Check for updates

SKILLED FINANCIAL PLANNING: THE COST OF TRANSLATING IDEAS INTO ACTION

F. Javier Lerch
Carnegie Mellon University

Marilyn M. Mantei
University of Toronto

Judith R. Olson
The University of Michigan

ABSTRACT

We use GOMS models to predict error rates and mental times for translating financial concepts into equations in two widely used interface representations. The first of these, common to spreadsheet packages, is characterized by non-mnemonic naming and absolute referencing of variables. The second, common to non-procedural command-driven software, is characterized by mnemonic naming conventions and relative referencing of variables. These predictions were tested in an experiment using experienced financial analysts. Although the interface that allows mnemonic and relative names (called keyword) takes longer overall, it produces seventy-five percent fewer simple errors and requires less mental effort. Given the overall serious cost of errors in financial models, we conclude that interfaces having the keyword representation are far superior.

KEYWORDS: GOMS models, skilled financial planning, error analysis.

INTRODUCTION

A recurring prescription in user interface design states that the interface should use terms and show a display that fit the objects and actions in the user's mental representation. Translating thought into action takes time and produces errors. If the interface requires a significant translation of the user's mental representation, we can expect an increase in mental time and errors.

Card, Moran and Newell analyzed how skilled users of text editing software translate the tasks to be performed into actions on the system. They did a similar analysis for the routine components of computer-aided circuit design [1]. Their research demonstrated that users break down tasks into small units of approximately 10-30 seconds. It also showed that users' performance in text editing and in the routine components of circuit design were similar both in terms of error frequency and command execution time.

Since financial modelling systems (e.g. spreadsheet packages, business modeling systems, financial modelers, etc.) are the second most utilized end-user software packages in business after word processors and are claimed to be the most frequently used among senior executives [10], an analysis of the cost of translating financial concepts into different kinds of interface representations is warranted. This research shows how interface features affect the performance of skilled financial model builders. Performance is measured in terms of execution times and the frequency of both simple and conceptual errors.

The research was executed in four steps:

1. Task Analysis. The task of building financial models was characterized by breaking it down into its essential operations independent of any given package. This characterization made evident the importance of describing cells referred to in formulas into different types. These cell types are used in different combinations inside common financial formulas in order to represent special time-relationships among generic financial variables.

2. Analysis of the Interface Representations. Two common interface designs (IFPS and Lotus 1-2-3) were analyzed for differences in how one refers to cells in formulas. The key differences are the mnemonic constraints of the cell naming conventions and whether the cells are referred to by their relative or absolute locations.

3. Cognitive Modeling. GOMS models [1] were built to represent the steps that skilled financial model builders are likely to follow when writing model formulas. These cognitive models served as the basis for making predictions on the relative frequency of errors among cell types in different representations. They were also used for predicting and explaining the amount of mental time needed to translate financial concepts into formulas.

4. Performance Evaluation. We conducted an experiment with 31 experienced users of financial modeling systems and compared their performance with our predictions.

TASK ANALYSIS

The basic structure of financial modeling systems is the two-dimensional matrix[1]. Model builders typically use one dimension (usually columns) to represent time periods and the other (rows) to represent financial variables such as categories of costs and revenues. The process of building these models is basically the same for all packages:

1. Rows (variables) and columns (time periods) are labeled. This produces a grid of cells.

[1]Some packages allow adding one or more dimensions in order to include different business units, products, sales regions, etc.

2. The content of each cell in the grid is specified. A cell may contain either numerical values or formulas. Formulas make reference to the content of other cells and include mathematical and/or logical operators and numerical constants.

Most formulas make reference only to cells in the same column where the formula resides. For example, the formulas to represent the relationships shown below only require cell references in the same column:

$\text{Assets}_t = \text{Current Assets}_t + \text{Fixed Assets}_t$
$\text{Profit}_{[t-1,t]} = \text{Revenues}_{[t-1,t]} - \text{Costs}_{[t-1,t]}$

Notice that the two relationships represent different concepts in time: the first equation shows a relationship among variables at a given time t (these are called level variables) while the second shows a relationship between events within a period of time (called flow variables). When financial relationships mix levels and flows, the formulas that represent these relationships are likely to include cells that are not in the same column.

Another important distinction for the financial planner is between income flows and cash flows. Income flows are the result of using the "accrual basis of accounting". In this view, revenues are recognized when the transaction related to the earning process occurs, that is when the sale is made or the service is rendered. An attempt is also made to match expenses with associated revenues. This method produces a good estimate of the profitability of the firm during a period of time. It does not, however, provide information about cash flows because income flows such as revenues, occur before the receipt of cash. The timing of cash flows is crucial for decisions regarding the allocation and acquisition of funds. Therefore, since income flows are important for assessing profitability and cash flows are indispensable for performing sound financial planning, both are included in financial models [2] [11]. They are often confused by the beginning analyst and present a mental burden for the experienced one. An example with timing differences between income flows and cash flows is:

$\text{Receipts}_{[t-1,t]} = .2{*}\text{Revenues}_{[t-3,t-2]} + .8{*}\text{Revenues}_{[t-2,t-1]}$

The formula to represent this equation has to refer to cells in a different column and in a different row with respect to the cell where the equation resides. Every time there is a time lag between income flows and cash flows, the formulas have to include cells in different rows and columns. These are difficult to think about, yet they represent the basic task of financial planning. They are even more difficult to specify in current software.

This research shows that the degree of difficulty a planner has in referencing cells in a formula is related to the kinds of cells: those in the same row and column, and those in different rows and columns. We refer to these types as type SR, type SC, and type DRC. The research also shows that how these cells are made available to the planner by the software package interface can add to the user's task difficulty.

INTERFACE REPRESENTATIONS

References to the content of other cells (cell references) can be specified and represented using different cell-naming conventions. There are two main differences in the cell naming conventions of financial modeling systems:

Mnemonic names vs Non-mnemonic names
Relative references vs Absolute references

Mnemonic cell naming uses the name of the variables such as REVENUES or COSTS to specify rows. Non-mnemonic cell naming only uses the row numbers. An absolute reference states the position of the cell by using its overall coordinates. A relative reference specifies the position of the cell by describing its location with respect to the cell where the formula resides. Four cell-naming conventions result from combining these two cell naming dimensions. Figure 1 shows examples of each.

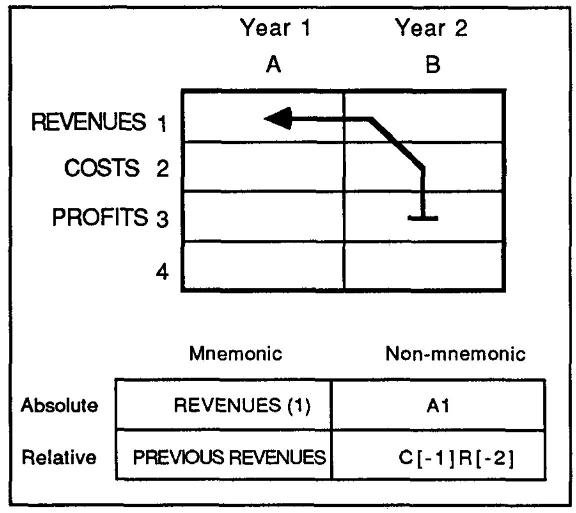


Figure 1. Cell naming conventions for referring to the cell Revenues in Year 1 (at the arrow head) when residing in Profit, Year 2 (the arrow foot)

This research investigates only two of these four naming conventions, the two most widely used in financial modeling systems. IFPS (Interactive Financial Planning System) embodies the relative, mnemonic cell naming convention, here called "keyword." Lotus 1-2-3 uses absolute non-mnemonic cell naming, here called "positional." These two packages exhibit other differences such as "style of interaction" (e.g. command language vs direct manipulation [9]), but it is only the users' performance with respect to cell referencing that is studied here.

COGNITIVE MODELS

The next step of this work involved building GOMS models to simulate skilled performance. The models were built by observing experienced financial analysts in a pilot study. In this study, the participants were asked to think aloud while building typical financial planning models.

During the pilot study model building behavior occurred in two general phases: the acquisition of the appropriate information and the translation of this information into formulas using one of the two interface representations. Skilled financial analysts exhibited the same type of problem solving behavior during the acquisition of information. This behavior is independent of interface representation because the acquisition process occurs before the representation is utilized. Problems in the acquisition phase will be referred to as conceptual errors in the performance evaluation section. In the translation phase the mental effort (cost) of translating financial concepts into formulas should be affected by the different interface representations. The GOMS models drawn from the pilot study specify different sequences of mental operations for the two representations studied here. An example of how the same task is executed with each representation is shown in Figure 2 (the full-fledged GOMS models are discussed in [4]).

GOAL: WRITE-OPERATOR-CELL

Keyword Naming

```
GOAL: WRITE-OPERATOR-CELL
   GOAL: ACQUIRE-INFORMATION-ABOUT-CELL
      SPECIFY INFORMATION-POINTER
      SPECIFY-OPERATOR
   WRITE-OPERATOR                              ..if required

   SPECIFY-VARIABLE-NAME
   GOAL: WRITE-LAG                             ..if lag
      SPECIFY-LAG-POINTER
      SPECIFY-KEYWORD
      WRITE-LAG-KEYWORD
   WRITE-VARIABLE-NAME
```

Positional Naming

```
GOAL: WRITE-OPERATOR-CELL
   GOAL: ACQUIRE-INFORMATION-ABOUT-CELL
      SPECIFY INFORMATION-POINTER
      SPECIFY OPERATOR
   WRITE-OPERATOR                              ..if required

   SPECIFY-VARIABLE-NAME
   GOAL: WRITE-COLUMN
      [Sel  GOAL: GET-COLUMN-NAME              ..if no lag
            SPECIFY COLUMN-NAME
            WRITE-COLUMN-NAME
            GOAL: CALCULATE-COLUMN-NAME        ..if lag
            SPECIFY-LAG-POINTER
            CALCULATE-COLUMN-NAME
            WRITE-COLUMN-NAME  ]
   SPECIFY-ROW-NUMBER
   WRITE-ROW-NUMBER
```

Figure 2. GOAL: WRITE-OPERATOR-CELL

The execution of this GOMS model translates information about the internal representation of the financial problem into cell types SC and DRC. Figure 2 shows that the positional representation requires more steps than the keyword representation because the model builder has to perform extra steps in order to translate variable names into cell coordinates.

Performance predictions were made of the translation process from the GOMS models: 1) Mental times for executing error-free equations were estimated by simply counting the number of mental operations required for each cell reference, and 2) The frequencies of simple errors caused by overload in working memory were expected to correlate with peak working memory loads (called overload errors). These predictions assume that as the peak number of working memory resources utilized gets closer to its limit, the likelihood of error behavior increases [1] [12]. Therefore, error hypotheses can be built by comparing the peak working memory loads of different cell types in each representation. Figure 3 shows the working memory loads for cells type SC and DRC with the keyword representation when executing the GOMS model of Figure 2.

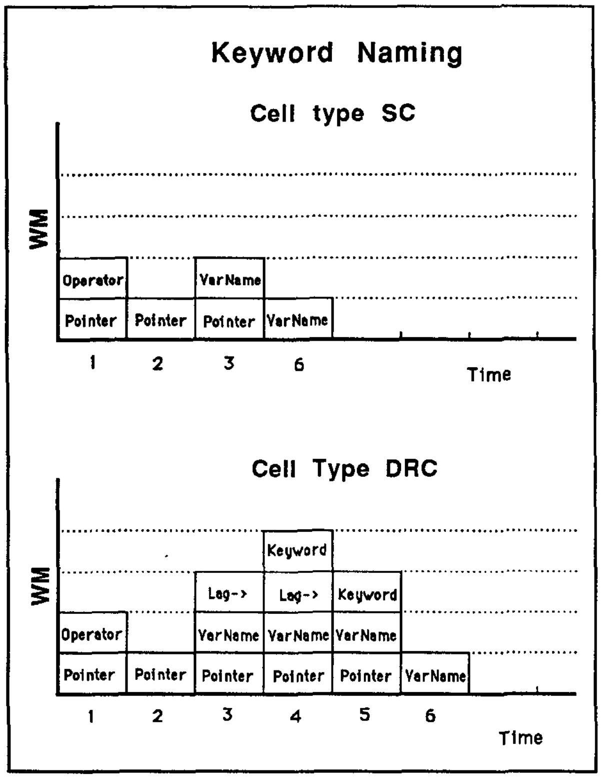


Figure 3. Working Memory Loads for Cell Types SC and DRC for the Keyword Representation

The horizontal axis in both charts represents time, but each interval does not represent the same length of time. The intervals were chosen only to represent when the content of working memory changes. The vertical axis shows a rough estimation of the number of pieces of information kept in working memory for translating concepts into formulas. It does not show symbols for other goals that are also kept in working memory because they are the same for both cell types. Figure 3 shows that cells of type DRC have greater demand for working memory than cells of type SC. We expect higher overload error rates for DRC than for SC in the keyword representation because the model builder has to remember and translate the concept of time-lag into the formula when writing cells of type DRC. The predictions made with the GOMS models for overload error rates are shown in Figure 4.

Similar predictions were made for error-free mental times (see [4] for details). A detailed task analysis is made for each cell type and each interface. The number of steps required is expected to correlate with the time spent performing the task.

Overload Error Rates Predictions

Positional Naming	Keyword Naming
Type SR < Type SC	Type SR < Type SC
Type SR < Type DRC •	Type SR < Type DRC •
Type SR < Type DRC •	Type SC < Type DRC •

Both representations

Positional Type SR > Keyword Type SR •
Positional Type SC > Keyword Type SC •
Positional Type DRC = Keyword Type DRC •

• Supported by empirical results

Figure 4. Overload Error Rates Predictions

In sum, it was expected that subjects using the keyword interface would make fewer overload errors and need less mental time than subjects using the positional interface because of the additional translation effort required with the positional representation. In contrast execution time for the keyword subjects was expected to be longer because subjects are required to write more characters. Similar mental effort estimations were found by Olson and Nilsen [7] in their study of cell referencing in positional style spreadsheet packages. They showed that substantially less mental effort was required for pointing to the cell as opposed to calculating the coordinates. This research expands the essence of these results by making both time and error predictions based on the estimation of the mental effort involved and by incorporating the characteristics of the task of building financial models into these estimations.

PERFORMANCE EVALUATION

The core of the comparison is the performance of users on a *keyword*-based interface versus users on a *positional*-based interface. Keyword subjects were experienced IFPS users; positional subjects were experienced Lotus 1-2-3 users. Each subject was asked to write equations using different cell type combinations. The length of the equations was restricted to equations containing only two or three cell references.

Subjects. Thirty-one experienced users from eleven firms served as volunteer subjects. The subjects work full-time in Fortune 500 financial and industrial firms in Michigan and Ohio. Participating organizations were asked for individuals with the following experience: a) at least two years working in accounting or finance in their organizations, and b) at least one year using either Lotus 1-2-3 or IFPS. Twenty-one subjects were Lotus users (positional); ten were keyword users (IFPS). Lotus users had an average experience of 7.3 years in accounting or finance whereas IFPS users had 6.3. All subjects had more than one year experience with their respective package. Ninety-four percent of the subjects reported using the package at least twice a week, with the remainder using it less often because they had recently been promoted to supervisory positions.

Practice and Test Problems. The subjects were asked to solve the problem with pencil and paper. This eliminated the differences in interactive style presented by the two packages. Subjects read a description of the problem aloud from an instruction booklet and were then given a summary of data and a model skeleton. The summary of data detailed all the input data for building the model. The model skeleton was a worksheet with labeled columns and rows that had empty rows in which the participants wrote the missing equations or data. There were five practice problems and six test problems with 24 empty rows.

Procedure. The experimental sessions were one hour and a half long and consisted of: oral instructions, practice problems, test problems, two tests of writing speed and a two-part questionnaire. The whole experimental session was videotaped. During the test and practice problems, subjects read a description of the problem and the objective for building the financial model. They then picked up the model skeleton and the summary of data. They were asked to write one variable at a time and to avoid jumping from row to row. After each problem subjects were given feedback on their solutions.

Writing speed tests were administered to control for differences in subjects' writing speeds. In the tests subjects were asked to write the correct answers for the test problems by copying them from a sheet of paper and by listening and transcribing items from a tape. As an additional experimental control, background data on the subjects was collected.

RESULTS AND DISCUSSION

Errors. Errors were classified by two judges into two categories: overload errors, such as forgetting part of a cell reference or mis-remembering the cell name, and conceptual errors, such as specifying the overall formula incorrectly. Of the 120 errors, five were judged to fall in either category and therefore were not included in further analysis.

Of the 115 remaining errors, 53 were overload errors and 62 conceptual. As predicted, interface representation had an impact on overload error rates ($p<.01$) but not on conceptual errors ($p=.96$) [analysis of variance of the logit transformation of error rates for both overload and conceptual errors]. These results provide support for the claim that conceptual errors are expected to occur before the translation into the interface representation starts. In contrast the frequency of overload errors for positional subjects was 4.5 times greater than for keyword subjects (5.9% vs. 1.3%) and almost the same for conceptual errors (5.4% vs. 4.2%). Figure 5 shows the distribution of both types of errors for each cell type in each interface representation.

For overload errors, cell type DRC was more error prone than SR and SC as predicted ($p<.05$ and $p<.01$; Tukey-Kramer test), but contrary to expectations there was no significant difference between SR and SC. The causes for this result, discussed further in [4], have to do with a mis-assumption in our GOMS models about how much effort was required in re-using a piece of information that was used in an earlier step. There was no interaction between cell type and the interface representation ($p=.20$). This means that cells that are difficult to translate in one representation are also difficult in the other. Finally, the difference in error frequency between the two representations for each cell type was highly significant for SC ($p<.01$), slightly significant for SR ($p<.10$) and not significant for DRC as predicted by the cognitive models shown in Figure 4.

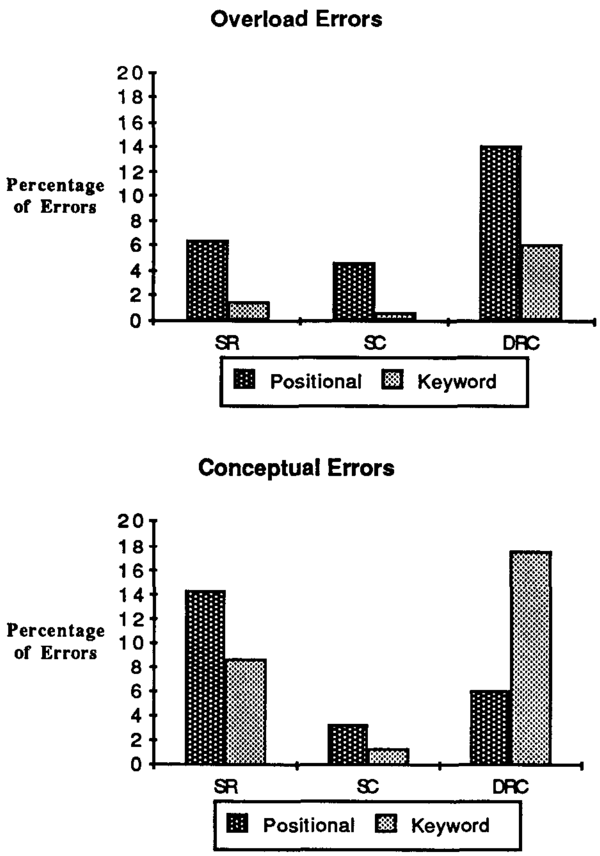


Figure 5. Overload and Conceptual Error Rates

For conceptual errors, cell type SC has a significantly lower error rate than type SR and DRC ($p<.01$). Our task analysis shows that cells in the same column (type SC) represent straightforward relationships among both level and flow variables. On the other hand, cells of type DRC are required for formulas that represent time lags and cells of type SR are included in formulas that mix levels and flows and in formulas that involve recursion. It was therefore expected that more conceptual difficulties would occur when formulas include these more complex financial relationships.

If interface representation does not affect the conceptual process executed before translation, then we should expect no interaction between representation and cell types for conceptual errors. Contrary to this reasoning the lower graph in Figure 5 shows a strong interaction between cell type and representation for conceptual errors. This interaction may be explained by other interface features of the two packages used in the experiment. IFPS encourages the user to think in terms of recursion by automatically propagating formulas; this may explain why the frequency of conceptual errors is smaller for positional subjects in type SR which is usually used in recursive formulas. On the other hand, IFPS makes it difficult to write equations with cells of type DRC by encouraging this same recursive thinking: it is conceptually difficult to formulate in a single equation both initial and steady-state conditions; this may explain why IFPS subjects had so many conceptual problems with DRC compared to Lotus users.

Times. Execution times for formulas with two and three references were between 7 and 35 seconds for both representations. These times are similar to the unit task times found by Card et al in their studies of text editing and circuit design [1]. Figure 6 shows the average time needed to execute each cell reference with both representations. Even though execution time is longer for the keyword representation, most of this time is spent writing because more has to be written.

We were interested not only in total times, but in how much mental effort was involved in each representation. We calculated mental time by subtracting from the execution time the raw writing time, which was estimated from separate writing speed tests. The results show that the positional representation forces skilled financial analysts to spend a little more than 50% of the time in mental operations versus only 20% in the keyword notation.

Average Time per Cell (sec)

	Positional	Keyword
Writing	3.8	8.9
Mental	3.9	2.2
Total	7.7	11.1

Figure 6. Error-Free Execution Times: General Statistics

Mental times for formulas with two and three cell references were between 1 and 10 seconds for the keyword representation and between 3 and 22 seconds for the positional notation. The mental time spent in each cell type was estimated by regressing the total mental time per formula against the number of each cell type in each formula. The results in Figure 7 show large differences in the mental effort required for each cell type in each representation, and support the hypotheses generated by the GOMS models.

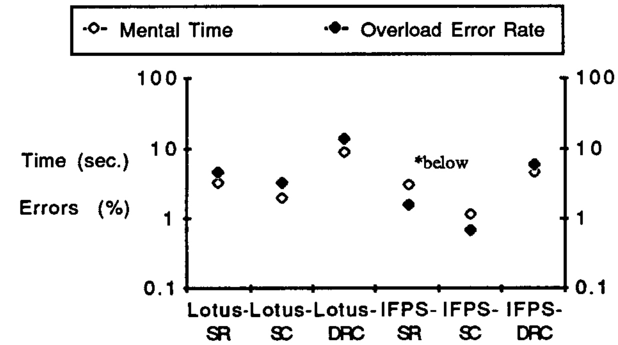
Mental Times
Regression coefficients (standard error)

	Positional	Keyword
Long-SR	3.34 (.38)	4.90 (.96)
Short-SR		1.25 (.75)
SC	2.09 (.12)	1.16 (.19)
DRC	9.25 (.53)	4.88 (.72)
R^2	.78	.59
N	286	148

Figure 7. Mental Times

It is important to clarify that the keyword representation has two methods for writing cells of type SR. In the short method, the user only writes the keyword (e.g. PREVIOUS) while in the long method both the keyword and variable are specified. The mental times for these two methods indicated that the GOMS models can predict times and errors for these variations if the task is accurately represented.

Finally, mental times for each cell reference type and each representation have a high correspondence with overload error frequencies, but not with conceptual error rates. This can be explained by speculating that in financial model building longer mental times are associated with increased effort and higher memory loads, some of which break down and produce simple errors. Figure 8 shows both the overload error rates and the mental times in a logarithmic scale.



* The mental time and error rate for IFPS-SR are composite of both short-SR and long-SR; therefore they are not comparable to the mental times and overload error rates for the other cell references.

Figure 8. Overload Error Rates vs Mental Times

CONCLUSIONS

Interface design decisions have grave consequences for the effort involved in difficult mental work that is supposed to be made easier with software. In particular, we found that for financial modeling systems, an interface that requires the user to "calculate" the cell coordinates imposes a mental workload that produces errors and increases the time attributed to mental effort. Even though the total time to enter a mnemonic cell reference is longer, it is not time involved with "hard work." If one takes into account the *cost of errors*, both the time to correct errors and the intangible cost of undetected errors [3,5,6,8], the keyword representation is considered far superior.

A second, far more reaching outcome of this research, is a demonstration that the GOMS model formalism is useful in predicting both time and error performance. It illustrates its potential in interface design. The user interface representations built into the GOMS models are abstractions of two common properties designed into financial modeling systems. They could readily have represented decisions of unbuilt software. The results of the experiment confirmed the cognitive load predictions of the GOMS models thereby supporting the worth of this analysis method for evaluating design decisions as well as differences in existing software packages.

Finally, the research in this paper extends the application of the GOMS model to a highly complex problem solving task. It demonstrates how it can be applied to the routine cognitive portions of the task and provides insight into how the user's internal problem representation interacts with the external constraints of the software package.

ACKNOWLEDGEMENTS

This research was supported in part by a University of Michigan Rackham Grant. The authors would like to thank Alan G. Merten, Raymond R. Reilly and J. Frank Yates for their valuable comments and suggestions throughout the execution of the research. Special thanks are due to the financial planners who took part in this study on a purely voluntary basis. Finally, thanks to Diane L. Morrow-Lerch for her help in data recording and analysis.

REFERENCES

1. Card, S.K., Moran T.P. and Newell, A. *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum, Hillsdale, N.J., 1983.

2. Davidson, S., Stickney, C.P. and Weil, R.L. *Financial Accounting*. The Dryden Press, Hinsdale, Ill., 1979.

3. Ditlea, S. Spreadsheets Can Be Hazardous to Your Health. *Personal Computing*, January 1987, pp. 62-67.

4. Lerch, F. J. Computerized Financial Planning: Discovering Cognitive Difficulties in Model Building, Ph.D. Dissertation, The University of Michigan, 1988.

5. Moskowitz, R. Unspoken Nightmare: "Stupid" Spreadsheets. *Software News*, May 1987, p. 51.

6. Moskowitz, R. Spreadsheets' Quiet Horror. *Computerworld*, May 5, 1987, p. 35 ff.

7. Olson, J.R. and Nilsen, E. Analysis of the Cognition Involved in Spreadsheet Software Interaction, *Human-Computer Interaction*, 1988.

8. Seymour, J. Left Unchecked, Spreadsheet Can Be a What-If Disaster, *PC Week*, August 21, 1984, pp. 37-38.

9. Shneiderman, B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, Reading, Mass., 1987.

10. The Wall Street Journal, Executives' Computers, February 23, 1987, p. 21.

11. Van Horn, J.C. *Fundamentals of Financial Management*. Prentice-Hall, Englewood Cliffs, N.J., 1983.

12. Waugh, N.C. and Norman, D.A. Primary Memory. *Psychological Review*, Vol. 72, 1965, pp. 89-104.