Young-Gul Kim and Salvatore T. March

Comparing Data Modeling Formalisms

ccurate specification and validation of information requirements is critical to the development of organizational information systems. *Semantic data models* were developed to provide a precise and unambiguous representation of organizational information requirements [9, 17]. They serve as a communication vehicle between analysts and users. After analyzing 11 semantic data models, Biller and Neuhold [3] conclude that there are essentially only two types of data modeling formalisms: *entity-attribute-relationship* (EAR) models and *object-relationship* (OR) models. Proponents of each claim their model yields "better" representations [7] than the other. There is, however, little empirical evidence to substantiate these claims.

This article presents an empirical study that compares two popular semantic data models: the *extended entity-relationship* (EER) model (an EAR model) [23], and the *Nijssen information analysis methodology* (NIAM) model (an OR model) [16, 24]. The EER model is a more powerful version of the original entity-relationship (ER) model [5]. It is among the most widely used data modeling formalisms [22]. The NIAM model [16] is based on the early binary modeling work by Abrial [1] and Senko [19]. It is widely used in Australia and Europe and is considered, along with the ER approach, to be among the major approaches used internationally [7, 10, 25]. The study analyzes the effects of these modeling formalisms on analyst tasks (building data models) and user tasks (validating data models).

Information Requirement Determination Process

Determining correct, consistent, and complete information requirements is a difficult and challenging task [6]. Figure 1 (adapted from [12]) shows a four-phase process model for requirements determination:

1. *Perception*—Users perceive the enterprise reality. The same enterprise reality may be perceived differently by different users (inconsistency). Any one user may perceive only a part of the reality (incompleteness).
2. *Discovery*—Analysts interact with users to elicit their perceptions.
3. *Modeling*—Based on the informa- tion identified in the discovery phase, analysts build a formal, conceptual model (representation) of the enterprise reality. This model serves as a communication vehicle between analysts and users.
4. *Validation*—Before concluding the model is correct, consistent, and complete, it must be validated. Validation has two aspects: comprehension and discrepancy checking. Users must comprehend or understand the meaning of the model. Then they must identify discrepancies between the model and their knowledge of reality.

This research studies the effects of different data modeling formalisms on the modeling and validation phases. Two experiments were performed, one for each phase. In the modeling experiment, groups of experienced analysts were trained in one of two data modeling formalisms: EER or NIAM. They then performed a data modeling task. In the validation experiment, groups of domain knowledgeable users were trained in one of the same two data modeling formalisms. They performed a validation task. Performances of the groups using each of the data modeling formalisms were evaluated to assess the effects of the formalism on the task performance.

Prior Research

Several prior studies have examined the effects of different data modeling formalisms. These varied in four dimensions: subjects, data models compared, experimental task, and dependent measures. Table 1 summarizes six such studies. All of the studies used students as subjects and all compared semantic data modeling formalisms such as the ER model [5] and the Logical Data Structure (LDS) model [4] with storage representations such as the Relational Data Model (RDM) and data access diagrams (DAD). Experimental tasks included model comprehension, model development, recall, and problem solving. The common dependent measure was "quality of the result."

Juhn and Naumann [12] studied end-user model comprehension. They found that semantic models (LDS and ER) were more effective than data storage models (RDM and DAD) in tasks related to understanding relationships. Ridjanovic [18] studied end-user model building. He concluded that the formalism itself is insufficient to drive the data modeling process. Jarvenpaa and Machesky [11] studied how formalisms support naive analysts' learning of data analysis skills. Analysts using a semantic formalism (LDS) performed better than those using a storage formalism (RDM), particularly in representing relationships.

Shoval and Even-Chaime [21] studied database schema design. They found that normalization, used in RDM, resulted in higher-quality data design, took less time, and was preferred by analysts over the information analysis technique used in

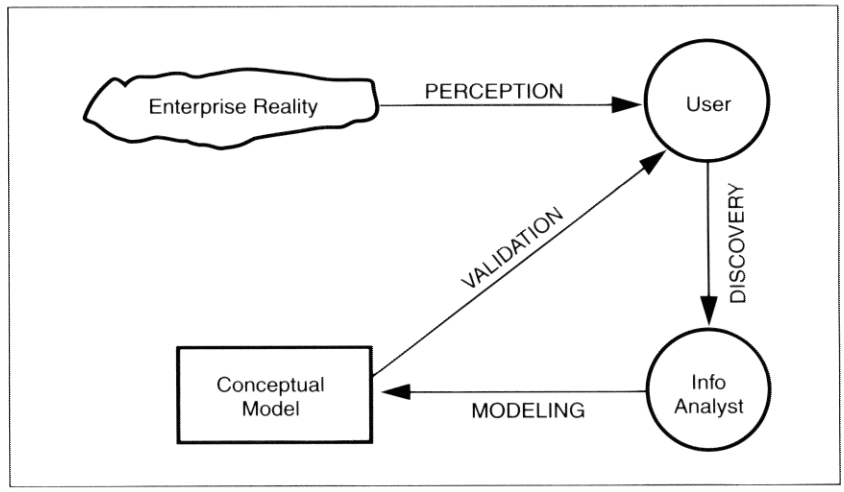


Figure 1. A process model of information requirement determination

Table 1. Summary of recent empirical data modeling studies

Study	Subject	Data Model	Exp. Task	Dependent Measure
Juhn and Naumann (1985)	MIS MBA students (end users)	LDS ER DAD Relational	Comprehension Modeling	Quality of data model
Ridjanovic (1986)	MIS MBA students (end users)	LDS Relational	Modeling	Quality of data model
Jarvenpaa and Machesky (1986)	Students in introductory IS course (analysts)	LDS Relational	Modeling	Quality of data model
Shoval and Even-Chaime (1987)	IS graduate students (analysts)	Normalization (Relational) NIAM (Binary)	Database schema design	Quality of design Time Preference
Leitheiser (1988)	Students in introductory IS course (end users)	LDS Relational	Comprehension Recall Problem solving Query writing	Time Task performance
Batra et al. (1990)	Students in introductory IS course (end users)	ER Relational	Modeling	Quality of data model

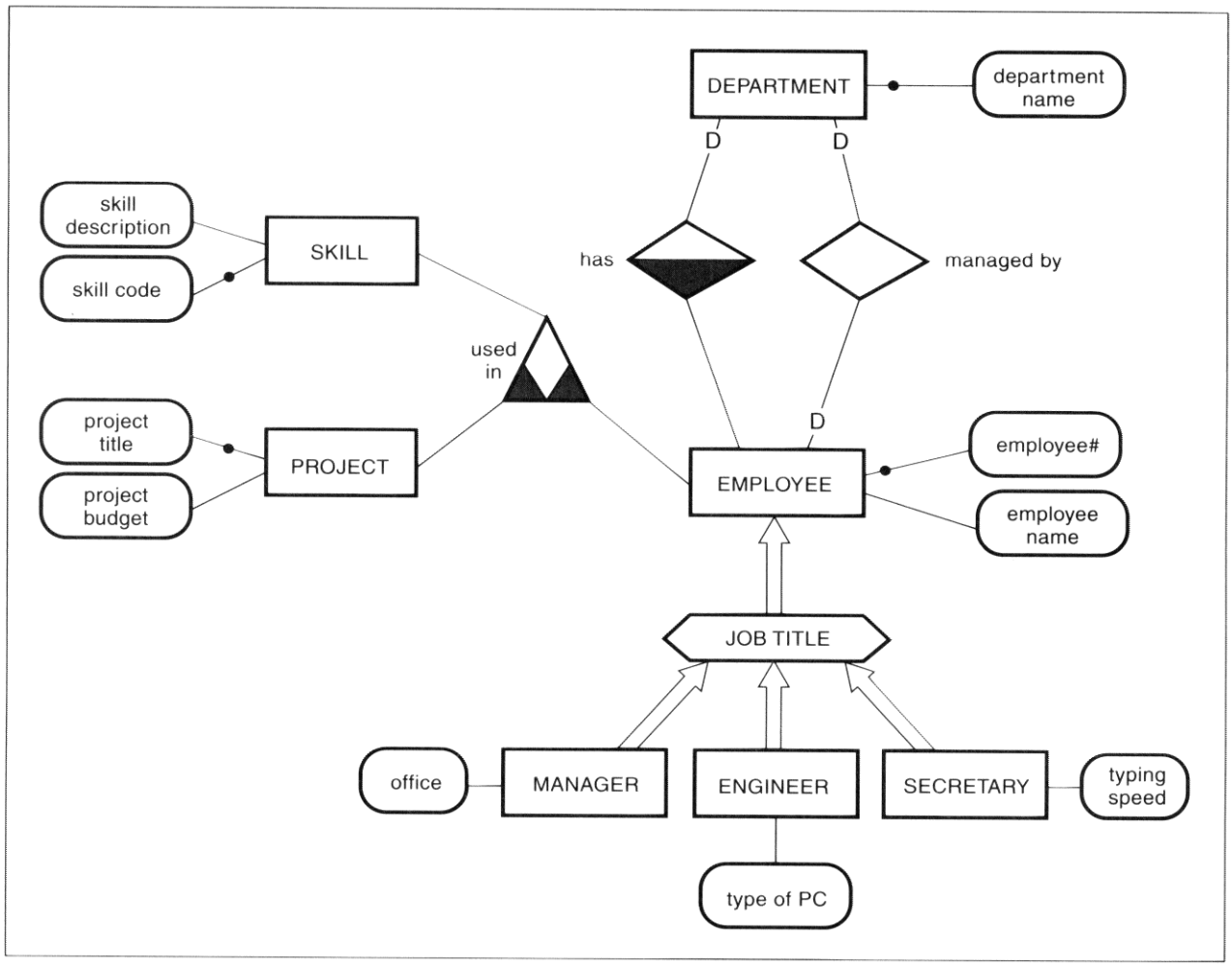


Figure 2. Employee database model in EER formalism

NIAM. Leitheiser [14] studied end-user model comprehension (among other things). He found that a semantic model (LDS) was easier to learn and resulted in higher understanding and recall of a database schema than a tabular representation.

Finally, Batra, Hoffer, and Bostrom [2] studied end-user model building. They found that a semantic model (EER) led to better performance in modeling binary relationships and a certain type of ternary relationship (one-many-many) than did a storage model (RDM). No significant evidence was found to claim that either model led to better overall performance.

This study builds upon the previous studies in terms of variables, evaluation schemes, training, and experimental procedures, but it distinguishes itself from the prior research in the following ways:

1. Subjects include both analysts and users (differentiated subjects).
2. Both model comprehension and model building tasks were performed (differentiated tasks).
3. Two major semantic data models (EAR and OR) are compared (rather than comparing semantic with storage models).
4. Realistic business problems are taken from a real business domain (operations management), including reports and supporting documentation, as would normally be available in a business situation.

EER and NIAM Formalisms

Figures 2 and 3 represent an employee database in the EER and NIAM formalisms, respectively. Comparing these figures illustrates the similarities and differences between these formalisms. They are similar in that they represent the basic facts in the application. For example, they both represent the facts that there are three nonoverlapping types of employees: managers, engineers, and secretaries; that each employee is identified by employee number and described by employee name; that each employee "belongs to" exactly one department (and that a department "has" zero or more employees); and that each department is "managed by" one employee.

However, these facts are represented using different symbols and different logical constructs. The EER formalism differentiates entities (represented by rectangles) from attributes (represented by ovals). It uses diamonds to represent binary relationships and triangles to represent

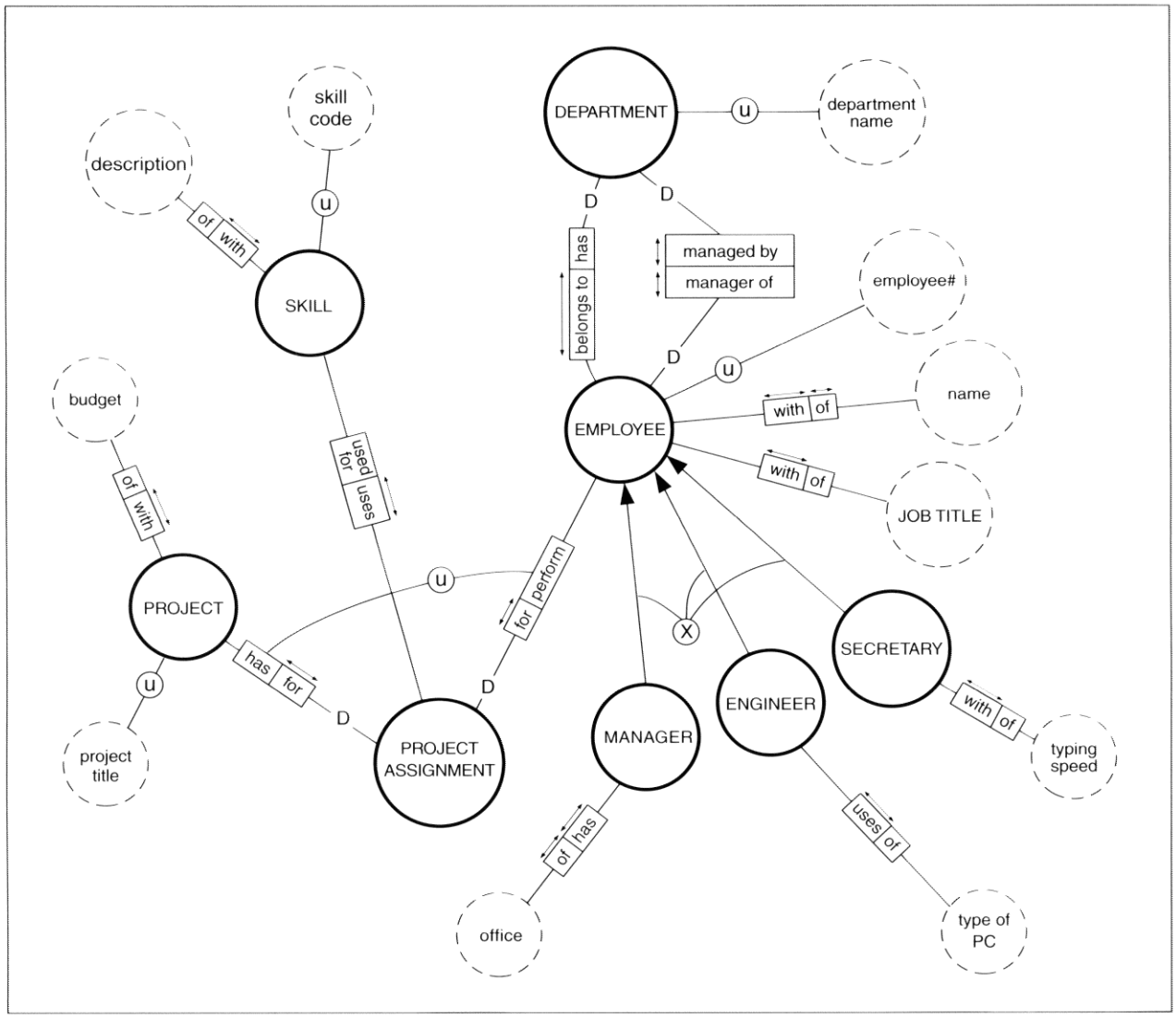


Figure 3. Employee database model in NIAM formalism

ternary relationships. Shading is used to represent the "many" angle(s) of a relationship—e.g., one department (unshaded) "has" many employees (shaded); one skill (unshaded) is "used in" many projects (shaded) by many employees (shaded). A dot represents an identifier and a "D" represents dependency (employee is identified by employee number, and each employee must "have" a department).

The NIAM model differentiates "non-lexical objects," or NOLOTs (represented by solid circles) from "lexical objects," or LOTs (represented by dashed circles). NOLOTs are equivalent to entities; however, LOTs represent domains of values rather than attributes of specific objects. Relationships (represented by boxes) form pairs of sentences describing facts in the application—e.g., employee "belongs to" department and department "has" employee. They represent both relationships and attributes in the EER model.

Arrows above the appropriate verb in the relationship box represent the "many" side of a relationship. Thus, employee "belongs to" one department but department "has" many employees. Ternary relationships are represented by adding non-lexical objects and appropriate constraints—e.g., project assignment. A circled "U" represents a uniqueness constraint (identification), and, as in the EER model, "D" represents a dependency—e.g., project assignment is identified by the combination of project and employee and project assignment is dependent upon project and employee (each project assignment must be "for" an employee and must "have" a project).

To understand the effects of these formalisms on analysts building data models and users validating them, we performed two controlled experiments. The methodology, hypotheses, and experiments are described here.

Research Methodology

Research Model

The research model for the study is shown in Figure 4. The model depicts the relationships among the vari-

ables, tasks, and subjects of the study. The central research question is:

What are the effects of different data modeling formalisms on: 1) the user's ability to perform validation tasks, and 2) the analyst's ability to perform modeling tasks?

Independent Variable: Both experiments have one independent variable: type of data model (EER or NIAM). In the user experiment, each subject was randomly assigned to one of the two treatment groups and trained in the appropriate data modeling formalism. In the analyst experiment, matching and group level randomization techniques were used to assign the analysts to the treatment groups. Again, appropriate training was provided.

Dependent Variables: There are two dependent variables: task performance (validation performance for users and modeling performance for analysts) and perceived usefulness of the formalism. The framework for evaluating user and analyst task performance has two major components: syntactic and semantic [18]. Syntactic performance reflects the subject's competence in understanding the constructs of the modeling formalism. Semantic performance reflects the subject's capability to apply that understanding.

User validation performance consists of two measures: comprehension (measuring syntactic performance), and discrepancy checking (measuring semantic performance). Comprehension performance is measured by the number of correct answers to questions dealing with basic modeling constructs. The grading scheme is based on that developed in [12]. Discrepancy-checking performance is measured by the number and type of model errors identified. The evaluation scheme differentiated types of errors such as entity errors, relationship errors, and attribute errors.

Analyst modeling performance measures the quality of a conceptual model developed. It is determined by the number of correct syntactic and semantic constructs in the subjects' conceptual models. The data model evaluation instrument is based on those developed by Ridjanovic [18] and by Batra, Hoffer, and Bostrom [2]. In addition to the objective performance measures, data for an important behavioral variable, perceived usefulness, was collected from the subjects through a debriefing questionnaire. This variable measures the ease of use and value of the modeling formalism as perceived by the subjects.

Controlled Variables: To guard against confounding effects, three variables were controlled during the experiment: training, time, and task complexity.

Cases

Two operations management cases were used for the experimental tasks. The first case (YBCL) was used for the user comprehension task. The case describes the production environment of a "make-to-order" manufacturing company. It contains 12 entities, 12 relationships (11 binary and one ternary), and 33 attributes (Appendix A).

The second case (Air King) was used for the user discrepancy-checking task and for the analyst modeling task. The case describes the production planning and materials purchasing activities of a "make-for-inventory" manufacturing company (Appendix B). It has two pages of textual descriptions and four supporting figures (containing standard forms and reports). It is larger and more complex than cases typically used in prior research. For the discrepancy checking task, a distorted data model of the case was developed.

User Experiment

Subjects: Twenty-eight graduate business students participated in the

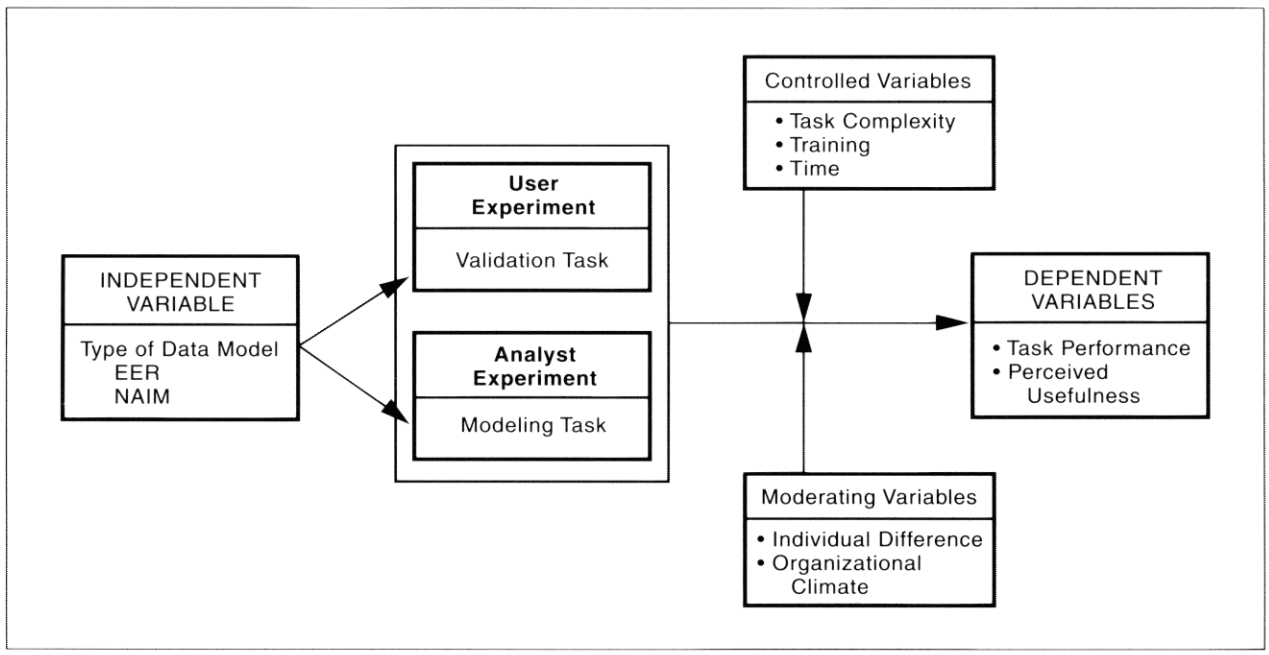


Figure 4. The search model

study as users. All had basic training in operations management (the domain of the experimental task), but none had data modeling experience. They were randomly assigned to one of the two treatment groups. Two kinds of incentives were used. The first was the educational value of learning a powerful modeling tool. Second, rewards of $100, $70, and $50 were given to the top three performers.

Hypotheses: Given equivalent training, we do not expect any significant differences between the NIAM group and the EER group in syntactic competence or in perceived usefulness of the formalism. Since NIAM and EER have about the same number of basic constructs and both have straightforward composition rules, there is no reason to expect that either formalism would be easier to learn or to apply than the other [15].

However, we expect the NIAM group to perform better than the EER group in discrepancy checking. NIAM models are characterized by a strong semantic equivalence between facts about the application, expressed in natural language, and sentences represented in NIAM [10]. NIAM's binary relationships with explicit, directional verbs describe single facts in the application [1,8]. In an EER model, on the other hand, multiple facts are grouped into a structured concept (an entity with attributes) [13]. Hence, the following hypotheses are posited:

HYPOTHESIS 1: *There will be no difference between the NIAM user group and the EER user group in their model comprehension performance.*

HYPOTHESIS 2: *The NIAM user group will perform better than the EER user group in the discrepancy-checking task.*

HYPOTHESIS 3: *There will be no difference between the NIAM user group and the EER user group in their perceived usefulness of the data modeling formalism.*

Training: Subjects were trained in one of the two data modeling formalisms (EER or NIAM). Training consisted of a one-hour lecture and three hands-on problem solving sessions. To ensure the provision of equivalent training for the two treatment groups, the same set of examples, application data models, questions, and instructional materials were used for both.

Experimental Tasks: Users performed a validation task consisting of two subtasks, model comprehension (measuring syntactic competence) and discrepancy checking (measuring semantic competence). In the model-comprehension task they answered a list of questions about the YBCL case based on a conceptual model prepared in their respective modeling formalisms.

In the discrepancy-checking task, each subject was given a correct textual description of the information requirements for the Air King case and a semantically incorrect conceptual model of the same case. After reading the written case, subjects identified all inconsistencies in the conceptual model. User performance was measured by the number of discrepancies found, weighted by the type of discrepancy.

Administration: The user experiments were performed in seven groups over a three-week period. The size of the groups ranged from two to five. Each experiment took about 220 minutes including training time. Subjects were informed of time constraints prior to the beginning of each activity. They were free to refer to all their training materials during the two experimental tasks.

Analyst Experiment

Subjects: Twenty-six practicing information science (IS) analysts from six organizations participated in the study. Most worked as either database analysts or systems analysts/designers. Due to the logistical difficulty inherent in dealing with practitioners, all analysts from one organization were assigned to the same treatment group. Despite the group level random assignment, no significant differences were found between the NIAM and EER treatment groups in their IS experience or familiarity with different modeling formalisms.

Hypotheses: In the modeling task, analysts create a conceptual model of user information requirements. McGee [15] asserts that the information modeling process is simplified if the data modeling formalism supports the direct modeling of real-world situations, that is, if the model provides structure types that are the direct counterparts of real-world information processing concepts.

EER models are more direct than NIAM models, since the structure types in the EER model match the entities as they are described in real-world information systems (i.e., their "record" orientation). Senko [19] observes that the "entity, attribute, relationship" classification gives analysts psychological comfort, since it can be mapped directly to records, something with which they are familiar.

Furthermore the syntactic/semantic model [20] predicts that it is easier to learn a new syntactic representation if a semantic structure already exists. For instance, it is relatively easy to learn another computer programming language if it has the same semantic constructs as a known programming language. However, learning a programming language with radically different semantic constructs may be as hard as or harder than learning the first one, since it will interfere with both the semantic constructs and the syntax of the first language.

We expect, as in Senko [19], that analysts will have greater familiarity with record-oriented semantic constructs than with NIAM constructs such as lexical/non-lexical objects and two-way role concepts. Hence, analysts in the NIAM group are expected to suffer more from interference effects. These observations lead us to the following hypotheses:

HYPOTHESIS 4: *EER analysts will produce a data model of higher semantic quality than NIAM analysts.*

HYPOTHESIS 5: *EER analysts will produce a data model of*

higher syntactic quality than NIAM analysts.

HYPOTHESIS 6: *EER analysts will perceive their modeling formalism to be more useful than NIAM analysts.*

Training: As with the user subjects, analysts were given training in an appropriate modeling formalism. The training consisted of a one-hour lecture and three hands-on problem-solving sessions.

Experimental Task: The two analyst groups performed a modeling task. Each analyst was given a written case description of an operations management problem (the Air King case) and asked to develop a data model in the appropriate modeling formalism. This type of task has been the predominant task in most of empirical data modeling research.

Evaluation of syntactic and semantic performance of analysts for the modeling task was based on the number of major and minor errors found in the data models produced as compared to the "correct" data model produced by an expert (the developer of the case scenario). The categories of syntactic and semantic modeling errors are listed in Appendices C and D, respectively. As in [18], a required construct is considered to be present if there is *any* semantically equivalent construct in the analyst's data model. For example, a concept represented by an entity in the expert's model could be represented as an attribute or a relationship in the analyst's model.

After identifying syntactic and semantic errors, the syntactic and semantic performances for each modeling construct (entity/NOLOT, relationship/role-pair, attribute/LOT) were computed as the percentage of syntactically correct instances of the construct created. A major syntactic error (e.g., failure to name an entity/NOLOT) is assigned 0.5 penalty, and a minor syntactic error (e.g., duplicate entity/NOLOT name) is assigned 0.25 penalty. Semantic performance is calculated similarly except that a major semantic error (e.g., missing an entity/NOLOT) carries 1.0 penalty and a minor semantic error (e.g., extra entity/NOLOT) carries 0.3 penalty. Thus:

$$\text{syntactic performance (\%)} = \frac{N - 0.5 \cdot T_1 - 0.25 \cdot T_2}{N} \cdot 100$$

where N is the number of instances of the construct produced; T_1 is the number of major syntactic errors; T_2 is the number of minor syntactic errors;

$$\text{semantic performance (\%)} = \frac{N[X] - M_1 - 0.3 \cdot M_2}{N[X]} \cdot 100$$

where N[X] is the number of instances of the construct in the expert version; M_1 is the number of major semantic errors; and M_2 is the number of minor semantic errors.

The overall semantic and syntactic performances for each analyst were calculated by averaging the analyst's performances for the individual modeling constructs.

Administration: The analyst experiments were held at each participating organization site over a one-month period. The size of the groups ranged from two to eight. Each experiment took about 235 minutes including training time. The same experimental procedures were followed as in the user experiments.

Results and Discussion

Subject Characteristics

Our hypotheses were based on the premise that users and analysts differ in characteristics such as familiarity with specific conceptual modeling formalisms and degree of record orientation. As shown in Table 2, the presumed differences between users and analysts were, in fact, exhibited by both subject groups. The analysts showed a significantly higher degree of record orientation and were more familiar with entity-relationship concepts than the users. There was no significant difference between the analysts and the users in their familiarity with NIAM.

Discussion of the User Experiment

Table 3 summarizes our findings for the user experiment. Hypotheses 1 and 3a were supported. Comprehension performance (Hypothesis 1) is determined by the competency of each user to understand different modeling constructs and their syntactic rules. Perceived difficulty (Hypothesis 3a) is determined by competence. These results are consistent with the expectation that both user groups would achieve about the same level of syntactic competence after being equivalently trained.

Hypothesis 2 was not supported. Contrary to the claims of superior semantic features of the NIAM model, there were no significant performance differences in the discrepancy-checking task. It is possible that NIAM is not superior to EER in this regard. However, there are several other possible explanations. First, time for the experimental task may have been overly constrained (on average 59.4 out of the allowed 60 minutes were used). Under severe time pressure, subjects may have focused on more abstract representations (entities/NOLOTS) rather than on detailed facts (relationships), where NIAM's advantages lie. The more detailed, two-way role descriptions of the NIAM model and its additional cardinality and dependency constraints may have been overwhelming.

Second, user subjects indicated

Table 2. Analysis of user-analyst characteristics

	Analyst	User	P-value
Logical record orientation	5.76	4.37	0.0002**
Physical record orientation	5.89	5.01	0.015**
ER familiarity	4.50	3.14	0.003**
NIAM familiarity	2.19	2.04	0.736

**With alpha = 0.05

Table 3. Summary of user hypothesis testing

User Hypotheses	Significant Difference?	Hypothesis Supported?	*P*-value
H1: Comprehension performance	No	Yes	0.372
H2: Discrepancy checking performance	No	No	0.919
H3a: Perceived difficulty of formalism	No	Yes	0.660
H3b: Perceived value of formalism	Yes*	No	0.054

*With alpha = 0.1

they were more familiar with the EER concepts than with the NIAM concepts and indicated a higher-than-expected degree of record orientation. These may have offset the semantic power of the NIAM model.

Despite the lack of theory to support expectations of a significant difference, Hypothesis 3b was not supported—the EER users valued their modeling formalism significantly more than the NIAM users. The EER users also perceived the case to be significantly more realistic than did the NIAM users. Given the tabular formats of the supplementary documents (forms and reports rather than sentences), it is possible that the EER constructs matched the case contents more directly than the NIAM constructs, resulting in the higher perceived value. This may also explain why Hypothesis 2 was not supported.

Discussion of the Analyst Experiment

Table 4 summarizes the findings of the analyst experiment. All six semantic performance hypotheses (H4, H4a, H4b, H4c, H4d1, H4d2) were supported. None of the four syntactic performance hypotheses (H5, H5a, H5b, H5c) were supported. That is, the data models developed by the two groups of analysts were significantly different in terms of their semantic quality but were not significantly different in terms of their syntactic quality.

The EER group represented the underlying business semantics significantly better than the NIAM group. The EER analysts' superior semantic performance supports the theoretical arguments made earlier. There, based on the assumption that analysts think in a highly record-oriented way and have a greater familiarity with EAR constructs, the NIAM analysts were expected to suffer more from the interference between their EAR-based knowledge and the different set of semantic constructs used in the NIAM modeling formalism.

As discussed, the syntactic/semantic model [20] predicts that it is easier to learn a new syntactic representation for an existing semantic structure. Why, then, was there no significant support for the hypotheses on syntactic performance? Despite the extensive database experience (20 of 26 analysts), relatively few (12 of 26) analysts had used data modeling in practice. That is, while most of the analysts were record-oriented and familiar with EAR semantic constructs, fewer than half of them had specific syntactic knowledge of any EAR data modeling formalism.

When an analyst experienced in EAR modeling tries to learn and use an entirely different (in syntax and semantic structures) formalism like NIAM, he or she will suffer from both syntactic and semantic interference [20]. This level of syntactic interference, however, should not occur in analysts without data modeling experience, since they lack specific syntactic knowledge. Consequently, the syntactic performance of the EER analysts was not significantly higher than that of the NIAM analysts.

Both hypotheses related to analyst perceptions (H6a, H6b) were strongly supported. The EER analysts perceived their modeling formalism to be less difficult to use and more valuable than that of the NIAM analysts. These results are consistent with the semantic performance results, suggesting that the NIAM analysts had to work harder to use less familiar modeling constructs. The fact that the NIAM analysts expressed a significantly lower confidence in their task outcome than the EER analysts also supports this assertion. The results of the debriefing questionnaire strongly support the external validity of the experiment. The realness ("true-to-life" quality) of the case used in the modeling task (Air King) was very highly rated by the analysts (5.61 on a 1-to-7 scale).

Conclusions

Previous empirical studies involving data modeling formalisms were subject to too much "context simplification." Despite the fact that the type of problem solver and the type of task have significant effects on human problem-solving performance, both context variables were frozen in previous studies. This research was a first step toward a more context-sensitive empirical research paradigm in the data modeling area, with strong emphasis on external validity. The study examined the effects of different data modeling formalisms on analyst performance in developing a data model and on user performance in validating a data model. It made a clear dis-

Table 4. Summary of analyst hypothesis testing

Analyst Hypotheses	Significant Difference?	Hypothesis Supported?	*P*-value
H4: Overall semantic performance	Yes**	Yes	0.003
H4a: Semantic performance (Entity/NOLOT)	Yes**	Yes	0.000
H4b: Semantic performance (Attribute/Lot)	Yes**	Yes	0.008
H4c: Semantic performance (Relationship/Role-pair)	Yes**	Yes	0.021
H4d1: Semantic performance (Dependency constraint)	Yes*	Yes	0.064
H4d2: Semantic performance (Identifier constraint)	Yes*	Yes	0.081
H5: Overall syntactic performance	No	No	0.178
H5a: Syntactic performance (Entity/NOLOT)	No	No	0.145
H5b: Syntactic performance (Attribute/LOT)	No	No	0.218
H5c: Syntactic performance (Relationship/Role-pair)	No	No	0.320
H6a: Perceived difficulty of formalism	Yes**	Yes	0.012
H6b: Perceived value of formalism	Yes**	Yes	0.046

*With alpha = 0.1
**With alpha = 0.05

tinction between how users and analysts utilize data modeling, maintaining that large-scale data models will continue to be developed by analysts interacting with users.

Implications of the Research

In terms of the process model for information requirement determination (Figure 1), previous data modeling research focused mainly on the modeling task. This research involved both modeling and validation tasks. Future research should examine the effects of alternative conceptual data modeling formalisms on the discovery task. This will require the observation of analyst-user interactions.

The findings of this research are encouraging for IS practitioners. Given a small amount of training, users were able to read and validate application data models of nontrivial size and complexity. When more users become data model–literate (capable of validating an application data model produced by analysts), the analysts' job of producing a complete and correct representation of user information requirements will be made much easier. This in turn will lead to the development of more effective information systems. For these things to happen, however, users as well as IS analysts should be trained in an appropriate conceptual data modeling formalism.

Finally, empirical data modeling research to date has been done primarily in an experimental setting. Despite the various research findings, not much is known about the conceptual data model usage in IS practice to which those findings are supposed to apply. Future work should include field studies and "active" research evaluating the effects of data modeling formalisms on real system development applications. ◻

References

1. Abrial, J. Data semantics. In *Data Base Management,* J. Klimbie and K. Koffeman, Eds. North-Holland, Amsterdam, 1974, pp. 1–61.
2. Batra, D., Hoffer, J.A., and Bostrom, R.P. A comparison of user performance between the relational and the extended entity relationship models in the discovery phase of database design. *Commun. ACM 33,* 2 (Feb. 1990), 126–139.
3. Biller, H., and Neuhold, E. Concepts for the conceptual schema. In *Architecture and Models in Data Base Management Systems,* G. Nijssen, Ed. North-Holland, Amsterdam, 1977, pp. 1–30.
4. Carlis, J.V., and March, S.T. Computer-aided physical database design

methodology. *Comput. Performance 4,* 4 (Dec. 1983), 198–214.
5. Chen, P. The entity-relationship model—Toward a unified view of data. *ACM Trans. Database Syst. 1,* 1 (March 1976), 9–36.
6. Davis, G.B. Strategies for information requirement determination. *IBM Syst. J. 21,* 1 (Jan. 1982), 4–30.
7. Everest, G.C. ER modeling versus binary modeling. In *Proceedings of the 16th International Conference on E-R Approach,* S.T. March, Ed. North-Holland, Amsterdam, 1988, pp. 63–78.
8. Falkenberg, E. Concepts for modelling information. In *Modelling in Data Base Management Systems,* G. Nijssen, Ed. North-Holland, Amsterdam, 1976, pp. 95–109.
9. Hull, R., and King, R. Semantic database modelling: Survey, applications, and research issues. *ACM Comput. Surv. 19,* 3 (Sept. 1987), 201–260.
10. *ISO/TC Concepts and Terminology for the Conceptual Schema and the Information Base,* J.J. van Griethuysen, Ed. Report of ISO/TC97/SC5/WG3, March 1982.
11. Jarvenpaa, S., and Machesky, J. End user learning behavior in data analysis and data modeling tools. In *Proceedings of the 7th International Conference on Information Systems* (San Diego, Calif.,). 1986, pp. 152–167.
12. Juhn, S., and Naumann, J. The effectiveness of data representation characteristics on user validation. In *Proceedings of the 6th Int. Conf. on Information Systems* (Indianapolis, Ind.,). 1985, pp. 212–226.
13. Kent, W. Fact-based data analysis and design. In *Entity-Relationship Approach to Software Engineering,* C. Davis et al., Eds. North-Holland, 1983, pp. 3–53.
14. Leitheiser, R. An examination of the effects of alternative schema descriptions on the understanding of database structure and the use of a query language. Ph.D. dissertation, Univ. of Minnesota, Minneapolis, 1988.
15. McGee, W. On user criteria for data model evaluation. *ACM Trans. Database Syst. 1,* 4 (Dec. 1976), 370–387.
16. Nijssen, G. Current issues in conceptual schema concepts. In *Architecture and Models in Data Base Management Systems,* G. Nijssen, Ed. North-Holland, Amsterdam, 1977.
17. Peckham, J., and Maryanski, F. Semantic data models. *ACM Comput. Surv. 20,* 3 (Sept. 1988), 153–189.
18. Ridjanovic, D. Comparing quality of data representations produced by nonexperts using logical data structures and relational data models. Ph.D. dissertation, Univ. of Minnesota, Minneapolis, 1986.
19. Senko, M.E. NIAM as a detailed example of the ANSI SPARC architecture. In *Modelling in Data Base Management Systems,* G. Nijssen, Ed. North-Holland, 1976, pp. 73–94.
20. Shneiderman, B. *Software Psychology: Human Factors in Computer and Information Systems.* Winthrop, Cambridge Mass., 1980, pp. 39–67.
21. Shoval, P., and Even-Chaime, M. Database schema design: An experimental comparison between normalization and information analysis. *Database 18,* 3 (Spring 1987), 30–39.
22. Tauzovich, B. An expert system for conceptual data modeling. In *Proceedings of the 8th International Conference on Entity-Relationship Approach* (Toronto, Canada, Oct.). Cognos Inc., Ottawa, Canada, 1989.
23. Teorey, T., Yang, D., and Fry, J. A logical design methodology for relational databases using the extended entity-relationship model. *ACM Comput. Surv. 18,* 2 (June 1986), 197–222.
24. Verheijen, G., and Van Bekkum, J. NIAM: An information analysis method. In *Information Systems Design Methodologies: A Comparative Review,* Olle, T.W., et al., Eds. North-Holland, Amsterdam, 1982, pp. 537–589.
25. Weber, R., and Zhang Y. An ontological evaluation of NIAM's grammar for conceptual schema diagrams. In *Proceedings of the 12th International Conference on Information Systems* (New York, N.Y.,), 1991, pp. 75–82.

About the Authors:

YOUNG-GUL KIM is an assistant professor of MIS at the Korea Advanced Institute of Science and Technology (KAIST) in Seoul. His research interests include information systems architecture, data and process modeling, and computer-aided software engineering. **Author's Present Address:** Department of MIS, KAIST, P.O. Box 201, Cheong-Ryang, Seoul, 130-650, Korea, email: ygkim@msd.kaist.ac.kr.edu

SALVATORE T. MARCH is a professor in the Information and Decision Science Department at the University of Minnesota. His current research interests are in system representations, database design, information system development, and distributed system design. **Author's Present Address:** Information and Decision Science Department. C. L. Carlson School of Management, University of Minnesota, 271 19th Ave. S., Minneapolis, MN 55455, email: smarch@csom.umn.edu

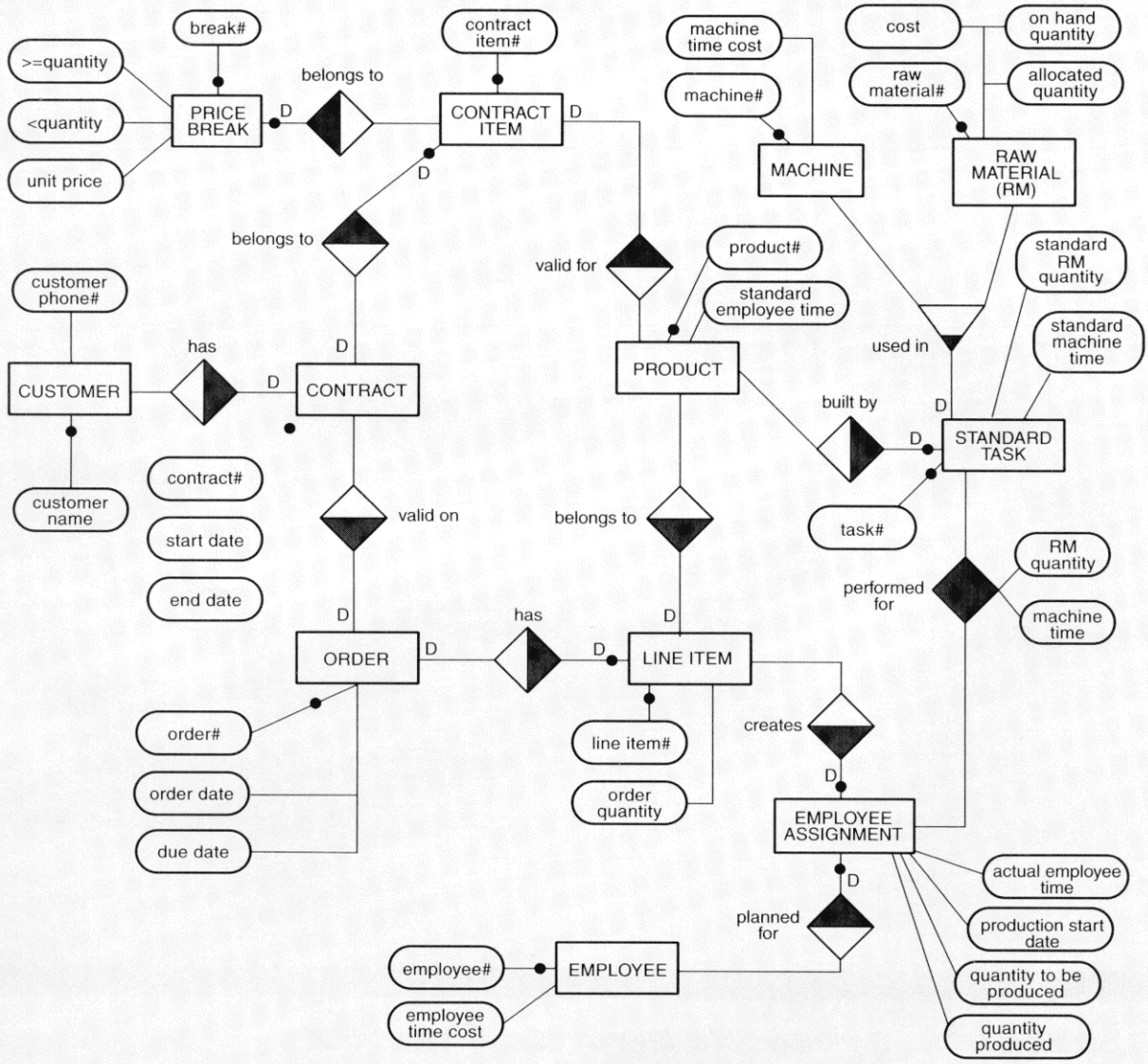
software **reuse**

Continued from page 87

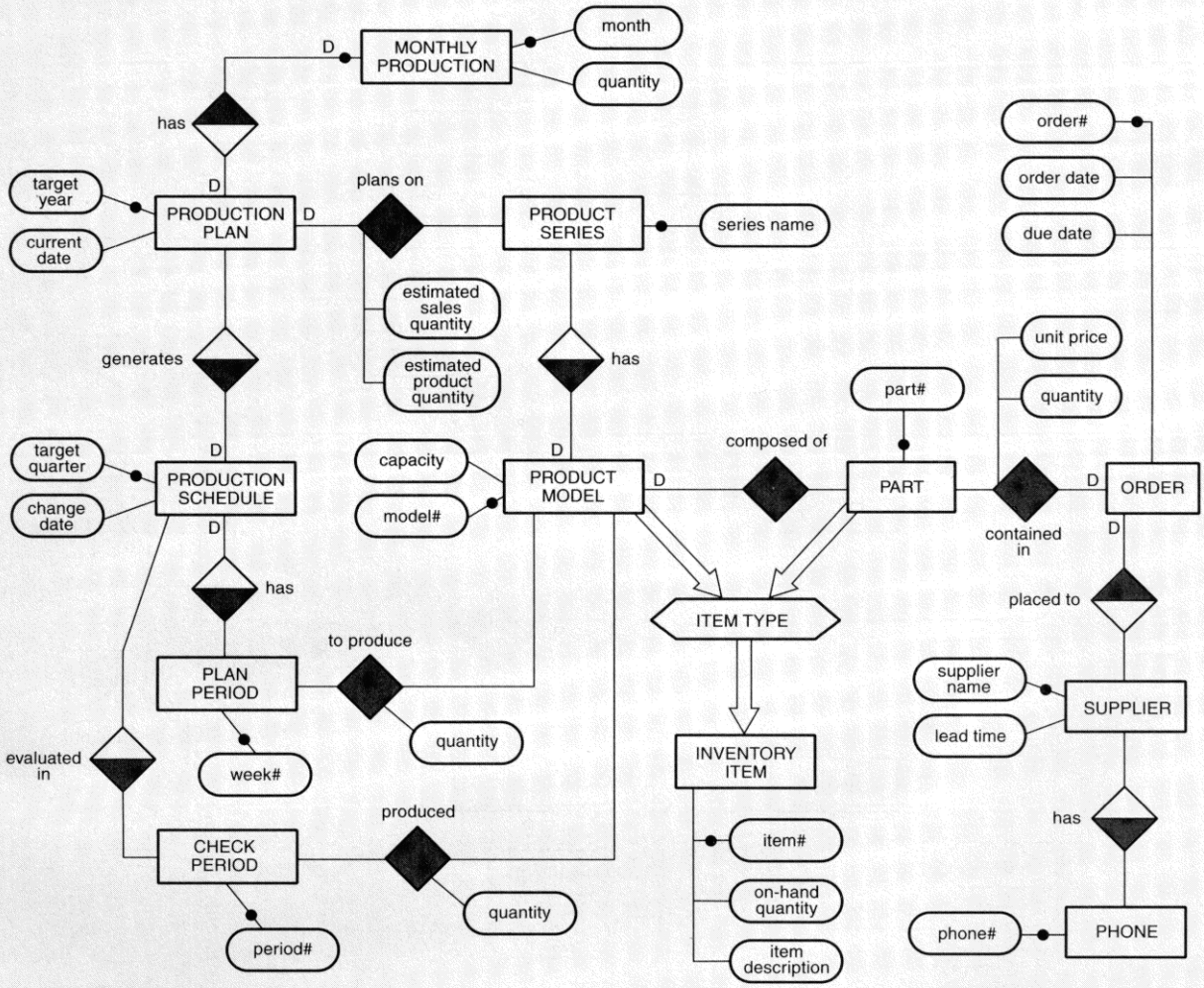
ware Reuse Research Group at the Software Productivity Consortium, and supervisor of the Intelligent Systems Research Group at AT&T Bell Laboratories. He edits *ReNews* and *ACM SIGIR Forum.* **Author's Present Address:** Department of Computer Science, Virginia Tech, 2990 Telestar Ct., Falls Church, VA 22042; email: frakes@sarvis.cs.vt.edu

CHRISTOPHER J. FOX is an associate professor of computer science at James Madison University. He has held positions as a Distinguished Member of the Technical Staff at AT&T Bell Laboratories, vice-president of engineering at Patent Search Systems Incorporated, Microcomputer Development Manager for Lockheed Dialog Information Services, and is also a member of the Editorial Panel of *Communications of the ACM.* Current interests include software reuse, software design, and software engineering education. **Author's Present Address:** Department of Computer Science, James Madison University, Harrisonburg, VA 22807; email: fox@mutt.cs.jmu.edu

Appendix A. User comprehension task



Appendix B. User discrepancy/checking task



Appendix C. Syntactic Error Categories

Entity (NOLOT):

Major
1. No name
2. No attributes/LOTs
3. No identifier

Minor
1. Duplicate entity/NOLOT names
2. Non-noun entity NOLOT names
3. Incorrect symbol

Attribute (LOT and Bridge):

Major
1. Repeating group—plural name (in EER)
2. Use of an existing entity/NOLOT name

Minor
1. Duplicate names within an entity/NOLOT or a relationship
2. Non-noun attribute/LOT names
3. Incorrect symbol

Relationship (Idea):

Major
1. No relationship/role names
2. No cardinality symbol

Minor
1. More than a binary relationship (in NIAM)
2. Attributes LOTs/present (in NIAM, also in EER unless M:N)
3. Partial role names (in NIAM)
4. Incorrect symbol

Generalization:

Major
1. Incorrect inheritance

Minor
1. Wrong symbol

Appendix D. Semantic Error Categories

Entity (NOLOT):

Major
1. Missing entities/NOLOTs

Minor
1. Incorrect extra entities/NOLOTs
2. Representation as an attribute/LOT (except for phone, week, and period)

Attribute (LOT, Bridge):

Major
1. Missing attributes/LOTs

Minor
1. Incorrect extra attributes/LOTs
2. Incorrect cardinality
3. Belonging to a wrong entity/NOLOT or relationship/role

Relationship (Role):

Major
1. Missing relationships/role-pairs

Minor
1. Wrong/incomplete/missing name
2. Incorrect or missing cardinality
3. Incorrect extra relationships/role-pairs
4. Incorrect degree
5. Redundant relationships/role-pairs
6. Connection of improper entities/NOLOTs

Generalization:

Major
1. Missing

Minor
1. Incorrect inheritance
2. Redundant hierarchy

Identifier:

Major
1. Missing

Minor
1. Incorrect

Dependency:

Major
1. Missing or incorrect