15.4 Predictive Models

Similar to inspection methods and analytics, predictive models evaluate a system without users being present. Rather than involving expert evaluators role-playing users as in inspections, or tracking their behavior as in analytics, predictive models use formulas to derive various measures of user performance. Predictive modeling provides estimates of the efficiency of different systems for various kinds of task. For example, a cell phone designer might choose to use a predictive method because it can enable her to determine accurately which is the optimal layout of keys on a cell phone for allowing common operations to be performed.

A well-known predictive modeling technique is GOMS. This is a generic term used to refer to a family of models that vary in their granularity concerning the aspects of the user's performance they model and make predictions about. These include the time it takes to perform tasks and the most effective strategies to use. The models have been used mainly to predict user performance when comparing different applications and devices. Below we describe two of the most well-known members of the GOMS family: the GOMS model and its daughter, the keystroke level model (KLM).

15.4.1 The GOMS Model

The GOMS model was developed in the early 1980s by Card, Moran, and Newell and is described in a seminal paper (Card *et al*, 1983). It was an attempt to model the knowledge and cognitive processes involved when users interact with systems. The term GOMS is an acronym that stands for goals, operators, methods, and selection rules:

- Goals refer to a particular state the user wants to achieve (e.g. find a website on interaction design).

- Operators refer to the cognitive processes and physical actions that need to be performed in order to attain those goals (e.g. decide on which search engine to use, think up and then enter keywords into the search engine). The difference between a goal and an operator is that a goal is obtained and an operator is executed.
- Methods are learned procedures for accomplishing the goals. They consist of the exact sequence of steps required (e.g. type in keywords in a Google search box and press the search button).
- Selection rules are used to determine which method to select when there is more than one available for a given stage of a task. For example, once keywords have been entered into a search engine entry field, many search engines allow users to press the return key on the keyboard or click the go button using the mouse to progress the search. A selection rule would determine which of these two methods to use in the particular instance.

Below is a detailed example of a GOMS model for deleting a word in a sentence using Microsoft Word.

Goal: delete a word in a sentence
Method for accomplishing goal of deleting a word using menu option:

Step 1. Recall that word to be deleted has to be highlighted
Step 2. Recall that command is 'cut'
Step 3. Recall that command 'cut' is in edit menu
Step 4. Accomplish goal of selecting and executing the 'cut' command
Step 5. Return with goal accomplished

Method for accomplishing goal of deleting a word using delete key:

Step 1. Recall where to position cursor in relation to word to be deleted
Step 2. Recall which key is delete key
Step 3. Press delete key to delete each letter
Step 4. Return with goal accomplished

Operators to use in the above methods:

- Click mouse
- Drag cursor over text
- Select menu
- Move cursor to command
- Press key

Selection rules to decide which method to use:

1. Delete text using mouse and selecting from menu if a large amount of text is to be deleted.
2. Delete text using delete' key if small number of letters are to be deleted.

15.4.2 The Keystroke Level Model (KLM)

The KLM differs from the GOMS model in that it provides numerical predictions of user performance. Tasks can be compared in terms of the time it takes to perform them when using different strategies. The main benefit of making this kind of quantitative predictions is that different features of systems and applications can be easily compared to see which might be the most effective for performing specific kinds of task.

When developing the KLM, Card *et al* (1983) analyzed the findings of many empirical studies of user performance in order to derive a standard set of approximate times for the main kinds of operators used during a task. In so doing, they were able to come up with the average time it takes to carry out common physical actions (e.g. press a key, click a mouse button), together with other aspects of user–computer interaction (e.g. the time it takes to decide what to do and the system response rate). Below are the core times they proposed for these (note how much variability there is in the time it takes to press a key for users with different typing skills).

Operator name	Description	Time (s)
K	Pressing a single key or button Skilled typist (55 wpm) Average typist (40 wpm) User unfamiliar with the keyboard Pressing shift or control key	0.35 (average) 0.22 0.28 1.20 0.08
P	Pointing with a mouse or other device to a target on a display	1.10
P_1	Clicking the mouse or similar device	0.20
H	Homing hands on the keyboard or other device	0.40
D	Draw a line using a mouse	Variable depending on the length of line
M	Mentally prepare to do something, e.g. make a decision	1.35
R(t)	System response time – counted only if it causes the user to wait when carrying out his/her task	*t*

The predicted time it takes to execute a given task is then calculated by describing the sequence of actions involved and then summing together the approximate times that each one will take:

$$T_{\text{execute}} = T_K + T_P + T_H + T_D + T_M + T_R$$

For example, consider how long it would take to insert the word 'not' into the following sentence, using a word-processing program like Microsoft Word:

Running through the streets naked is normal.

So that it becomes:

Running through the streets naked is not normal.

First we need to decide what the user will do. We are assuming that she will have read the sentences beforehand and so start our calculation at the point where she is about to carry out the requested task. To begin she will need to think about what method to select. So, we first note a mental event (M operator). Next she will need to move the cursor into the appropriate point of the sentence. So, we note an H operator (i.e. reach for the mouse). The remaining sequence of operators are then: position the mouse before the word 'normal' (P), click the mouse button (P_1), move hand from mouse over the keyboard ready to type (H), think about which letters to type (M), type the letters n, o, and t (3K), and finally press the spacebar (K).

The times for each of these operators can then be worked out:

Mentally prepare (M)	1.35
Reach for the mouse (H)	0.40
Position mouse before the word 'normal' (P)	1.10
Click mouse (P_1)	0.20
Move hands to home position on keys (H)	0.40
Mentally prepare (M)	1.35
Type 'n' (good typist) (K)	0.22
Type 'o' (K)	0.22
Type 't' (K)	0.22
Type 'space' (K)	0.22
Total predicted time:	5.68 seconds

When there are many components to add up, it is often easier to put together all the same kinds of operator. For example, the above can be rewritten as

$$2(M) + 2(H) + 1(P) + 1(P_1) + 4(K) = 2.70 + 0.80 + 1.10 + 0.2 + 0.88 = 5.68 \text{ seconds.}$$

A duration of over 5 seconds seems a long time for inserting a word into a sentence, especially for a good typist. Having made our calculation it is useful to look back at the various decisions made. For example, we may want to think why we included a mental operator before typing the letters n, o, and t, but not before any of the other physical actions. Was this necessary? Perhaps we don't need to include it. The decision when to include a time for mentally preparing for a physical action is one of the main difficulties with using the keystroke level model. Sometimes it is obvious when to include one, especially if the task requires making a decision, but for other times it can seem quite arbitrary. Another problem is that, just as typing skills vary between individuals, so too do the mental preparation times people spend thinking about what to do. Mental preparation can vary from under 0.5 of a second to well over a minute. Practice at modeling similar kinds of task and comparing the results with actual times taken can help overcome these problems. Ensuring that decisions are applied consistently also helps, e.g. applying the same modeling decisions when comparing two prototypes.

ACTIVITY 15.6

As described in the GOMS model above there are two main ways to delete words from a sentence when using a word processor like Word. These are:

1. Deleting each letter of the word individually by using the delete key.
2. Highlighting the word using the mouse and then deleting the highlighted section in one go.

Which of the two methods is quickest for deleting the word 'not' from the following sentence?

I do not like using the keystroke level model.

Comment

1. Our analysis for method 1 is:

Mentally prepare	M	1.35
Reach for mouse	H	0.40
Move cursor one space after the word 'not'	P	1.10
Click mouse	P_1	0.20
Home in on delete key	H	0.40
Press delete key four times to remove word plus a space (using value for good typist)	4(K)	0.88
Total predicted time = 4.33 seconds		

2. Our analysis for method 2 is:

Mentally prepare	M	1.35
Reach for mouse	H	0.40
Move cursor to just before the word 'not'	P	1.10
Click and hold mouse button down (half a P_1)	P_1	0.10
Drag the mouse across 'not' and one space	P	1.10
Release the mouse button (half a P_1)	P_1	0.10
Home in on delete key	H	0.40
Press delete key (using value for good typist)	K	0.22
Total predicted time = 4.77 seconds		

The result seems counter-intuitive. Why do you think this is? The amount of time required to select the letters to be deleted is longer for the second method than pressing the delete key three times in the first method. If the word had been any longer, for example, 'keystroke,' then the keystroke analysis would have predicted the opposite. There are also other ways of deleting words, such as double clicking on the word to select it and then either pressing the delete key or the combination of Ctrl+X keys. What do you think the keystroke level model would predict for either of these two methods? ■

Case Study 15.1

Using GOMS in the redesign of a phone-based response system

Usability consultant Bill Killam and his colleagues worked with the US Internal Revenue Service (IRS) several years ago to evaluate and redesign the telephone response information system (TRIS). The goal of TRIS was to provide the general public with advice about filling out a tax return – and those of you who have to do this know only too well how complex it is. Although this case study is situated in the USA, such phone-based information systems are widespread across the world.

(*Continued*)

Typically, telephone answering systems can be frustrating to use. Have you been annoyed by the long menus of options such systems provide when you are trying to buy a train ticket or when making an appointment for a technician to fix your phone line? What happens is that you work your way through several different menu systems, selecting an option from the first list of, say, seven options, only to find that now you must choose from another list of five alternatives. Then, having spent several minutes doing this, you discover that you made the wrong choice back in the first menu, so you have to start again. Does this sound familiar? Other problems are that often there are too many options to remember, and none of them seems to be the right one for you.

The usability specialists used the GOMS keystroke level model to predict how well a redesigned user interface compared with the original TRIS interface for supporting users' tasks. In addition they also conducted usability testing. ■

15.4.3 Benefits and Limitations of GOMS

One of the main attractions of the GOMS approach is that it allows comparative analyses to be performed for different interfaces, prototypes, or specifications relatively easily. Since its inception, a number of researchers have used the method, reporting on its success for comparing the efficacy of different computer-based systems.

Since Card *et al* developed GOMS and KLM, many new and different types of product have been developed. Researchers wanting to use the KLM to predict the efficiency of key and button layout on devices have adapted it to meet the needs of these new products. Typically, they considered whether the range of operators was applicable and whether they needed additional ones. They also had to check the times allotted to these operators to make sure that they were appropriate. This involved carrying out laboratory tests with users.

Today, mobile device and phone developers are using the KLM to determine the optimal design for keypads (e.g. see Luo and John, 2005). For example, in order to do a keystroke model analysis to evaluate the design of advanced cell phone interaction, Holleis *et al* (2007) had to create several new operators including a Macro Attention Shift (SMacro) to describe the time it takes users to shift their attention from the screen of an advanced cell phone to a distant object such as a poster or screen in the real world, or vice versa, as indicated in Figure 15.9.

Figure 15.9 Attention shift (*S*) between the cell phone and objects in the real world

From their work these researchers concluded that the KLM could be adapted for use with advanced cell phones and that it was very successful. Like other researchers they also discovered that even expert users vary considerably in the ways that they use these devices and that there is even more variation within the whole user population.

While GOMS can be useful in helping make decisions about the effectiveness of new products, it is not often used for evaluation purposes. Part of the problem is its highly limited scope: it can only really model computer-based tasks that involve a small set of highly routine data-entry type tasks. Furthermore, it is intended to be used only to predict expert performance, and does not allow for errors to be modeled. This makes it much more difficult (and sometimes impossible) to predict how average users will carry out their tasks when using a range of systems, especially those that have been designed to be used in very flexible ways. In most situations, it isn't possible to predict how users will perform. Many unpredictable factors come into play including individual differences among users, fatigue, mental workload, learning effects, and social and organizational factors. For example, most people do not carry out their tasks sequentially but will be constantly multitasking, dealing with interruptions and talking to others.

A challenge with predictive models, therefore, is that they can only make predictions about predictable behavior. Given that most people are unpredictable in the way they behave, it makes it difficult to use them as a way of evaluating how systems will be used in real-world contexts. They can, however, provide useful estimates for comparing the efficiency of different methods of completing tasks, particularly if the tasks are short and clearly defined.