

# Method Engineering: From Data to Model to Practice

*Erik Nilsen,*      *HeeSen Jong,*  
Lewis and Clark College      National University of Singapore

*Judith S. Olson, and Peter G. Polson*  
University of Michigan      University of Colorado  
contact at:  
701 Tappan Street  
Ann Arbor, MI 48109-1234  
jso@csmil.umich.edu

## ABSTRACT

This paper explores the behavior of experts choosing among various methods to accomplish tasks. Given the results showing that methods are not chosen solely on the basis of keystroke efficiency, we recommend a technique to help designers assess whether they should offer multiple methods for some tasks, and if they should, how to make them so that they are chosen appropriately.

**Keywords:** User-interface design issues, design techniques, models of the user.

## INTRODUCTION

In attempting to help users achieve high levels of productivity, designers of application software commonly provide users with multiple methods. These are often specialized methods to deal with frequently occurring special cases of important tasks, such as navigation through a document or spreadsheet, or formatting a document bit by bit or through style templates. For example, in Lotus 1-2-3® there are numerous (more than eight) methods for navigating through spreadsheets. Examples include using arrow keys for moving one cell at a time, page keys for moving a screen at a time, and a GOTO command for jumping to a named cell or range. If there were only one navigation method such as arrow keys, there would be significant loss of productivity especially when working with large spreadsheets. The rich collection of navigation methods provided in spreadsheet programs enables users to select a method appropriate to a particular situation and use this method to efficiently move to the location of their next task.

Unfortunately, such multiple specialized methods have surprisingly large cognitive costs.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

Improvements in productivity have to be balanced against the cognitive costs of learning the specialized methods and selecting the most efficient method to use in a specific context.

This paper presents data showing how real experts of a piece of popular software (Lotus 1-2-3) select from among a set of methods to accomplish a variety of tasks in a variety of situations. From the results, we show that efficiency in performance is not the only criterion on which people choose methods. We follow the discussion of these results with a technique intended for use by designers to help them determine first whether there is an expected net productivity gain from a special method, and second, how to design the method so users will use it optimally--so they will recognize when to use it and recall it quickly.

## EXECUTION COSTS

A specialized method is typically designed to reduce execution time in the particular situation in which it is applicable. The method almost always takes fewer keystrokes, mouse movements, or other user actions. However, there are other time costs: The method has to be recalled successfully. Furthermore, when the method is complicated, the user has to not only recall the command itself but the specialized parameters within it, such as names of ranges or names of locations. And if several methods are retrieved, the appropriate method has to be chosen from the set. The retrieval time and the decision times can be large (approximately one second), nullifying any potential improvements in execution efficiency.

Execution costs can be described by the keystroke level model (Card, Moran, & Newell, 1983; Olson & Olson, 1990). The keystroke model makes the assumption that time to execute a method can be decomposed into a serially executed sequence of operators where the total time to perform the method is the sum of the times for the individual operators. Card et al. (1983) and Olson and

Olson (1990) list a set of perceptual, cognitive, and motor operators that describe performance.

The most common and visible operator is the keystroke, which takes approximately 250 msec. Retrieval of a command or the next step in a method can take from 1000 to 1350 msec. Removing the hand from the home row of the keyboard, acquiring control of the mouse, and selecting a typically sized target can take 1900 msec. In terms of tradeoffs, then, each memory retrieval operations for a well learned command can take from four to six keystroke units of time. The time to select with a mouse is nearly eight keystroke units of time.

Olson and Nilsen (1988) describe in detail how one does the analysis of performance times and how to compare the execution costs of two competing methods.

### THE LARGER COGNITIVE CONTEXT

The designers of modern software packages seem to implicitly or explicitly assume that: users can quickly select one of several methods, and that users will select the most efficient method from the set of choices offered. In general, neither of these assumptions is correct.

#### The time to decide

Specialized methods impose an extra performance burden, possibly reducing total productivity. In order to choose the best method, the user has to recognize the special features of the current situation and recall the appropriate method. Olson and Nilsen (1988) found that users of spreadsheet software who knew two methods for entering formulas were significantly slower than users who knew just one method. This reduction in productivity was due to an increase in planning time, the delay between the completion of the previous task and the start of the new one. The delay was on the order of about 1000 msec or four keystroke units of time. They attributed this increase time to the decision process required to select between the two alternative methods.

#### Do users always choose the most efficient method?

It is also the case that people faced with several methods do not always choose the method that will maximize productivity. Young and MacLean (1988) presented subjects with a task in which they were to enter values in a matrix of cells. The subjects had a choice of either entering them one by one, moving with a mouse-click and return keys to navigate the cells and rows, or to invest in a significant set-up time to allow a mere "enter" key to move the cursor appropriately from cell to

cell and row to row. Their results showed that in this highly structured, repetitive task situation, people persisted in selecting the method that involved the set-up costs, even when the brute-force, cell by cell method provided a significant (15 second) savings in time.

Since this study is somewhat narrow, in that they tested people in a highly contrived situation where subjects performed a single task over and over again, we decided to investigate people's choices in a more realistic setting. We asked skilled users of Lotus 1-2-3 to enter and change various parts of a number of realistic spreadsheets. We sought both to confirm the results from Young and MacLean that people do not choose efficiently and to reveal, if possible, the bases on which users choose. We expected, from casual observation and personal experience, that factors involving ease of retrieval, avoidance of disasters, and maintenance of context in the global work setting would be significant factors in driving the users' choice of methods.

### THE EMPIRICAL STUDY

#### Subjects

Six expert users of Lotus 1-2-3 were selected from among the MBA and Ph.D. students at the Business School at the University of Michigan. They were selected from their responses to a survey of Lotus use given to all MBA students and Ph.D. students. They indicated that they had used Lotus 1-2-3 continuously for four or more years. In an assessment of their skill level as part of another study on the growth of skill over time, these 6 experts showed that they were far faster, more accurate, and more knowledgeable about the range of possible methods in Lotus than even the best of the students who learned Lotus 1-2-3 at the Business School and used it for two years in class work.

#### The Tasks they performed

The subjects both entered and then changed a number of realistic spreadsheets. They *navigated* to locations that were either at an edge, to a named location, or within a cluster of cells; *specified a range* of cells within another task (such as a range inside a SUM or a range to which a formatting command was to be applied); entered *sums* of values of various lengths, *edited* the contents of title cells; *centered* titles; set the *width* of a range of columns, and then *altered* those column widths by a certain amount. For each of these tasks, the subjects knew at least two methods. For example, to specify how wide a column should be, all subjects knew that they could either type in a value or move the right hand boundary of the

column with the arrow keys one by one until it is the desired width.

Expecting that one basis for a decision about what method to choose would be the time it normally takes people to perform these tasks, we calculated the time for each method for a variety of task situations. That is, for example, we determined how long it would take to move various distances using the arrow keys, the edge key, the page key plus arrow keys, etc. And, we calculated the time to add a series of numbers using either repeated "+" operators or the @SUM function. The basis for this prediction was the Keystroke Level Model from Card, Moran, and Newell (1983), with the extended set of parameters from Olson and Olson (1991). The predictions were specific to some important physical details of our situation, including the particular layout of the keyboard, where the arrow keys were and the edge and page keys.

We made two modifications to the standard assignment of cognitive parameters, based on extensive study of experts. Experts do not select menu items one at a time (incurring an M before each one), but rather "chunk" them into a unit, as if spelling a word. Thus, for example, to insert a row, the command /WIR incurs an Mk for the "/", and an Mkkk for the "WIR." Furthermore, the Mental operator used was 300 msec., rather than the standard value of 1350 msec. This value was calculated from the experts' own performance of other tasks outside this study. These experts were obviously very skilled and very fast in their performance of these standard tasks. A more detailed discussion of these expert-novice differences in keystroke level parameters are found in Nilsen, Olson, Jong, and Polson (1991) and Nilsen, Olson, Biolsi, Rueter, and Mutter (1991).

We then selected task situations that were clearly favoring one method, some for the alternate method, and some at the very point where the two methods were predicted to take the same amount of time to perform, the point we call the "cusp."

The subjects performed tasks involving 14 different spreadsheets in each of eight sessions over a 6 week period, such that no two sessions were within 3 days of each other, but that there was at least one session per week. These sessions were spaced so that subjects were unlikely to recall the exact form of a spreadsheet they had seen previously, avoiding the situation in which they behave mechanically rather than with their normal thinking and decision making capabilities.

In the first session, they practiced on a small spreadsheet, demonstrating to us that they knew all the methods that we were examining in the study. They also learned the notation we used to indicate what should be done to each cell or column in the experiment. Then they ran through one full set of 14 spreadsheets with the designated tasks noted on them: Twelve small spreadsheets and two large ones, the large ones being necessary to test various aspects of long-distance navigation. In seven successive sessions they edited 14 spreadsheets, just as on day one. We constructed two full sets of these 14 spreadsheets, and alternated them over sessions. These two sets were formally equivalent in the key tasks that had to be performed, but different in actual titles, values, layouts, and the order in which the tasks were to be performed. In four of the later sessions, in addition to these 14 spreadsheets, the experts did a variety of other experimental tasks geared to acquiring baseline data on mental organization and timing, suited for other studies. For each of the datapoints in the results shown in the following sections, there are from 96 to 144 individual data points per cell.

## Results

The results are described in two sections. First we discuss the cases having to do with navigation around a spreadsheet. Second, we discuss the behavior in the other tasks, like summing numbers and changing column widths. The navigation tasks tell a story of method choice based on a mixture of time-efficiency and other more situational characteristics; the other tasks elaborate on the features people seem to use in making their choice of methods beyond time-efficiency.

**Navigation Tasks** For each task situation (e.g., moving 14 cells away), we plotted the percentage that were accomplished using each of the various applicable methods. Figure 1 shows the cross-over in choice for the task of navigating various distances to an ordinary (not named, not at an edge of a block of cells), using either the arrow keys, the page keys, and other methods including the edge key.<sup>1</sup> At short distances, the arrow method is more time efficient, and clearly preferred; at large distances, the page method is more time efficient, and clearly preferred. The dotted line here is the point at which both methods are predicted to take the same amount of time, the cusp.

Two things are of interest in this figure. At long distances, where it is clearly inefficient to use

<sup>1</sup>To move to the edge of a block of cells, one hits the "end" key and an arrow key in sequence, indicating which way to move, to which edge.

arrow keys, experts are still using arrow keys 20% of the time. The most time-efficient method, the

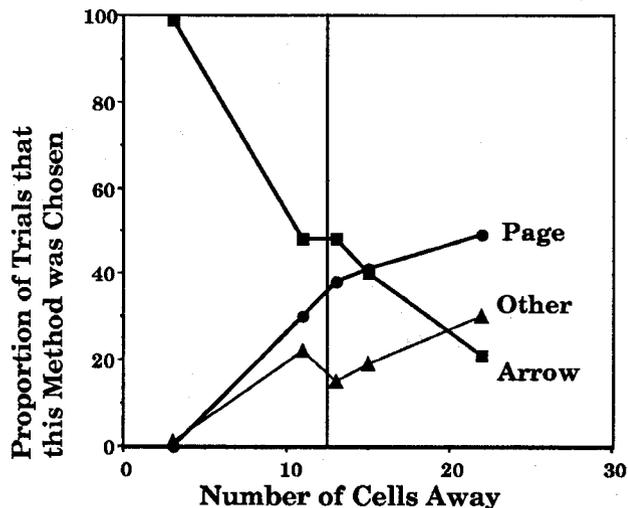


Figure 1. Navigation to a cell that is neither named or on an edge, where appropriate methods are arrow keys, the page key, or other methods, including the edge key.

page key, is used only half the time. The second thing to notice is that the choice of using arrow keys drops below the choice of using the page key very near the cusp, the point that the Keystroke Level Model predicts both methods to be equally fast. At distances below 14 cells, the arrow key should be used exclusively; after 14 cells, the page plus correction arrows is the method that is shortest in execution time. At the longer distances, the edge command is used more often than we predicted, even though it is less time-efficient. We conjecture that the edge method is occasionally preferred to the page command because it moves the view of the spreadsheet to a sensible area (the block or filled spreadsheet area, surrounded by blank columns and rows). The use of these keys preserves the task orientation of the user, rather than merely move a fixed distance.

In summary, the cross-over in choice is at the point at which time-efficiency dictates, but the shape of this function is not optimal. If people were making their choice solely on the basis of time-efficiency, this figure would show 100% choice of arrow before 14 cells away, and 100% choice of page thereafter. The edge key method would be at zero.

Figure 2 shows a second navigation example. Here the people are again navigating around a spreadsheet, but this time to cells that are both named and on an edge. The experts are choosing between using the arrow keys, using the edge key, using the page key plus small arrow corrections, and using the goto-name command. According

to the KLM predictions, the edge key is preferred throughout this range.

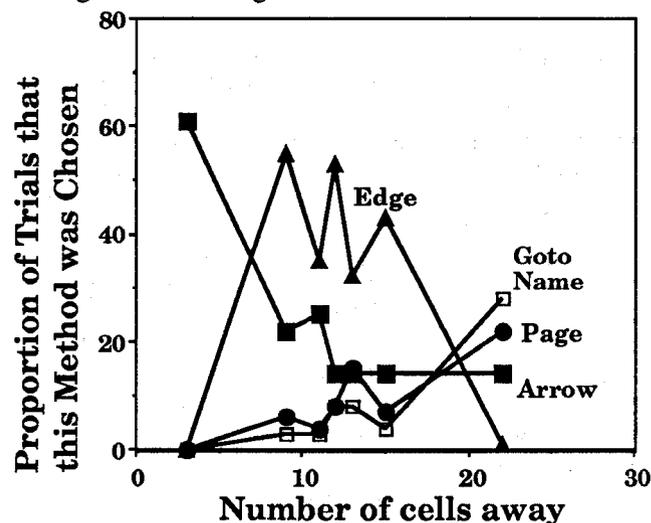


Figure 2. Navigation to a named cell that is also on an edge, where appropriate methods include using the arrow keys, the edge command, and the goto-name command, as well as others.

Two points are striking. At the very short distances, even though the edge key provides the shortest execution time, experts are choosing to use the arrow keys. At the very long distances, the edge key loses to the goto-name command. People recognize that there is something that is needed to move this long distance, and since the cell is named, they choose to use the goto-name command.

It appears in this situation that people are pulled into non optimal methods because of striking situational characteristics. Navigation can always be accomplished by arrow keys and while people work in a local area of a spreadsheet, that is the common method. Because of its high frequency of use, it is retrieved quickly. When the situation calls for *short* navigation to an edge, the arrow key method is chosen, in spite of the actual savings that the edge command can provide. Similarly, looking at a cell that is distant, the experts are choosing a variety of other commands, even though the edge command is once again the most efficient. And, even at the long distances, they are choosing the inefficient arrow method about 15% of the time.

The most interesting phenomenon, however, involves the choices of methods for navigation to an edge when this navigation was *between* tasks or *within* a task, such as specifying a range to format or delete. Although the tasks look different, they are formally equivalent in terms of predicted performance times. For all these conditions, the edge command is the most time-efficient. The

results show that experts persist in using the arrow keys *between* tasks, whereas they are more optimally choosing the edge method *within* tasks. In the post-experiment debriefing, the experts reported that they often chose arrows between tasks because it is "mindless." By doing the mindless method, they can think ahead to the next task and prepare for it. If they move more quickly (using the more time-efficient edge key), they would have to take extra time when they get to the target cell to plan the next task. Thus, the choice of method is not confined to efficiency for each subtask separately, but more sensitive to the total cognitive load and global view of performance efficiency.

**The other tasks.** For the tasks involving summation, altering actual column widths, and formatting, experts chose the method appropriate to execution time-efficiency. But the choice of methods on the other two tasks did not appear to be based solely on time-efficiency. A number of other situational features appeared to drive the choice.

Changing a range of column widths involves a choice between doing it globally or one column at a time. (The early version of Lotus, the one the experts knew, did not allow changing a range of column widths directly). For the task situations we presented them with, it was always more efficient to choose the global command over doing it one at a time. Our experts, however, avoided the global command. As they reported in a post-experiment debriefing session, they did not understand that this command changed only those columns that hadn't been changed explicitly already; they feared it would undo their previous work.

Furthermore, also in the post-experiment debriefing, the experts reported that the only time they used the global command for setting column widths was at the beginning of entering a spreadsheet. Thus, it was a rare choice, and one that was chosen only when the consequences were "safe." This is a good example of "risk control," (Card, et al, 1983), where behavioral choices are determined in part by a combination of the likelihood of making an error and the cost of repairing errors if they occur.

For the task of making editing changes within a cell, the choices facing the subject are to retype the entire title, value, or formula, or after using the edit key (F2) to move to the offending characters, delete them and retype them. There are several specialized ways of moving within a cell: moving with arrow keys, jumping by 5's with the tab or shift-arrow keys, or hitting "home" and

moving with arrows from there. Even though the optimal methods often included these specialized jump key-combinations, the experts almost exclusively chose to move with arrows. Although they knew the jump methods, these methods were unfamiliar enough (and therefore infrequent) making them slow to be retrieved or missed altogether.

### On what bases do people choose methods?

Our results show that people are looking for efficiency, but that efficiency is not exclusively captured in the calculation of the time to perform the method. Several other situational characteristics often drive the choice:

a) Methods that are commonly used are chosen over rare ones that are more time-efficient (as evidenced in the use of the arrow key over the less familiar page method).

b) Methods that exactly fit the perceived characteristics of the particular situation will be selected more often than pure efficiency would predict (as evidenced in the use of the arrow and named-cell methods over edge).

c) Methods that preserve the context of the task are preferred over ones that change the view without regard to meaningfulness (as evidenced in the choice of the edge navigation method over page command).

d) Methods that steer clear of perceived, potential catastrophic consequences are preferred (as evidenced in the avoidance of the global column width command).

e) And, most importantly, methods are chosen to fit the efficiency of the wider context of work rather than optimizing sub-task by sub-task (as evidenced by the choice of arrow navigation instead of the edge command between tasks, a method that frees cognitive resources for planning ahead).

### A MODEL OF THE CHOICE PROCESS

Our model for choice of methods arises as an extension to the simple method selection component of GOMS (Card, Moran, and Newell, 1983). GOMS clearly cannot accommodate the variety of behaviors we have seen here nor their irregularity. In this model, we sought to expand the idea of selection of methods to account for the more situational aspects of the choice and its time course.

Our model of method choice substitutes a retrieval and decision process where GOMS had used a simple selection rule. Methods are retrieved in response to cues about the situation and the goal. If multiple methods are retrieved, the user must decide. This decision can involve additional search in the environment for special features, extra memory search for a special "best fitting" method, or a rough calculation of which of several methods will be most efficient. Also, it is unlikely that users will keep accessing memory or searching the environment or calculating precise tradeoffs without limit; we have assumed that if these processes do not converge in a certain amount of time, either the first method retrieved will be enacted or the default will. More precisely:

**Stage One.** The first stage of method selection involves formulation of the goal and a collection of the significant parameters of the current specific situation. These two sources of information serve as a composite retrieval cue. One or more candidate methods is retrieved. If there is only one, that one is executed. Often the first one retrieved is executed.

Speed of retrieval is heavily dependent on frequency of use *and* the specific fit of the situation to the method. This stage accounts for the emergence of the choice of specialized methods that fit the perceived characteristics, and the frequent methods. We argue also that those methods that preserve the context of the task, those that are easy, and those that avoid risk of error will over time be frequently chosen, and thus quick to be retrieved in this stage. In the long run, frequency of use of methods accounts for much of the quick choice in this stage.

**Stage Two.** If two or more methods are retrieved quickly, however, the model says that a second stage of processing is initiated. In this stage, the user may attempt to elaborate the situation description by searching for more detail, enough to narrow down the choice to only one appropriate method. For example, having retrieved both the page and goto-name method, as appropriate, one searches memory for the name of that cell. If that name is found, the goto-name method is chosen as more fitting.

The second way in which Stage Two resolves the choice from multiple methods is a quick assessment of the efficiency of each method. One way is to refer to rules of thumb (e.g., choose the method with the lesser mental energy required); the other is to retrieve a past situation for which a method was deemed most appropriate.

**Conditions of Time-out.** The processes in this second stage are iterative and may not converge rapidly. A user could go through several passes of attempting to formulate better retrieval cues or making more accurate estimates of efficiency. Since we know that users will not persist in analysis indefinitely, the second stage runs under a global time-out. That is, if a sufficient amount of time has been used in the service of retrieval and search, the user will select either the method first retrieved, the one more often used, or the default, the one that *can* be used universally.

## DESIGNING FOR MULTIPLE METHODS

The technique outlined below is an attempt to make the results of the above empirical studies accessible to designers. We know that the translation from data and models is not easy for designers to make, and thus they don't do it (Bellotti 1990 a and b). What we present here is a method that incorporates the findings from above and the generalizations from the model into a form that designers can use. It steps designers through questions and activities formulated to help them make the right decisions about adding specialized methods. Designers have to be aware of the fact that a user who acquires a specialized method may or may not experience productivity enhancement. The designer who is contemplating the addition of a specialized method should verify that the new method can lead to significant productivity gains.<sup>2</sup> Since the decision process involved in selecting between alternative methods may impose a penalty of over a second or more, at least *four* keystroke units of time must be saved if the method has any hope of providing an advantage.

The first two stages of this method focus on calculating the potential performance gain from a proposed new method. Since a key piece of the model also focuses on the retrieval of a method, the third stage guides the designer to make salient the situation in which a method is appropriate. One could also train users to recognize situations that favor one method over the other.

The following is a detailed step-by-step procedure for carrying out the above analysis. It is reminiscent of the cognitive walkthrough methods of Lewis, Polson, Wharton and Rieman (1990). In it, the designer "walks through" the

<sup>2</sup>There are many methods that we examined early that *never* produce a productivity gain. For example, editing the contents of a cell in order to change whether it is left, center, or right justified is never faster than changing the alignment through use of menu commands.

actual use of the candidate method from the user's perspective examining performance implications to verify if its use results in actual gains in performance.

### Stage I: Considering the set

1. Choose a set of tasks for which alternatives are being considered. Often there are families of tasks that vary on some identifiable dimension (e.g., navigation distance, range of columns that the operation is to effect).
2. List the methods appropriate for each task situation. If appropriate, designate which method is the default method, the one that can cover all possible situations, and which method(s) cover special cases.
3. List the action sequences for each method.

### Stage II: Cost-benefit analysis

1. Determine the time to perform each method using cognitive and action parameters from the KLM analysis, as summarized in Olson and Olson (1990). Include in the calculation the time to decide to use a method as well as the time to perform the method.
2. Determine roughly how frequently each method would be appropriate. It is sufficient to merely judge the frequency as common, intermediate, or rare. The important point is to consider the real task situations in which users will be working and looking at the frequency with which the specialized methods might apply.
3. Multiply the time savings of the method by the frequency of use to determine the overall potential savings. If the savings are small, they likely do not warrant the large extra learning time required for reliably remembering the method. A rare method is often forgotten.

### Stage III: Making the methods retrievable.

1. For each of the analyzed situations, show how the user will retrieve only the most efficient method by satisfying one or more of the following: a) show that there are appropriate visual (and visibly discriminable) clues that trigger the association (e.g., named cells are highlighted), b) show that there are well known features of this situation that can be remembered easily (e.g., recommend short mnemonic cell names for frequently accessed cells), or c) show that there is a clearcut rule that, if taught, helps users determine when each method is appropriate (e.g.,

for navigating, "use page key if the target cell is more than 8 cells away.")

If the most appropriate method is not likely to be the first one retrieved, then the default will be chosen and the special method will be progressively more rarely used. Effort spent on learning the method will be wasted. If this is the case, then the designer is to either drop the method from the offering, or alter the training, visual clues, or mnemonics to make the special features of the situations more easily noticed and their appropriate methods more easily remembered. A method that is never retrieved never provides a performance enhancement. The effort to develop it is wasted.

### CONCLUSIONS.

This paper recommends that part of the interface design process include explicit cost-benefit analysis of the set of methods offered. The user must find significant productivity gains in using a specialized method (far more than 1 second to recoup the time to retrieve it, and even more if the user must choose among several seemingly appropriate ones). And, if there are productivity gains to be had, the designer must insure that the method be retrieved and be the only one retrieved when it is appropriate. Methods must be crafted to be remembered and then more efficient in the global context of work.

**ACKNOWLEDGEMENT** This research was supported by grant MDA 903-89-K-0025 from the Army Research Institute.

### REFERENCES

- Bellotti, V. (1990a) *Applicability of HCI Techniques to Systems Interface Design* Ph.D. Dissertation from the Queen Mary and Westfield College, University of London .
- Bellotti, V. (1990b) "A framework for assessing applicability of HCI techniques" in D. Diaper et al (eds.). *Interact'90 Proceedings*, Elsevier Science Publishers B. V. (North-Holland) IFIP 1990. pp 213 - 218.
- Card, S. K., Moran, T. P., and Newell, A. (1983) *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Assoc.

Lewis, C.H., Polson, P.G., Wharton, C. and Rieman, J. (1990) Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. *Proceedings of CHI'90 Conference on Human Factors in Computer Systems*. New York: Association for Computing Machinery. pp. 235-243.

Nilsen, E., Jong, H., Olson, J. S., and Polson, P. G. (1991) How experts choose among methods to perform tasks: Keystroke efficiency is not the sole basis of choice. Technical report, Cognitive Science and Machine Intelligence Laboratory, The University of Michigan.

Nilsen, E., Olson, J. S., Biolsi, K., Rueter, H. R., and Mutter, S. (1991) The growth of skill in learning and using software. Technical report, Cognitive Science and Machine Intelligence Laboratory, The University of Michigan.

Olson, J.S. and Nilsen, E. (1988). Analysis of the cognition involved in spreadsheet software interaction. *Human Computer Interaction*, 3, 309-349.

Olson, J.S. and Olson, G.M. (1990) The growth of cognitive modeling in human computer interaction since GOMS. *Human Computer Interaction*, 5, 221-265

Young, R. M., and MacLean, A. (1988) Choosing between methods: Analyzing the user's decision space in terms of schemas and linear models. *Proceedings of CHI'88, Human Factors in Computing Systems*, NY: ACM, 201-206.