

OWL: A Recommender System for IT Skills

Frank Linton

The MITRE Corporation
202 Burlington Rd.
Bedford MA 01730 USA
+1 781 271 2877
linton@mitre.org

ABSTRACT

This research addresses the problem of keeping the skills of information technology (IT) users up to date when both information technology and job tasks are evolving rapidly. The approach we have taken is to use recommender system technology to pass experience from an immediate community of information technology users to the individual who is trying to decide which IT functionality to learn next. User profiles are constructed automatically from implicit input. We compare each user's profile to the pooled knowledge of their peers to identify gaps in their knowledge and to determine the most useful commands at the boundaries of their knowledge.

Keywords

Recommender system, OWL, learning, training, information technology, implicit recommendation

INTRODUCTION

One of the major learning tasks confronting us in the workplace today is to master the information technology (IT) that appears on our desktops. New applications appear, old applications are updated, and the underlying operating system changes from time to time. We need to learn constantly just to maintain competence, but simply maintaining competence is not enough. More and more, work is accomplished by means of information technology. Those individuals and organizations who take advantage of the power of IT have an advantage over those who do not.

What does it mean to master this technology? Learning the minimum functionality is not enough, yet attempting to learn all the functionality is equally misguided, since the majority of an application's functionality usually is not relevant to the job tasks of any one individual.

Mastering the technology, then, means mastering the portion of the technology that is relevant to ones' job tasks.

Yet job tasks are constantly changing (partly as a result of the changes in information technology), new jobs appear, and people change jobs frequently. In short, "the portion of the technology that is relevant to ones' job tasks" is not easily defined. A few minutes of reflection on your own experience will confirm the perspective of sociotechnical systems theory that the consequences of technological

innovation evolve over time, that the changes cannot be predicted in advance, and that, since social and technical systems are reciprocally influential, successful innovation is a function of their joint optimization in a given context.

How can we acquire optimal IT skills when they are evolving, unpredictable, and context dependent? One project that addresses this question is OWL, a recommender system [4] for the Organization-Wide Learning of IT skills. OWL software augments and assists the natural social process of people learning from each other. When individual users ask OWL what software functionality they should learn next, OWL makes recommendations based on the activities of their peers. OWL pools users' knowledge and individualizes instruction. OWL observes people using their applications and logs the functionality they use. By comparing each person's knowledge to the pooled knowledge of their peers, OWL can recommend the most valuable new features to each person, selecting, from the hundreds of features that they don't know, those of proven utility.

While we continue learning after our IT training has ended, our rate of learning is slow because the effort of finding new functionality is high, the likelihood of finding the desired functionality is uncertain, and the future utility of any newly acquired functionality is unknown. OWL's recommendations reduce the cost of finding useful features, and increase the likelihood that the new features one does learn are beneficial.

The approach taken in OWL is applicable in organizations where groups of people perform similar tasks using software on networked computers. OWL has been used to recommend URLs on MITRE's intranet to managers, and to recommend Microsoft Word commands to engineers. In an ongoing study, the average engineer used 57 distinct Word commands, while the pooled group of engineers used nearly 150 commands. OWL's recommendations fill in the gaps and extend the boundaries of their knowledge of Word. OWL recommended selected Word features to each individual, features that their peers had already found useful.

By pooling and sharing knowledge, OWL raises the overall level of knowledge within the organization. By changing the cost/benefit relation of learning, OWL increases every-

one's rate of learning. By observing current IT usage in the workplace, OWL makes learning recommendations that are context-specific, recommends features that have already proven useful, and makes recommendations that reflect the most recent knowledge to be found within the organization [3].

LOGGING

In OWL, each time a user issues a Word command such as Cut or Paste, the command is written to a log, together with a time stamp, and then executed. The logger 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 rows 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 row to the log file. Each row contains a time stamp, the command name, and possibly one or more arguments. For example, the row 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 a user's activities but the record may be sketchy since the logged data is neither a complete census of the user's actions (for example, the user might work on other machines), nor a random sample, but rather an arbitrary selection of them.

ANALYSIS

A portion of summarized user data for the calendar year 1997 is displayed below in Table 1. Kay [2], and Thomas [5] report on related work for college students using a programming editor. While the pooled data exhibits strong regularities, 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, Edit Clear (the Delete Forward key), was used by only ten of the sixteen users logged in 1997: four users did not use the command at all and two others used it only once or twice, probably accidentally.

```

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

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 be told about it. Similarly, underuse of a command may indicate a willingness to learn more 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. There is a high correlation between volume of observed data and number of distinct commands used, thus we must be careful not to equate our non-observation of a command with a lack of knowledge of that command on the user's part. It may simply be that we have not yet acquired enough data to observe it.

These learning opportunities (nonuse, underuse, overuse, and edge of use) can be prioritized and presented to the user in terms of learning recommendations. Table 2 shows a portion of one user's information. 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 useful, thus reducing the effort of finding instruction while simultaneously increasing the benefit of learning.

Seq.	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

Table 1. Command Sequences and Percentages

The first column of Table 2 lists the 10 most-frequently-used Edit commands (Commands nominally under Word's Edit menu), sequenced by their overall frequency of use. Not shown are the vast majority of text editing commands.

The second column of Table 2 lists the expected value for each of these 10 commands. The expected value is the usage the command would have had if the individual had used it in a manner consistent with his/her usage of related commands and consistent with his/her peers' usage of the command. The expected value is computed with a method reported in Howell [1].

The third column of Table 2 lists the actual usage of these commands by this individual during the time s/he was logged.

The expected values are a new kind of *expert model*, one that is unique to each individual and each moment in time. The reason for differences between observed and expected values, between one's actual and expert model, might have several explanations such as the individual's tasks, preferences, experiences, or hardware, but we are most interested when the difference indicates the lack of knowledge or skill.

The fourth column of Table 2 contains various symbols, these are indicators of learning opportunities that an automated tutoring process could make use of. For example,

these indicators are data that, combined with domain and curriculum knowledge, would result in recommendations for learning. The five symbols are: "OK," " " (blank), "New," "More," and "Alt."

COMMAND Type and Name	Expected Usage	Observed Usage	Recom-mendation
Edit Paste	170	274	OK (1)
Edit Delete	129	0	New (2)
Edit Copy	107	97	OK
Edit Cut	48	100	OK
Edit Undo	16	14	OK
Edit Find	12	1	More (3)
Edit SelectAll	9	12	OK
Edit DeleteWord	4	0	New
Edit Replace	3	0	New
Edit PasteSpecial	2	0	New

Notes: 1. OK: The person is using this command as expected.

2. New: The person is not using this command; OWL recommends s/he learn how to use it.

3. More: The person rarely uses this command. OWL recommends s/he learn more ways to use it.

Table 2. Individualized recommendations for User 274.

A command whose expected value is zero need not be learned, and can be ignored; its indicator is a blank (not shown). A command that has an expected value, but is one the individual has never used, is a command the individual probably would find useful if s/he were to learn it; its indicator is "New." A command whose usage is reasonably close to the expected value can also be ignored. The current value of 'reasonably close' was set arbitrarily. Eventually the value can be determined empirically. The indicator for a command reasonably close to the expected value is "OK." A command that is used less than expected may be a component of text editing tasks that are unknown to the user but potentially valuable; its indicator is "More." A command that is used more than expected may indicate ignorance of more powerful ways of accomplishing some text editing tasks; its indicator is "Alt."

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. We focus 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, and finally, we intend to create individualized instruction based on comparing the knowledge of each individual to the pooled knowledge of his or her peers.

REFERENCES

1. Howell, D., (1982). Statistical methods for psychology. Boston: Duxbury Press.
2. Kay, J., Thomas, R. (1995). Studying long-term system use; computers, end-user training and learning. Communications of the ACM Volume 38 number 7.
3. Lave, J., and Wenger, E. (1991). Situated learning: Legitimate peripheral participation. Cambridge: Cambridge University Press.
4. Resnick, P., and Varian, H. (1997). Introduction to Special Section on Recommender Systems. Communications of the ACM Volume 40 number 3.
5. Thomas, R. (1996). Long term exploration and use of a text editor. Unpublished doctoral dissertation. University of Western Australia.