1986

END USER LEARNING BEHAVIOR IN DATA ANALYSIS AND DATA MODELING TOOLS

Sirkka L. Jarvenpaa
University of Texas at Austin

Jefry J. Machesky
Washington Square Capitol Inc.

END USER LEARNING BEHAVIOR IN DATA ANALYSIS AND DATA MODELING TOOLS

Sirkka L. Jarvenpaa
School of Business
University of Texas at Austin

Jefry J. Machesky
Washington Square Capitol Inc.

ABSTRACT

The research examined naive user analysts' learning of data analysis skills; namely, (1) the difficulty of learning data analysis, (2) the differential learning rates among development tools, and (3) the dimensions of the tools contributing to the learning differences. A total of fifty-six students participated in two experiments. The experiments involved repeated trials of practice and feedback in drawing application-based data models. On average, the participants were experienced end users of computer systems in organizations. The two tools examined in the experiments were the logical data structure model (LDS), which is based on the entity-relationship concept, and the relational data model (RDM). The correctness of the models improved over the trials in both LDS and RDM groups with LDS users performing better than RDM users, particularly in terms of representing relationships. LDS users were found to be more top-down motivated in their method of analysis than RDM users. The study suggests that among end users, the LDS formalism is more easily learned than the RDM formalism. The results also imply that end-user training should stress conceptual top-down analysis, not bottom-up output directed analysis.

INTRODUCTION

The development of high quality systems by end users requires effective training and tools that support and improve the users' problem-solving approaches (Davis, 1982; Alavi, 1985). Yet to provide effective training and tools, we must understand the skills needed to perform analysis, design, and programming tasks. Much of the existing behavioral work on development has investigated programming (Pennington, 1982; Sheil, 1981). Few studies have addressed analysis or design (Jeffries, *et al.*, 1980; Vitalari and Dickson, 1983). Within analysis and design, data instead of procedure specification is of particular importance because with non-procedural languages, users (i.e., naive analysts) primarily need to describe the data and relationships for an application (Harel and McLean, 1985). Little empirical research exists on how people learn to conceptualize, analyze, and design data.

The objective of this research is to investigate how available development tools support naive analysts in learning data analysis. The key questions of the research are: (1) How difficult is data analysis for naive analysts? Does the rate at which naive analysts learn vary for different tools? And if so, which dimensions of a tool contribute to learning differences? The next section presents the theoretical concepts underlying the research and the propositions studied.

The research methodology and the analysis of data for the two experiments then follow. The paper concludes with the discussion of the results obtained and directions for further research.

CONCEPTUAL FRAMEWORK

Data Analysis

Data analysis is concerned with the identification and definition of data objects and relationships required by the system under analysis.[1] Data analysis is a subset of systems analysis. Weinberg (1980) defines systems analysis as "the examination, identification, and evaluation of the components (data and processes) and their interrelationships involved in systems..." (p. 6).

The process of data analysis is primarily cognitive in nature; other skills, such as interpersonal interaction and organizational skills, facilitate the cognitive process. Data analysis involves (Jeffries, *et. al.*, 1980; Borgida *et. al.*, 1985; Ridjanovic, 1985):

1. partitioning the original problem into a collection of subproblems with manageable data structures.
2. deriving the relevant information objects.
3. understanding and representing the relationships among objects.
4. formulating questions to refine and discover omissions or inconsistencies in objects and relationships.

These cognitive procedures used to accomplish the task of data analysis combine to form a cognitive skill or a set of cognitive skills.

[1]Note that some authors use the term 'data analysis' to refer to a much broader range of tasks, including conceptual design, schema design and database design (e.g., Howe, 1983).

Learning

Anderson (1982) has proposed a three-stage learning model for cognitive skills (Figure 1). In the first (cognitive) stage, the instruction for the skill being taught is encoded as a set of declarative statements about the skill. This is called *declarative knowledge*. In the second (associative) stage, a smooth procedure is worked out to perform the skill as the compiled statements reveal their procedural form. This is called *procedural knowledge*. In the third (autonomous) stage, the procedural form of the skill undergoes a process of continual refinement, which results in increased speed and accuracy in performance of the skill. Automation is believed to occur primarily in low-level skills (Wiederbeck, 1985).

Stage 1:	Cognitive
Stage 2:	Associative
Stage 3:	Autonomous

Figure 1.
3-Stage Learning Model of Cognitive Skills.

Data analysis, like programming or reading, involves both low- and high-level skills. Low-level skills include knowledge of the notation and grammar of the formalism, and knowledge of the formation and meaning of a simple data model. High-level skills involve the knowledge used to construct complex data models and may even require some level of automation of low-level skills. The current research examines the learning of low-level skills among naive analysts. We expect that:

> **Proposition 1:** Construction of even simple data models requires learning.

Tools for Data Analysis

The rate of learning is expected to vary by the tool used. A tool for data analysis is any combination of *formalism* (notation and grammar) and *method* that helps an analyst interpret and represent the meaning of data. The tool that best supports naive analysts in their learning data analysis skills is believed to be the one that has the closest "cognitive fit" with the analyst's

natural skills and abilities. This is because naive analysts do not have the frequent exposure or conditioning to a particular tool necessary to adjust their behavior to the tool's idiosyncracies and limitations. Both (1) the formalism of the tool and (2) the method of analysis that the tool promotes are believed to affect the cognitive fit.

First, we postulate that the tool that has a formalism with the greatest syntactical clarity and discriminality in its notation and grammar puts the least amount of burden on a naive analyst's memory and processing resources. The minimized mental load should favorably contribute to cognitive fit and, in turn, to learning. Perceptual obviousness of the syntactical notation has been argued by others to influence learning and performance (Green, 1980).

Second, a tool that promotes a top-down production of data models is postulated to have a better cognitive fit to a naive analyst than a tool that promotes a bottom-up production of models. Bottom-up processing entails abstraction from basic inputs, or data, to general principles such as entities, whereas top-down processing relies on first deriving the general concepts, such as entities, followed by detailed attributes (Palmer, 1975; Norman and Bobrow, 1976). Simon (1981) has argued that people process information more efficiently when complex structures are represented in a top-down hierarchical fashion.

To test the arguments for the formalism and method of analysis, two data modeling tools are selected for comparative testing: (1) logical data structure (LDS), which is based on the entity-relationship model, and (2) relational data model (RDM)(see Carlis (1985) for an explanation of the LDS formalism and Tsichritzis and Lochovsky (1982) for an explanation of the RDM formalism). Note that the research only used the formalisms of the tools to examine learning behavior in the construction of simple static data models. The purpose of the research was not to establish the overall superiority of either formalism.) LDS is believed to have a formalism with greater syntactical clarity and discriminality than RDM (see Figure 2). An example of perceptual clarity in LDS is the symbol of a relationship. A line inherently implies connection. In contrast, a relationship in RDM is represented implicitly by a data element. The clarity of RDM is further reduced by the use of a 'table' as a relation, because novices might easily mistake a table for a report. LDS uses an oval to represent an entity, which is not likely to be confused with a report. LDS also fulfills the discriminality criterion better than RDM; namely, there is one and only one symbol to express each concept in the formalism. RDM, on the other hand, uses a box for the three primary constructs.

The differences in perceptual characteristics of the LDS and RDM are expected to lead naive analysts to produce data models differently. The centrality and prominence of the symbol for an entity in LDS is expected to promote top-down processing. Ridjanovic (1985) has suggested that LDS leads a naive analyst to concentrate first on entities and relationships, followed by attributes, thereby eliciting top-down processing. Conversely, due to the dominance of "attribute boxes," a naive analyst using RDM is expected to proceed bottom-up, identifying attributes first, then drawing a "box" around the attributes, and finally naming the "box." Another reason that a bottom-up approach to data analysis is more likely when using RDM than LDS is that RDM does not force the analyst to draw entities (or relations). With RDM, attributes can be identified and grouped to form an entity.

In summary, LDS is postulated to provide a better "cognitive fit," and thus, result in faster learning and more top-down motivated analysis than RDM. We expect that:

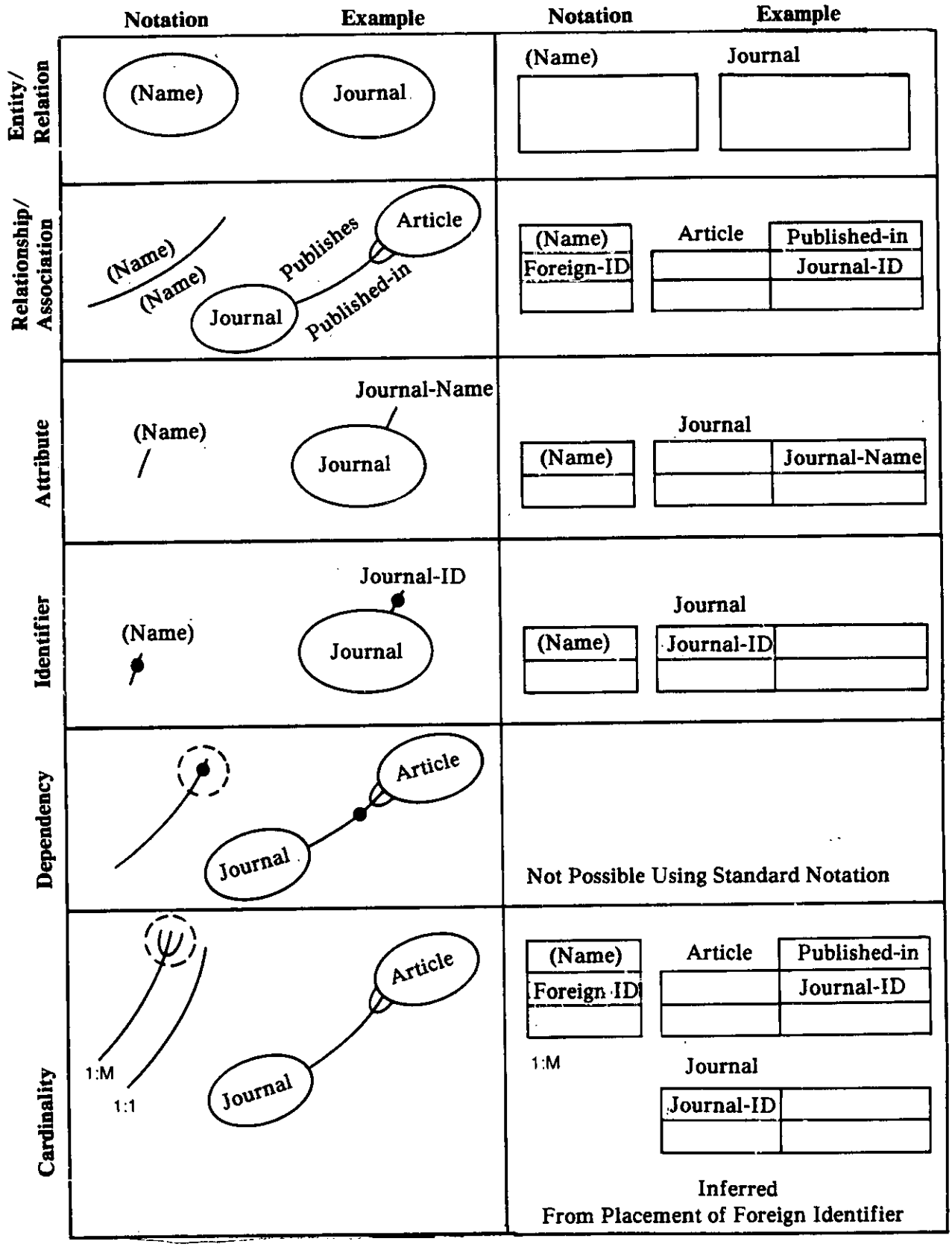
> **Proposition 2:** LDS users produce more accurate data models and in less time than RDM users
>
> **Proposition 3:** LDS users adopt a top-down approach; RDM users adopt a bottom-up approach to data analysis.

EXPERIMENT I

Design

The two treatment variables were (1) the data modeling tool (RDM vs. LDS), and (2) the amount of practice (number of trials). The dependent variables were (1) accuracy of the data model produced, (2) time to draw the data model, (3) knowledge about the notation and grammar of the formalism, and (4) approach followed in drawing the data model.

Figure 2. LDS and RDM Notation.



Independent variables

The *tool* variable consisted of providing subjects with the notation and grammar for either the LDS or RDM formalism, and instructions for carrying out the analysis (i.e., method). The instructions for RDM included rules to normalize models using the concept of functional dependency. Instructions to normalize were not included in LDS because it is argued that the use of LDS automatically results in normalized models (Carlis, 1985).

Figure 3 illustrates the general data analysis method that was included in instructions for both LDS and RDM groups. The method was included because, while LDS and RDM formalisms were expected to guide the processing of data for analysis, their subtle guidance was not considered sufficient help for naive analysts. The method in Figure 3 was constructed using the data modeling literature (e.g., Carlis, 1985), introspections of an expert data analyst, an experienced and a naive data analyst, and the second author's practical experience in data analysis. The method was constructed so as not to impose any direction for processing of data. Subjects were free to process data either top-down, bottom-up, or some variation of both.

The *practice* variable consisted of exposing subjects to four experimental trials. Drawing a data model for a particular application task constituted an experimental trial. The order of presentation of the four tasks was balanced across the subjects to control for order effects.

In the development of the four application tasks, no attempt was made to replicate the complexity and difficulty of real-world applications. The quest for realism was avoided because of the likely confounding effect of the environmental variables on the experimental results, and the importance of controlling for the equivalency of the four tasks. To ensure equivalency of the tasks, we controlled for task complexity, task structure, task difficulty, and time pressure. To control task complexity, each application included five entities, thirteen attributes, and four (two-way) relationships (see the example in Figure 4). This combination of elements and relationships for the tasks was chosen because pilot tests showed that more complex tasks were too time consuming, while simpler tasks proved to be too trivial even for naive analysts.

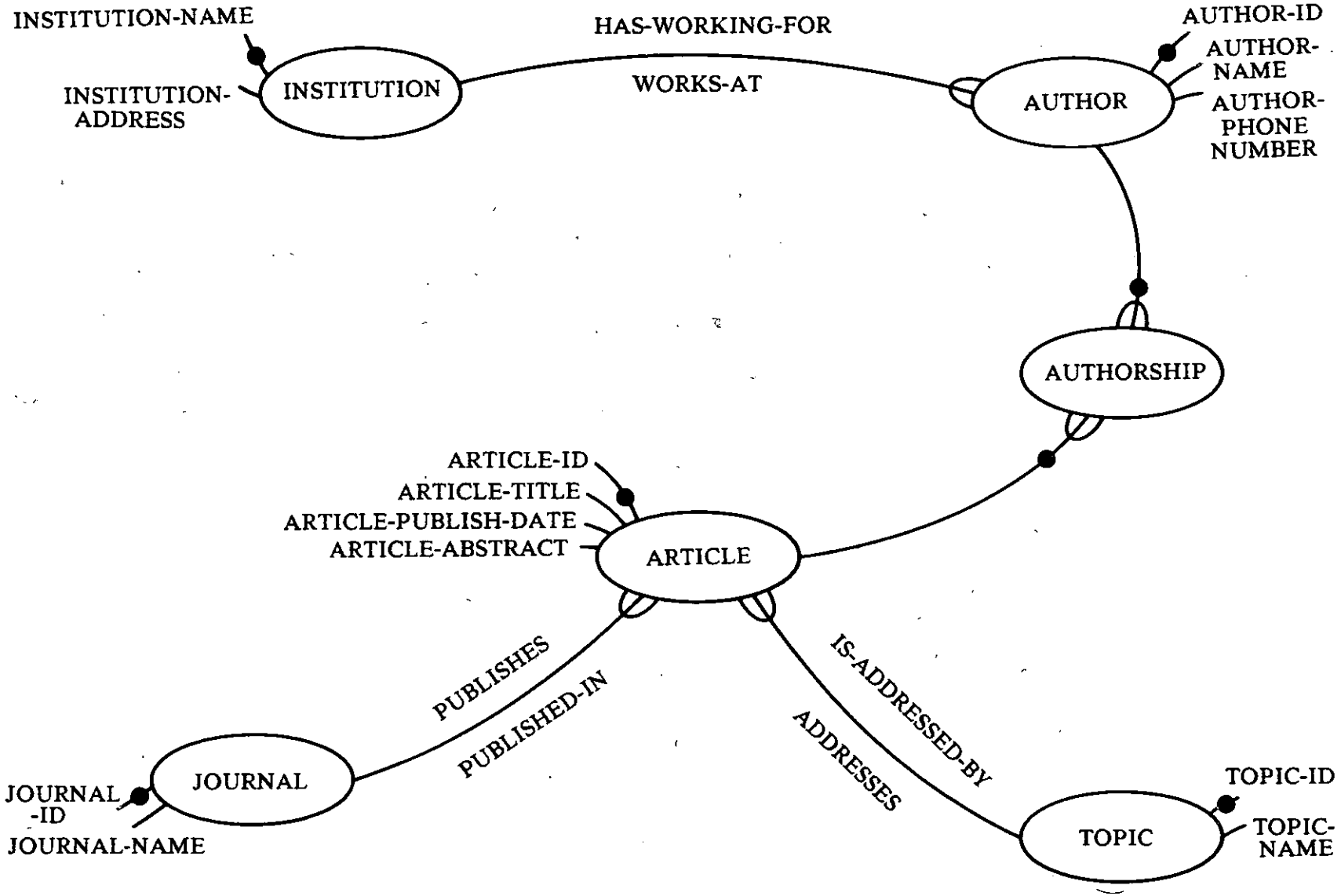
Task structure was controlled by keeping the presentation sequence of task information constant for each application. Each set of task materials contained (1) a task description, which included the enterprise rules (e.g., an article never appears in more than one journal), (2) output requirements (e.g., a list of articles and authors including article-title and author-name), and (3) sample output reports for the application (see Appendix A). Task difficulty was held constant by using applications that were familiar and easy for people to understand, such as keeping track of articles for future reference. Additionally, to insure that subjects could not get the correct model by copying the model from previous tasks, none of the entities, attributes, or relationships occurred more than once. To control time pressure, the four tasks shared the same time limit and had one-page task descriptions with a mean of 261 words and a standard deviation of 5 words.

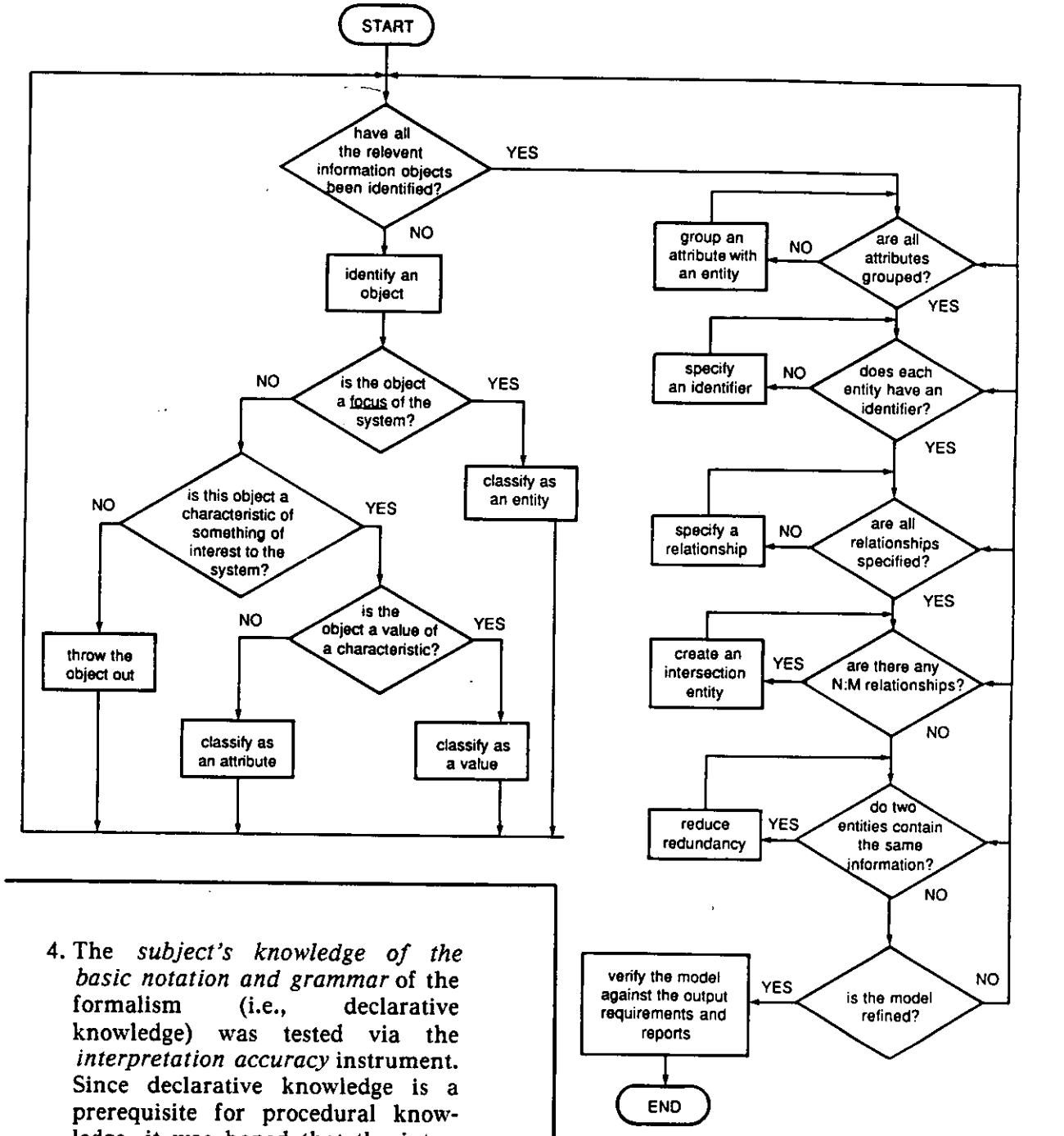
Measurement of dependent variables

Subjects were measured on four dependent variables.

1. The *accuracy* of data models produced was the primary measure of procedural or skill knowledge gained by the subject. The accuracy was assessed against the "correct" data model on three dimensions:
   a. the number of required entities
   b. the number of required attributes
   c. the number of required relationships
2. The "required" implies that the object drawn by the subject existed in the "correct" model.
3. The *time* required by the subject to complete the data model was measured. Measurement of time was considered important because prior laboratory studies (e.g., Bettman and Zins, 1979) have found a tradeoff between accuracy and time.

Figure 4. Article Reference System.





4. The *subject's knowledge of the basic notation and grammar* of the formalism (i.e., declarative knowledge) was tested via the *interpretation accuracy* instrument. Since declarative knowledge is a prerequisite for procedural knowledge, it was hoped that the interpretation accuracy would explain some of the observed differences in the accuracy of data models produced. The instrument required the subject to:

- recognize the notation for different constructs in the formalism.
- recognize the rules pertaining to representing relationships of varying degrees.
- specify the search path for retrieving information using the data model.

5. Whether the subject followed a top-down or bottom-up motivated approach in constructing data models was captured through the subject's verbalization of thoughts, or protocols. The protocols of the ten most fluent verbalizers from each treatment group were transcribed and examined via an index that measured the proportion of objects that the subject conceptualized top-down (vs. bottom-up):

   - an entity (E) conceptualized before (vs. after) its attributes
   - an attribute (A) conceptualized after (vs. before) its entity
   - a relationship (R) conceptualized in terms of entities (vs. in terms of attributes).

The index ranged from -1 (pure bottom-up) to +1 (pure top-down), and was calculated for each subject as below.

Note that subjects were only required to speak aloud during the second and fourth trials. This was done because the pilot tests indicated that speaking substantially tired the subjects which, in turn, slowed down their rate of learning. However, while speaking might have slowed down the process of automating the analysis skills, verbal reports should not have altered performance (see Ericsson and Simon, 1984).

Subjects

Thirty six continuing education students enrolled in an introductory information systems (IS) course participated in the study. On average, the students were 28 years old and had three and one-half years of full-time experience in business or administrative positions. Sixty four percent (64%) used computers daily at their job and 81% at least weekly. Seventy eight percent (78%) had written at least one computer program; 64% reported that they had never designed a file. Seventy five percent (75%) were unfamiliar with any data modeling or data or systems analysis tools; the rest indicated 'slight' familiarity. Subjects were randomly assigned to one of the two treatments: LDS and RDM. The experiment required three hours of the subject's time and was administered individually.

Procedure

Prior to the data modeling tasks, subjects completed a research participation consent form and an agreement to keep the nature of the study confidential. As a performance incentive, subjects were informed that prize money of $50, $35, $25, and $10 would be awarded to the top four performers on the data modeling tasks. Next, subjects read the exercise scenario which described the data analysis task within systems development and stressed the importance of not relying on any other knowledge about the applications than what was described in the task materials. Then, subjects were provided with instructions on the analysis method as well as the notation and grammar rules for the formalism (seven pages). At the end of the instructions, the method for data analysis was summarized in one page to provide a quick reference for the subjects. Subjects could refer to the instructions at any time.[2] Once the subjects announced that they had completed reading the instructions (twenty to thirty minutes), they were provided with a description, output requirements, sample output reports for each application, and blank sheets of paper for drawing the data models. A maximum of twenty-five minutes was allotted for drawing a data model for each application. After completing their data model for one application, subjects were shown the "correct" data model prior to working on a data model for the next application. After each trial, subjects were asked to complete the interpretation accuracy instrument. During the second and fourth applications, subjects were asked to think aloud, that is, verbalize their thoughts. During the verbal report, it was sometimes necessary for the experimenter to remind the subject to speak aloud. The experimenter, however, never

[2]The instructional material was extensively pilot-tested.

$$\text{Index} = \frac{\#\text{ of objects (E,A,R top-down} - \#\text{ of objects (E,A,R) bottom-up)}}{\#\text{ of objects (E,A,R) top-down} + \#\text{ of objects (E,A,R) bottom-up}}$$

probed for specific motivations or reasons for behavior exhibited by the subject. The session ended with a short debriefing.

Results

Table 1 shows summary statistics for the accuracy of the data models produced, completion time, and interpretation accuracy for the two experimental groups. Except for completion time, larger scores correspond to better performance. Simple observation of means suggests that while improvement in performance occurred in both groups, the LDS group performed generally better than the RDM group in all four trials. Table 1 also shows that LDS users were more top-down motivated in their method of analysis than RDM users.

Data from non-protocol trials of 1 and 3 were examined to test Propositions 1 and 2 on learning and tools. Protocol data from trials 2 and 4 were examined to test Proposition 3 on the method of analysis. This division of data was performed because it was apparent from observations of subjects that talking aloud created a significant additional burden on the subjects' mental resources; thus, protocol and non-protocol trials could not be compared directly.

The data from trials 1 and 3 were analyzed in two steps. First, a doubly-multivariate analysis of variance with repeated measures model was fitted to the data for data model accuracy (number of required entities, attributes, and relationships) and completion time (see Bock, 1975). A multivariate analysis was necessary because Pearson product-moment correlation coefficients indicated high correlation among dependent variables. The results from mul-

Table 1. Means and Standard Deviations of Dependent Variables for LDS and RDM Treatment Groups.

		N	Trial 1 Mean (SD)	Trial 2 Mean (SD)	Trial 3 Mean (SD)	Trial 4 Mean (SD)
Required Entities (Max: 5)	LDS	20	4.50 (1.24)	4.75 (0.44)	4.95 (0.22)	5.00 (0.00)
	RDM	16	3.88 (2.85)	4.44 (0.96)	4.56 (0.63)	4.13 (1.50)
Required Attributes (Max: 13)	LDS	20	10.70 (4.26)	11.75 (2.34)	12.20 (2.07)	11.90 (2.15)
	RDM	16	8.25 (4.49)	11.12 (2.50)	11.56 (1.79)	11.38 (3.42)
Required Relationships (one-directional) (Max: 8)	LDS	20	6.60 (2.35)	6.10 (2.63)	7.50 (1.10)	7.00 (2.00)
	RDM	16	1.25 (1.44)	2.88 (3.01)	4.50 (2.25)	3.50 (3.46)
Completion Time (Max: 25)	LDS	20	21.26 (4.19	20.56 (5.18)	16.26 (4.42)	17.22 (5.12)
	RDM	16	24.50 (1.55)	23.64 (3.27)	21.88 (4.46)	21.93 (4.81)
Interpretation Accuracy (Max: 11)	LDS	20	8.00 (1.65)	8.05 (2.26)	8.70 (1.69)	8.75 (1.71)
	RDM	16	6.25 (2.44)	7.75 (3.00	7.38 (2.78)	7.44 (2.31)
Analysis Approach (-1 purely bottom-up;	LDS	10		0.91 (0.24)		1.00 (0.00)
+1 purely top-down)	RDM	10		0.39 (0.57)		0.59 (0.59)

tivariate analysis showed that the tool used (between-subject factor), the amount of practice (within-subject factor), and their interactions were significant at the .002 level.

The second step in the analysis was to employ a procedure described by Messmer and Homans (1980) to determine which dependent variable produced significant effects. The procedure consists of a series of step-down tests in which a single dependent variable is tested while adjusting for the effects of the other dependent variables. The adjustment entails entering the preceding dependent variables as covariates for tests on remaining dependent variables. Progressively, all except the last dependent variable enter as covariates in the model. To use the procedure, data were recoded to fit a one-way ANOVA model. Because of the correlation between dependent variables, the individual statement levels of significance were .0125 assuming .05 family level of significance (Neter and Wasserman, 1974).

To use the Messmer and Homans' approach, it was necessary to set up *a priori* ordering of the importance of the dependent variables. Required entities were selected as the most important dependent variable on the premise that correct conceptualization of entities is a prerequisite for correct identification of relationships and attributes. Required relationships were the second most important variable on the premise that it is easier to add attributes than relationships to an existing model or system. Attributes followed relationships in importance. Time was considered least important because subjects were told that they were to strive for accuracy of the models, not for maximum speed. Accuracy was emphasized because, in the initial stages of skill learning, performance improves in accuracy, but not necessarily in time (Anderson, 1982).

Significant differences in the study were found at the .0125 level only in relationships and in completion time (relationships, $F(3,59)=36.4$; completion time,$F(3,57)=124.06$). While a significant tool effect (between-subject factor) for required relationships was detected across the trials, pairwise contrasts showed that only the RDM group improved at .0125 significance level in required relationships from Trial 1 to Trial 3 (within-subject factor). Completion time was not significantly different between LDS and RDM groups between-subject factor in Trial 1, but was significantly different in Trial 3. The LDS group improved in completion time at the .0125 level from Trial 1 to Trial 3; no similar improvement was observed in the RDM group.

The interpretation accuracy data were analyzed separately from performance data. The LDS group recognized the concepts and rules of the tool better than the RDM group at the .05 significance level both in trials 1 ($t=2.56$) and 3 ($t=1.77$). However, only the RDM group improved from Trial 1 to Trial 3 at .05 level ($t=1.93$). Overall, learning in both declarative and procedural knowledge was observed across the trials. Learning was more pronounced in LDS than in RDM in all three components - entities, attributes, and relationships - although it was significant at the .0125 level only for relationships. The data from the final questionnaire also indicated that RDM users found the task more difficult than LDS users ($t=4.76$; $p=.000$).

The analysis of protocol reports indicated that LDS users were significantly more motivated to use a top-down approach than RDM users in both the second and fourth trials ($F= 12.03$, $p=.001$). Eight out of ten LDS users were classified as purely top-down in the second and fourth trials; two out of ten RDM users were purely top-down in the second and fourth trials.

EXPERIMENT II

The purpose of the second experiment was to examine the causes of poor performance among RDM users found in the first experiment. Specifically, we investigated the ease of learning the notation and grammar associated with the RDM formalism, without the added procedure of normalization. This meant that the instructions for the method of analysis given to the revised-RDM group of the second experiment were the same as in the LDS group of the first experiment; only the formalisms of the tools varied across groups. In the second experiment, 20 subjects received the revised RDM treatment. No significant differences were found in subject profiles between the first and second experiment.

The data for LDS and RDM from the first experiment were analyzed with data for the revised-RDM from the second experiment. The

procedures used to analyze the data were the same as those used for Experiment I. The results showed that revised-RDM users performed better than RDM users over the four trials except in terms of entities. Nevertheless, LDS users still performed significantly better than revised-RDM users in required relationships in Trials 1 and 3, and in completion time in Trial 3 (Tables 2 and 3).

In terms of the analysis approach (see Table 2), a significant difference in top-down processing was found between the three experimental groups ($F=2.638$, $p=.043$). The pairwise contrasts indicated that revised-RDM users were less top-down motivated than LDS users in Trial 2 ($t=1.83$; $p=.073$) and in Trial 4 ($t=2.00$; $p=.050$). No significant differences existed between RDM and revised-RDM users.

DISCUSSION

The results from the two experiments provide partial support for Proposition 1. As expected, data analysis required learning. Contrary to expectations, participants were able to construct good data models and recognize the concepts involved in data analysis tools fairly quickly. These findings are, of course, limited to the type of structured tasks the subjects were exposed to in the study. Much more gradual procedural learning patterns might be found in more complex analysis tasks. The results of the comparative effectiveness of LDS and RDM support Proposition 2. LDS users produced more accurate data models and in less time than RDM users. The results also support Proposition 3. LDS users were more top-down motivated than RDM users in their method of analysis.

The results from the experiments suggest that low-level skills related to data analysis are indeed learnable by novices over a relatively short time within a set of structured tasks. Subjects rated their motivation high and were observed by the experimenters to be highly motivated in the experiments; thus, the experiments can be argued to have tested subjects' *ability* to learn about data analysis.

Another noteworthy finding is that LDS was more easily learned than RDM. The results generally agree with the findings of Juhn and Naumann (1985) who found that LDS is more comprehensible than RDM. Our findings also

Table 2. Means and Standard Deviations for Dependent Variables and Revised-RDM Treatment Groups.

		N	Trial 1 Mean (SD)	Trial 2 Mean (SD)	Trial 3 Mean (SD)	Trial 4 Mean (SD)
Required Entities (Max: 5)	Rev.-RDM	20	3.55 (1.67)	3.95 (1.47)	4.40 (1.19)	4.40 (0.75)
Required Attributes (Max: 13)	Rev.-RDM	20	10.60 (4.37)	11.30 (3.33)	12.10 (2.92)	12.65 (0.67)
Required Relationships (one-directional) (Max: 8)	Rev.-RDM	20	1.70 (2.77)	3.60 (3.65)	4.20 (3.66)	4.90 (3.40)
Completion Time (Max: 25)	Rev.-RDM	20	24.14 (1.62)	20.89 (3.90)	18.76 (4.90)	17.79 (4.42)
Interpretation Accuracy (Max: 11)	Rev.-RDM	20	6.95 (3.32)	7.25 (2.22)	7.30 (2.39)	7.65 (2.70)
Analysis Approach (-1 purely bottom-up; +1 purely top-down)	Rev.-RDM	20		0.52 (0.55)		0.57 (0.59)

Table 3. Step Down Tests for Performance Measures LDS, RDM and Revised-RDM Groups.

EXPERIMENT I AND II

Performance Measure	Order	DF	F Value	Sig. of F
Required Entities	1	5,90	2.640	.028
Required Attributes	3	5,88	1.716	.139
Required Relationships	2	5,89	13.283	.000*
Completion Time	4	5,87	11.574	.000*

Significant at the .0125 level

extend the conclusion by Brosey and Schneiderman (1978) that people do not only better comprehend, but also better construct relationships, if they are specified in a two-entity/two-way fashion as in LDS. However, these results are contrary to those of Ridjanovic (1985) who did not find any differences in the quality of data representations between LDS and RDM users. Possibly, the analysts in the Ridjanovic study were less sensitive to the type of tool used because they were more experienced and educated than the naive analysts in our study. Subjects in the Ridjanovic study also received classroom training in tools prior to the experiment.

The results of the current study also indicate that LDS users, who were more successful in general, employed more conceptual top-down, rather than bottom-up output directed analysis. This finding supports the argument that top-down processing is a more natural approach for naive analysts than bottom-up processing. The tentative implication of this argument is that, contrary to the advocacy of bottom-up output directed analysis for nonprocedural languages (Hayden, 1983), end-user training should stress conceptual top-down analysis.

The results also suggest that it is important for the data modeling formalism to have a perceptually clear and discriminating notation for the different constructs. Analysis tools are very seldom designed in light of the cognitive needs of their users, particularly of their more naive users. However, a close cognitive fit is likely to be critical in order for naive users to voluntarily use the tools, and moreover, use them successfully. Voluntary use is essential because, unlike the professional analyst, it is difficult to force the end-user to employ a particular analysis tool.

Thus, further research on tools should concentrate on accumulating knowledge about the features that increase cognitive fit. Future studies should investigate specific notation and grammar rules for relationships, attributes, and entities, as well as for concepts not covered in the current experiments - dependency, composite keys, roles, and normal forms. Also, a study investigating the recall of concepts and rules associated with tools after varying periods of elapsed time (e.g., one week, one month, three months) would provide further insight into the applicability of the tools for naive analysts, assuming that a typical naive analyst employs data analysis tools quite infrequently. Further studies also need to examine whether the relative strengths of the tools are contingent on the complexity of the application tasks.

In terms of learning, further research might replicate the current study by examining the analyst's learning over a greater number of experimental trials. However, the results from such studies might not be very relevant to naive user analysts. Because of their infrequent exposure to data analysis, user analysts may never progress beyond the beginning stages of the learning curve. Instead of extending the length of the study, further research might experiment with different instructions or types of process feedback to find ways to expedite the naive analyst's learning. The results from such studies should help to increase the effectiveness and efficiency of training programs in analysis which, in turn, could mean improved quality of applications.

Acknowledgement

The authors are indebted to Iris Vessey for her detailed comments on an earlier version of this paper. We also thank Joni Johnson, Dzenan Ridjanovic, and V. Sambamurthy for their help in the research project.

REFERENCES

Alavi, M. "Some Thoughts on Quality Issues of End-User Developed Systems," *Proceedings of the 21st ACM Annual Computer Personnel Research Conference,* May 2-3, 1985, Minneapolis, Minnesota, pp. 200-207.

Anderson, J.R. "Acquisition of Cognitive Skill," *Psychological Review,* Volume 89, Number 4, 1982, pp. 369-406.

Bettman, J.R. and Zins, M.A. "Information Format and Choice Task Effects in Decision Making," *Journal of Consumer Research,* Volume 6, September 1979, pp. 141-153.

Bock, R.L. *Multivariate Statistical Methods in Behavioral Research,* McGraw- Hill, New York, New York, 1975.

Borgida, A. Greenspan, S., and Mylopoulos, J. "Knowledge Representation as the Basis for Requirements Specification," *Computer,* Volume 18, Number 4, April 1985, pp. 82-91.

Brosey, M. and Shneiderman, B. "Two Experimental Comparisons of Relational and Hierarchical Database Models," *International Journal of Man-Machine Studies,* Volume 10, 1978, pp. 625-637.

Carlis, J.V. "Logical Data Structures," Working Paper #TR 85-23, University of Minnesota, Minneapolis, Minnesota, 1985.

Davis, G.B. "Caution: User Developed Systems can be Dangerous to Your Organization, MISRC-WP-82-04, University of Minnesota, Minneapolis, Minnesota, 1982.

Ericsson, K.A. and Simon, H.A. *Protocol Analysis: Verbal Reports as Data,* The MIT Press, Cambridge, Massachusetts, 1984.

Green, T.R.G. "Programming as a Cognitive Activity," in *Human Interaction with Computers,* H.T. Smith and T.R.G. Green (eds.), Academic Press, London, England, 1980.

Harel, E.C. and McLean, E.R. "The Effects of Using a Nonprocedural Computer Language on Programmer Productivity," *MIS Quarterly,* Volume 9, Number 2, June 1985, pp. 109-120.

Hayden, R.L. "Design Strategy for Non-Procedural Languages," *Journal of Systems Management,* June 1983, pp. 12-15.

Howe, D.R. *Data Analysis for Data Base Design,* Edward Arnold, London, England, 1983.

Jeffries, R., Turner, A.T., Polson, P.G., and Atwood, M.E. "The Processes Involved in Designing Software," Science Applications, Englewood, Colorado, 1980.

Juhn, S.H. and Naumann, J.D. "The Effectiveness of Data Representation Characteristics on User Validation," in *Proceedings of the Sixth International Conference on Information Systems,* L. Gallegos, R. Welke, and J. Wetherbe (eds.), Indianapolis, Indiana, 1985, pp. 212-226.

Messmer, D.J. and Homans, R.E. "Methods for Analyzing Experiments with Multiple Criteria," *Decision Sciences,* Volume 11, Number 1, 1980, pp. 42-57.

Neter, J. and Wasserman, W. *Applied Linear Statistical Models,* Richard D. Irwin, Homewood, Illinois, 1974.

Norman, D.A. and Bobrow, D.G. "On the Role of Active Memory Processes in Perception and Cognition," in *The Structure of Human Memory,* C. N. Cofer (ed.), W.H. Freeman and Company, San Francisco, California, 1976.

Palmer, S.E. "Visual Perception and World Knowledge: Notes on a Model of Sensory-Cognitive Interaction," in *Explorations in Cognition,* D. A. Norman and D. E. Rumelhart (eds.), W.H. Freeman and Company, San Francisco, California, 1975.

Pennington, N. "Cognitive Components of Expertise in Computer Programming: A Review of the Literature," Cognitive Science Technical Report Series #46, University of Michigan, Ann Arbor, Michigan, 1982.

Ridjanovic, D. "Comparing Quality of Data Representations Produced by Nonexperts Using Data Structure and Relational Data Models," Unpublished Doctoral Dissertation, University of Minnesota, Minneapolis, Minnesota, 1985.

Sheil, B.A. "The Psychological Study of Programming," *Computing Surveys,* Volume 13, Number 1, March 1981, pp. 101-120.

Simon, H.A. *The Sciences of the Artificial,* (2nd Ed.), The MIT Press, Cambridge, Massachusetts, 1981.

Tsichritzis, D.C. and Lochovsky, F.H. *Data Models,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982

Vitalari, N.P. and Dickson, G.W. "Problem Solving for Effective Systems Analysis: An Experimental Exploration,"

Communications of the ACM, Volume 26, Number 11, November 1983, pp. 948-956.
Weinberg, V. *Structured Analysis,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1980.
Wiederbeck, S. "Novice/Expert Differences in Programming Skills," *International Journal Man-Machine Studies,* Volume 23, 1985, pp. 383-390.

Appendix A

Task: Article Reference System

Most students quickly forget the articles they have read while in college. This is unfortunate because articles can be a valuable reference, both during and after college. To help maintain this valuable reference source, you have decided to develop an Article Reference System that will keep track of the articles that you have read.

Conceptual Model

USe the following conceptual model to guide your effort in designing the system:

An article can be published in only one journal, but you might read more than one article from the same journal. You want to be able to reference an article by its topic and so you will assign only one topic to a single article. however, you might read more than one article that has the same topic. An article might have more than one author and a single author can write many articles. Each author will be associated with a single institution (i.e., University of Minnesota, Control Data) and it is possible that a single institution will have more than one author who has written an article that you have read.

Output Requirements

The system should be able to provide the following information:

1. A list of articles including, article-id, article-title, article- publish-date, and article-abstract.

2. A list of journals and articles including, journal-id, journal- name. article-id and article-title.

3. A list of topics and articles including, topic-id, topic-name, article-id and article-title.

4. A list of articles and authors including, article-id, article- title, author-id and author-najme.

5. A list of the institutions that authors work for including, institution-name, institution-address, author-id, author-name and author-phone.

SAMPLE REPORTS OF THESE OUTPUT REQUIREMENTS ARE ATTACHED.

Report 1. Article List

Article ID	Article Title	Article Publish Date	Article Abstract
01	Management Tips	07/01/81	Tips for Managers
02	Computers Today	08/01/84	Computer Industry
03	Investing	08/15/85	Investment Tips
04	Motivation	02/15/85	Motivating Employees

Report 2. Journal Articles

Journal ID	Journal Name	Article ID	Article Title
01	Fortune	03	Investing
		04	Motivation
02	Business Week	01	Management Tips
		02	Computers Today

Report 3. Topics Addressed

Topic ID	Topic Name	Article ID	Article Title
04	Management	01	Management Tips
		04	Motivation
10	Computers	02	Computers Today
20	Finance	03	Investing

Report 4. Authors of Articles

Article ID	Article Title	Author ID	Author Name
01	Management Tips	25	Dr. Pete Bright
		10	Martha Hodding
02	Computers Today	25	Dr. Pete Bright
03	Investing	26	Mary Starr
04	Motivation	01	Dr. Harry George
		26	Mary Starr

Report 5. Institution

Institution Name	Institution Address	Author ID	Author Name	Author Phone
Burroughs Corp.	Detroit, MI	10	Martha Hodding	313-633-3949
Dayton-Hudson	Minneapolis, MN	26	Mary Starr	612-345-3950
U of Minnesota	Minneapolis, MN	01	Dr. Harry George	612-622-1111
		25	Dr. Pete Bright	612-622-3212