4 shows the command frequency-of-occurrence trendline, based on all the collected data, as a power curve. We might question whether a power curve also provides the best fit to the data of individual users. Curves were fit to the data from several individual users; indeed, power curves do describe individual as well as group data.

It is tempting to hypothesize a relationship between users' job tasks and their editing tasks, such that certain sets of users would exhibit a preference for certain sets of commands, but in the observed group of users (perhaps too small and too diverse), no such relationship was found. If such a relationship were to be found, the users should then be partitioned or clustered into subgroups which share similar tasks and similar usage patterns, so that recommendations to each user are based on the pooled knowledge of their peers.

Also, one might hypothesize that more-expert users would use a different subset of commands from novices, but they did not; instead they added more commands to their repertoire.

If the hypothesis had been true, individual learning recommendations would have a different character, focusing on mastering a subset of commands at each level.

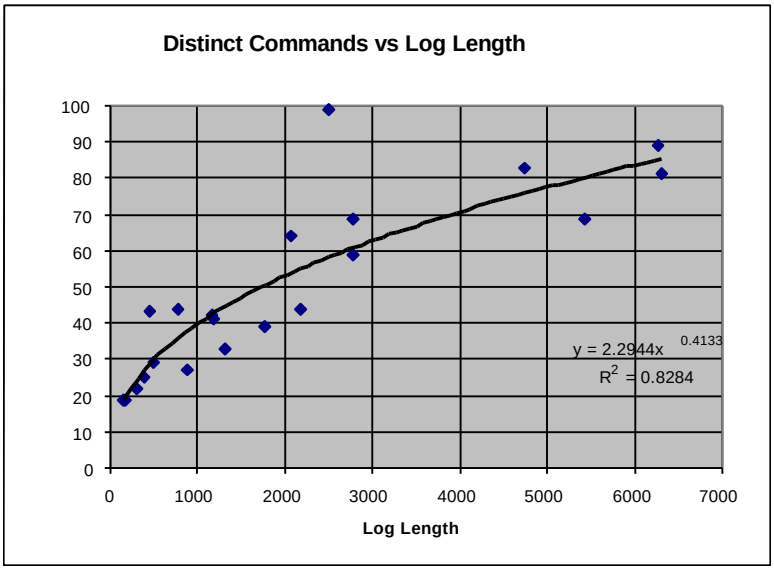

Figure 5. Distinct commands occur as a function of log length.

4.4 Individual Models Point to Learning Opportunities

Our user model is a simple one. It is the list of distinct commands each person has logged, together with their respective frequencies of use. Our expert models are equally simple; they are the commands and frequencies each person would use if her behavior were consistent with that of her peers. By comparing an individual's actual and expert models we can determine whether a particular command is not used, underused, overused, or at the boundary of use.

The pooled data exhibits strong regularities, while individual user models vary not only in the number of distinct commands used, but also in the relative proportions of the commands used. For example, the second most frequently used Edit command, the Delete Forward key, was used by only ten of the sixteen users: four users did not use the command at all and two others used it only once or twice, probably accidentally.

Let us assume for the moment that we have an adequate sample of a user's behavior. In that case, when an individual is seen not to use a command that her peers have found useful, we can assume she might use the command if she were to learn about it. Similarly, underuse of a command may indicate a willingness to learn other ways to apply the command.

Overuse may indicate reliance on a weak general-purpose command, such as Delete, when a more powerful specific command, such as DeleteWord, might be more appropriate.

A given volume of logged data will provide more reliable estimates of the user's knowledge of the more frequently used commands than of the less frequently used ones. For the less

frequently used commands we must do a different sort of analysis. First, the high correlation between volume of logged data and number of distinct commands used (Figure 5) means we must be careful not to equate non-observation of a command with a lack of knowledge of that command by the user. It may be that we have not yet acquired enough data to observe it.

For analyzing the usage of the less frequently occurring commands and making learning recommendations regarding them, we turn to the notion of *confidence interval* (Triola, 1983). The confidence interval is determined by the relative frequency of use of a command and the total volume of logged data. The confidence interval describes the range around the observed value within which the actual value may lie. Thus if a user is observed to use a command zero times, and the confidence interval around the zero value includes the expected value, we cannot conclude that the user does not know the command. For infrequently used commands, the confidence interval is broad. Consequently, estimates of the limits of a user's knowledge must be tempered by the broad confidence interval required by the infrequent use of the commands and the small volume of logged data.

We are interested in determining the boundaries of a user's knowledge because, of all the commands an individual is not using, the commands just beyond the edge of a user's knowledge (in terms of frequency of use) are the most likely to be useful; they represent another learning opportunity.

We assume that the features of an application that are most useful to an individual will evolve over time, not only as her own knowledge and that of her peers grows, but also as their tasks and organizational circumstances change.

These learning opportunities (nonuse, underuse, overuse, and edge of use) can be prioritized and presented to the user in terms of learning recommendations. Learning recommendations determined by pooling the knowledge of a set of peers and by individualizing the instruction (by showing a user what her peers have found useful that she is not yet doing), may result in recommendations that the individual finds particularly helpful in deciding what to learn next, thus reducing learners' efforts while simultaneously increasing their benefits.

Earlier we noted that the high correlation between volume of observed data and number of distinct commands used (Figure 5) means we must be careful not to equate non-observation of a command with a lack of knowledge of that command on the user's part, since we may not have acquired enough data to observe it. Our first observation of a command is not necessarily the individual's first use of the command. We observe *learning* when we observe first use after confidently concluding nonuse.

5 Further Applications of the User and Expert Models

In this section we list some further research questions that can be addressed by this sort of user modeling, describe some practical applications of the models, describe some other activities which could be analyzed fruitfully in this same manner, and summarize the paper.

The user modeling described here can be applied to a number of questions regarding skill development in individuals. How do individuals acquire application software skills? What is their rate of learning? What factors influence learning rate? What factors influence plateauing? What factors distinguish expert users from others?

Researchers in the training arena have their own set of questions. How to encourage low-skilled and average users to become more expert, and experts to continue developing, evolving, and contributing their expertise? What sort of training interventions are the most effective, at the individual, group, and organizational levels? How to build systems that 'automatically' recognize, capture, and instruct new knowledge.

Besides using individual and expert models for individualized coaching and feedback, as described above, the data can be analyzed to improve the content of conventional training (what are users doing that we should be training but aren't), and to improve the methods of conventional training (what have we 'trained' users to do that they aren't actually doing). Data may also be analyzed to improve the application itself, and the data may be combined with network traffic data to analyze the effect of operator actions on network loads.

The model building process described here may be applied to modeling other kinds of user actions besides invoking application commands. For example, an organization's intranet contains a large numbers of information sources. The intranet server logs record who viewed which sources; these can be analyzed to recommend sources to peer groups of users. Lastly, certain programming languages have large numbers of commands or objects; again, an analysis of how they are used, and by whom, could result in an ongoing series of recommendations of certain objects to certain programmers.

To summarize, not only has IT become the medium in which much work is performed, IT skills have become a significant portion of workers' knowledge. In contrast to other tasks, IT tasks are observable, and can be logged and analyzed for several purposes. Here we have focused on analyzing IT usage for the purpose of constructing individual user models based on long term observation of users in their natural environment and on building expert models based on pooling the knowledge of individual users. Finally, we have shown how we might create individualized instruction based on comparing the knowledge of each individual to the pooled knowledge of her peers.

References

Cheikes, B., Geier, M., Hyland, R., Linton, F., Rodi, L., and Schaefer, H. (1998). Embedded Training for Complex Information Systems. In *Proceedings of ITS 98*. Springer-Verlag.

Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. (1998). The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, July 1998*. pages 256-265. Morgan Kaufmann: San Francisco.

Kay, J., and Thomas, R. (1995). Studying long-term system use; computers, end-user training and learning. Communications of the ACM Volume 38 number 7.

Linton, F. (1999). Dataset: Usage of Microsoft Word Commands. A repository of 70,000 rows of data at the Machine Learning for User Modeling web site: http://zeus.gmd.de/ml4um/

Patton, M. (1990). Qualitative evaluation and research methods. 2nd. Ed. London: Sage.

Resnick, P., and Varian, H. (1997). Introduction to Special Section on Recommender Systems. Communications of the ACM Volume 40 number 3.

Thomas, R. (1996). Long term exploration and use of a text editor. Unpublished doctoral dissertation. University of Western Australia.

Triola, M. (1983). Elementary Statistics. 2nd. Ed. Menlo Park CA: Benjamin/Cummings.