ONTOLOGICAL MODELING RULES FOR UML: AN EMPIRICAL ASSESSMENT
Joerg Evermann; Yair Wand
The Journal of Computer Information Systems; 2006; 46, 5; ABI/INFORM Global
pg. 14

ONTOLOGICAL MODELING RULES FOR UML: AN EMPIRICAL ASSESSMENT

JOERG EVERMANN
Victoria University of Wellington
Wellington, New Zeland

YAIR WAND
The University of British Columbia
Vancouver, BC, V6T 1Z2

ABSTRACT

The Unified Modeling Language (UML) is widely accepted as the de-facto standard for object-oriented information systems (IS) design and software modeling. Recent research has proposed to extend the use of UML to conceptual modeling of application domains. Conceptual models serve both as the basis for communication and domain understanding among analysts, and as the starting point for IS software design. Prior research has proposed a set of modeling rules to provide the analyst or modeler with guidelines for using UML for conceptual modeling.

This paper provides an empirical assessment of the benefits derived from using these modeling rules. Using an experimental study involving 53 subjects, it examines the effects of different levels of modeling rule application on application domain understanding. The results show statistically significant differences in the level of domain understanding that was derived from the various models.

Keywords: Analysis, Methodologies, Specification, Object-oriented design methods, Design Concepts, Ontology

INTRODUCTION

IS development begins by developing an understanding of the application domain. A common technique to support this understanding is the use of conceptual models (39). "Conceptual modeling is the activity of formally describing some aspects of the physical and social world around us for purposes of *understanding* and communication" (45, emphasis added).

Recent research (18, 20, 30, 48, 49, 52) has proposed extending the use of software description languages to conceptual modeling. Software description languages are those whose primary motivation is to describe software artifacts (such as UML and its precursors, (Dataflow Diagrams (14, 23), Jackson's Structured Programming (JSP) diagrams (36), and Yourdon's Structure Charts (69)). While such languages have clear meaning in the software domain, they lack application domain semantics. For example, it is unclear what language constructs such as 'object,' 'class,' 'attribute,' 'operation' refer to in the application domain. To attach application domain semantics, language constructs should be mapped to elements of the application domain (31). Hence, we must first specify what exists in an application domain. For this, we turn to ontology, which is the branch of philosophy that deals with what exists in the world (2). The work described in this paper is based on the assumption that mapping language constructs to ontological concepts attaches application domain semantics to a language (68).

Siau and Cao have analysed UML and concluded it is two to eleven times more complex than other languages (58). Further research by Siau and colleagues (61, 62, 63) using cognitive methodologies confirms this complexity and shows potential problems with learnability and ease of use. However, Siau et al. have demonstrated that this complexity does not always translate into usage problems (59). Modeling support in the form of rules or guidelines can facilitate the use of complex modeling languages by limiting the number of permitted combinations of model constructs. This may address the UML deficits in the area of clear semantics, shown by Siau and Loo (61). In this vein, Evermann and Wand describe a process to derive modeling rules for using software description languages in conceptual modeling, based on existing ontological mappings (21). Such modeling rules have been shown useful in a medium-size case study (20), but their benefits have not been tested in controlled experiments. This paper describes an experimental study of the use of ontologically derived modeling rules for UML.

The remainder of the paper is structured as follows. The next section briefly describes the chosen ontology and the UML modeling rules. This is followed by the theoretical model and the experimental hypothesis. Following this, the paper describes the development of the experimental instrument and the results of the pilot test. The results of the full study are then presented, followed by a discussion of the results and a conclusion. Two appendices include a list of modeling rules used in the study (Appendix I) and the UML models used (Appendix II).

ONTOLOGY AND MODELING RULES

The most widely used ontology in IS research is that of Mario Bunge (7, 8). This ontology has been chosen due to its agreement with the natural sciences, its generality, its axiomatic development, its formalization in terms of set theory and logic, and because predictions based on its application to IS have been tested empirically (5, 24, 25). Table 1 provides a very brief description of the main concepts that are postulated by this ontology, as relevant to this research.

Mappings of UML to Bunge's ontology have been widely researched (16, 17, 18, 19, 20, 49). Based on these mappings, modeling rules for using UML in conceptual modeling have been developed (18, 19, 20, 21) (Appendix I). The modeling rules were derived by translating ontological premises into constraints on the allowed combination of language constructs (20). To demonstrate, we briefly discuss two examples of modeling rules (further examples are described in (19, 21)).

Evermann (17), and Evermann and Wand (18, 19, 20) map Bunge's things to UML objects and Bunge's properties to UML attributes and vice versa. They define no direct mapping for the UML association construct. Instead, they refine the property↔attribute mapping as follows: Intrinsic properties are mapped to attributes of classes, while mutual properties are mapped to attributes of association classes. Therefore, while associations themselves have no ontological semantics, association classes represent sets of mutual properties (arising out of the same interaction) and association class attributes represent mutual properties. This argument leads to rules 5 and 13 and corollary 3.[1] Since sets of properties cannot themselves possess properties, association classes cannot be associated with other classes (corollary 4). Finally, sets of properties cannot be generalized (corollary 5).

TABLE 1
Concepts of Mario Bunge's Ontology (7, 8) Relevant to This Study

Concept	Description
Thing	The world is made of things that exist physically.
Property	Things possess properties. Properties are either intrinsic (belonging to a single thing), or mutual (belonging to two or more things). Mutual properties can be either binding (related to interaction of things), or non-binding. Properties cannot possess other properties and cannot be generalized. Properties can be constrained by laws, in particular a law can be described in terms of property precedence: Property B precedes property A, if every thing possessing property A also possesses property B.
State	Things can be modeled by state functions that represent properties of the things. The values of all functions in a particular model define the state of the thing with respect to the particular model.
Action	A thing x acts on another thing y, if the state history of y depends on the existence of x. Two things interact, if each acts on the other.

Rule 5: *An association class represents a set of mutual properties arising out of the same interaction.*
Rule 13: *Every ordinary association (i.e. not aggregations or compositions) must be an association class.*
Corollary 3: *An association class must possess at least one attribute.*
Corollary 4: *An association class must not be associated with another class.*
Corollary 5: *An association class must not participate in generalization relationships.*

A case study has shown these rules and guidelines to be usable and applicable to conceptual modeling (20). Independent of the study, project team members had prepared UML models of the university student admission domain at a large North American university. A second model independently created by the researchers following the proposed rules showed that using the rules was feasible. Discussions of both models with project team members (who were not told of the rules) showed that the resulting models were sensible. Team members commented that the rules-guided models led to explication of hidden assumptions: "Ours have all sorts of stuff around that is assumed but not modeled." The rule-compliant model was also perceived to be more comprehensive: "Certainly more comprehensive here, but even in the smaller, there's a somewhat simpler, more elegant view in a few cases." The model was felt to be suitable for further IS design: "I don't see any reason why you couldn't just take these and run with them." After project team members expressed their opinions about the models, the rules were disclosed to them. Members agreed that the rules would have been helpful: "Such rules would have helped in our group," "Rules can force the modelers to think deeper about what they're modeling."

While the rule set may appear complex, Lu and Parsons (41) describe the implementation of a CASE tool that incorporates the rules and conclude that the rule set is operationalizable and usable in practice.

THEORETICAL DEVELOPMENT

Kung and Solvberg (39) suggest that conceptual models support both domain *understanding* and *communication* about a domain. This paper focuses on domain understanding, an important factor in reducing risks to system implementation (47), facilitating requirements engineering and elicitation (35, 37), and reducing costly rework later in the development process (6). Understanding goes beyond communication, examined in a recent paper by Parsons and Cole (51). Dobing and Parsons (15) show that this distinction is made in practice: Different UML diagrams are used to different extents for client verification (communication) and for clarifying analyst's understanding of the problem domain. This emphasis on understanding is reflected by Mylopoulos: "The adequacy of a conceptual modeling notation rests on its contribution to the construction of models of reality that promote a common *understanding* of that reality among their human users." (45, pg. 51, emphasis added).

A single UML model may encompass several diagrams that may be of different types. The diagram types most often used are class and use case diagrams (15). Class diagrams are used primarily for purposes of understanding (by system developers), while use case diagrams are used primarily for purposes of communication with clients (15). As the focus of this paper is on understanding, we focus on class diagrams. Furthermore, the description of object classes is required for all other diagram types. For example, states are states of objects, and are exchanged between objects classes.

While it is often assumed that class diagrams need to be complemented with a diagram showing dynamic aspects, such as use case diagrams, Siau and Lee found that their combination has no influence on completeness of domain interpretation (60). Moreover, examining multiple diagrams in a single study introduces many variables that must be controlled: Switching behaviour between diagrams (34), and the effects of visual cues linking different diagrams on cognitive integration (38); Partial diagrams of a global model have been found easier to verify than single, global diagrams (50), perhaps due to the negative effect of the number of concepts in a diagram on diagram readability (3). For these reasons we focus on class diagrams only.

Examining model construction and interpretation together makes it difficult to separate their relative effects (25, 29). This study therefore focuses on model interpretation. In terms of Gemino and Wand's framework (29), we are interested in the effectiveness of the product of the script interpretation task. We do not discuss effects of learning or learnability of ontological rules. Also, we do not discuss effects due to the creator and interpreter having an agreed upon set of rules. In fact, participants in the study to be described were not aware of the modeling rules, or even the fact that the diagrams were rule compliant.

When examining knowledge representation forms and their cognitive effects, we need to fill in the 'missing link' (29, 56, 66) of relating conformance to an external ontology to cognitive performance, such as understanding. Gemino and Wand argue that, "since ontology is intended to formalize the way the real world is modeled by humans, the ontological constructs might also be viewed as related to constructs in human cognition" (29, pg. 255). They further suggest, based on work by Mayer (42), that understanding the information contained in a conceptual model requires the integration of this information into existing knowledge (26). As model understanding occurs only with partial models (34, 38, 50), cognitive effort must be expended on their integration. When the presentation of the model is of a form corresponding to cognitive structures, this integration requires less extraneous cognitive load and hence can occur faster and more effectively (26).

The semantic network model of human cognitive structures (10, 11, 12) supports the above argument. In this model, relationship edges connect concept vertices. As an ontology is a set of concepts and their relationships, it constitutes part of a semantic network. Understanding of information is the integration of new concepts and relationships into this network. If the conceptual model conforms to the same ontology (by complying with modeling rules derived from the ontology), it will correspond well to the interpreter's semantic network. This will facilitate integration. If the model contradicts or is incompatible with the interpreter's semantic network, it may be integrated improperly, inefficiently, or not at all.

Both Mayer's theory and the semantic network model of cognition are well accepted for experimental studies on conceptual modeling languages (5, 9, 24, 25, 28, 29). Based on this discussion, we advance the following proposition:

Proposition: Interpreting application domain models conforming to ontologically derived rules leads to increased domain understanding, compared to non-conforming models.

To operationalize domain understanding, we again look to Mayer (42) who suggests that understanding is the outcome of a learning process and manifests itself in problem solving ability. This operationalization has found empirical support in conceptual modeling research in the studies of Bodart et al. (5), Burton-Jones (9), Gemino (24), and Gemino & Wand (25, 27, 28, 29). It is also supported by cognitive research on problem solving. Problem solving requires inferences based on existing knowledge, i.e. reasoning (1, 46). Problem solving can be used as a proxy measure of the amount of reasoning that a given semantic network supports. It can thus be used to measure domain understanding (25, 28, 29). When the interpreted model is well integrated into the existing mental structures, it is possible for the interpreter to use pre-existing relationship edges between concept vertices for reasoning. In the IS literature, this argument has been taken up by Siau (57). Also, based on Newell & Simon's work on problem solving (46), Siau argues that some diagrams are computationally more efficient than others, even though they are informationally equivalent to the others. Computational efficiency is the ability to make *inferences* based on the information presented in a diagram (57). Siau's notion of computational efficiency is therefore closely related to our problem solving ability, and thus to learning and domain understanding. If we assume that the ontology used conforms to the way people view the world, we derive the following hypothesis:

Hypothesis: Learning about a domain by interpreting diagrams conforming to ontologically derived modeling rules will lead to better performance in *problem solving tasks* than interpreting non-conforming diagrams.

Parsons and Cole suggest that research into the more immediate effect of conceptual models on communication is still required, before moving to their slightly less immediate effect on understanding (51). Accordingly, they propose that, when evaluating conceptual models for the purpose of communication, the dependent variable should be the performance with respect to information contained in the script, e.g. recall as a proxy for communicative ability. While we agree with this argument in the case of communication, we have presented a well tested model of learning and cognitive knowledge organization. It can be, and has been, used to test effects of conceptual models that go beyond information delivery and include learning and understanding. As the focus of this paper is on understanding, rather than communication, we feel justified in using problem solving rather than recall measures, as suggested by Parsons and Cole (51).

To summarize in terms of the framework by Gemino and Wand (29), the examined constructs are those of UML class diagrams, the scope is a single grammar (UML), the nature of the study is an intra-grammar comparison[2] and the use of the grammar is varied, i.e. different levels of modeling rule conformance are tested. Content delivery is graphical, and the content material is drawn from multiple domains. The task is model interpretation. As discussed above, the focus of observation is the product of the task and the criterion of comparison is the effectiveness (i.e. domain understanding).

INSTRUMENT DEVELOPMENT

Independent/Affecting Variable

The study investigates three diagrams in each of two domains, which differ in their level of rule conformance. An example of the rules was presented above. The complete set of rules used for this study, together with application examples, can be found in (17, 20) and in Appendix I. The car-rental example from (17) is re-used here as one of the domains investigated.

The first diagram (factor level "N", No rules) for each domain was selected from textbooks (22, 43) to ensure that it reflects current practice in UML use. We selected models that are described as reflecting the domain or as being analysis level models. These models are not intended to be used as software or database designs. The selected models are shown in Appendix II as Figs. 3 and 4 respectively. Four independent judges produced a list of rule violations for each model. Two further models are based on an ontological re-analysis of the two domains and constructed to satisfy all applicable rules (factor level "R," Rule conforming, Appendix II, Figs. 5, 6).

Some rules force the introduction of model elements not in the original model. For example, rule 5 and the corollaries discussed above force the modeler to introduce association class attributes representing domain elements not represented by the original model. Therefore, a third experimental condition ("R2," partial rule conformance) was created. For this condition, the model for the second condition was modified by transforming association classes to classes participating in the association or by moving all of their attributes to participating classes (Appendix II, Figs. 7, 8). This violates only rule 5, allowing us to examine the effects of a single rule violation, and does not add or remove information, guaranteeing informational equivalence of models.

Models for all three conditions were judged by four independent judges and modified until the judges unanimously agreed on their level of rule conformance.

To minimize effects due to the choice of application domain, Parsons and Cole (50, 51) suggest that diagrams be evaluated in an artificial domain, and that subject matter experts not be used in order to not confound their existing background knowledge with that conveyed by the diagram. While this is appropriate when investigating *communication* by means of recall, it is inappropriate when investigating *understanding*. As described above, theories of human cognition suggest that understanding is the integration of new concepts into the existing cognitive network. Abstract domains cannot be integrated as they have no overlap with the existing cognitive network.

Furthermore, Siau et al. have shown that "when structural constraints were given and presented in a way that contradicted the surface semantics,[3] almost all the subjects based their interpretation on the structural constraints and ignored the surface semantics." (64, pg. 162). This indicates that model content and familiarity play a subordinate role in interpretation.

Finally, Parsons and Cole claim that "participants are using knowledge primed by the words in the diagram, rather than the semantics conveyed by the diagram's structure, to answer the questions" (51, pg. 337f). However, from their study one could equally well conclude that the presence of known concepts in the diagram facilitated learning and thus increased the problem solving scores. The only way in which one can conclusively show that background knowledge is used directly to answer questions, is to control for it in different conditions of subject/domain combinations, which was not done by Parsons and Cole.

The effect studied in this research deals with integrating information contained in diagrams into existing knowledge, i.e. *understanding*, rather than the communicative capacity of diagrams. While the argument of Parsons and Cole is valid for research into *communication*, for our focus on *understanding*, we address the issue of potential domain biases by using multiple domains (car rental "CR," order processing "OP"), chosen to be familiar to subjects (to enable integration into semantic networks), but without subjects being experts (to minimize reliance on existing knowledge). The use of multiple domains mitigates the potential of domain-specific results.

Dependent Variable

The instrument for problem solving ability was taken from Gemino & Wand (25) and Bodart et al. (5) and changed to reflect the chosen task domains. It is a set of open-ended problem solving questions that cannot be directly answered only with the information given in the diagram, but rather require deeper domain understanding. A set of comprehension questions administered prior to problem solving ensures that subjects examine all aspects of the model. In contrast to the recommendation by Gemino (24) and Gemino and Wand (25, 27, 28, 29), they are not used to establish informational equivalence.[4]

The adapted instrument was revalidated first by a graduate student in MIS, experienced in information modeling, who examined the problem solving questions and the models to ensure that the information in the model was insufficient to answer the questions. Subjects of a pilot test (described below) were found to provide a wide variety of correct as well as incorrect answers. This indicated that a single correct answer could not be deduced from the models. Table 2 shows examples of the problem solving questions for the two domains. After validation, seven questions remained for the order processing domain (OPProb) and five for the car rental domain (CRProb).

TABLE 2
Example Problem Solving Questions

OPProb-1	Suppose that an important customer needs to order products urgently. What problems could he/she face?
OPProb-2	Suppose that a shipment does not contain all ordered products. What could have happened?
OPProb-3	Suppose the warehouse has run out of items and an order cannot be fulfilled. How could this have happened?
CRProb-1	Suppose that a customer arrives for pick-up but no car is available. What could have happened?
CRProb-2	Suppose a customer receives two invoices. What could have happened?
CRProb-3	Suppose a car which is reserved for a customer is being sold at auction. How could this have happened?

Control Variables

Informational Equivalence. While informational equivalence is usually argued to be a pre-requisite of comparing diagrams, the nature of the rules requires informational non-equivalence.[5] For example, rule 5 requires the modeler to include association classes and their attributes.[6] We address this issue in two ways.

First, we argue that information shown in a diagram influences problem solving only when it is *useful* for that task. To this effect, we include measures for the usefulness of information provided by both diagrams.

Second, the third experimental condition, in which a specific rule is violated, was created from the fully rule-conforming diagram in a way that maintained all information in the model. This provides a better control than comprehension questions, which cannot realistically cover all model elements, as there are more than 50 elements in each model.

The effect studied here relates to how well the information contained in the model can be integrated with existing information to enable inferences about the domain. This can be interpreted as the diagrams being *computationally* non-equivalent.

Usefulness and Ease of Interpretation. The instruments for usefulness of information (USE) and ease of interpretation (EOI) were developed from existing instruments (4, 13, 25, 40, 44), which are reported as highly reliable. Table 3 shows the initial item set.

TABLE 3
Ease of Use and Usefulness (R: reverse coded, *: item dropped from final version after pilot-testing showed difficulty with convergent or discriminant validity).

EOI-1	The class diagram was easy to read.
EOI-2	The class diagram was clear and understandable.
EOI-3	The class diagram was easy to understand.
EOI-4 (R)(*)	Interpreting the class diagram was frustrating.
EOI-5 (*)	The class diagram was simple.
EOI-6 (R)	I believe that the class diagram was cumbersome to interpret.
EOI-7	I feel comfortable with interpreting the class diagram.
EOI-8	I believe that it is easy to interpret the class diagram.
EOI-9 (R)	Interpreting the class diagram required a lot of mental effort.
EOI-10	Overall, I found the class diagram easy to interpret.
EOI-11	It was easy for me to understand what the diagram was trying to model.
EOI-12 (*)	Overall, I found the class diagram was easy to use.
USE-1 (*)	I feel confident that my answers are correct.
USE-2	The information in the class diagram was helpful for answering the questions.
USE-3	Interpreting the class diagram increased my effectiveness in answering the questions.
USE-4	Interpreting the class diagram improved the quality of my answers to the questions.
USE-5	Understanding the class diagram made answering the questions easier.
USE-6	Interpreting the diagram enhanced the quality of my answers.

UML Knowledge. UML knowledge was assessed by a self-reported measure on a 7-point Likert scale (UML.A) and by a self-reported estimate of the number of months of UML usage (UML.B). As self-reported measures, both items may be problematic (32). Therefore, a set of 19 binary response and multiple-choice questions were added from the examination software to a system analysis textbook (53) (the sum of correct responses to these forms the variable UML.TTL).

Domain Knowledge. Task domain knowledge was assessed by a self-reported measure on a 7-point Likert scale ("CR-1" for the car rental domain, "OP-1" for the order processing domain). Subjects were also asked whether they had worked in a car-rental company or order processing department ("CR-2", "OP-2") and whether they had previously rented a car ("CR-3").

Pilot Test

A pilot test with 14 subjects was conducted using preliminary versions of the UML models. Each subject was presented with both domains in random order and a post-test questionnaire was administered after each domain. Scale reliabilities were excellent for the Ease of Interpretation (Cronbach-α= .9366) and Usefulness (Cronbach- α = .9121) constructs. A factor analysis showed that a two-factor model of the combined instrument explained 67.8% of the variance. Both Usefulness and Ease of Interpretation showed good convergent and discriminant validity with clear factor loadings. Due to the small sample, the pilot-test data is indicative only.

EXPERIMENTAL PRECEDURE AND RESULTS

Subjects

As discussed above, a certain level of domain knowledge is necessary to enable learning and understanding. The subjects were drawn from three groups of fourth year undergraduate students. One group of subjects was drawn from a business school course in system analysis (factor level "3," n=14), the other two were drawn from two sections of a computer science system analysis course (factor level "C," n=26 and factor level "C2," n=13). Subjects were recruited by in-class advertising and received a monetary incentive for participation. While the use of student subjects may be problematic in managerial decision-making contexts (33), the task in this study is cognitively more basic and less demanding of experience. Hence, performance characteristics on the present tasks are likely more universal.

Procedure

The experimental design included a within-subjects factor, as the number of subjects that could be gained was not very large. Because carry-over and domain learning effects may arise when rule-conformance is chosen as within-subjects factor, we have chosen the application domain (factor "Domain," with levels "CR," car rental, and "OP," order-processing) as within-subjects factor. Subjects were randomly assigned to one of three rule conformance conditions (factor "Rules" with levels "N," "R," and "R2"). The order of the domains was varied randomly to minimize carry-over or learning effects.

Since it is possible that subjects may attempt to solve problems based on information in the diagram, or not provide answers if they cannot find the information in the diagram, the diagram was removed after the administration of the comprehension questions, prior to administering the problem solving questions. This is in agreement with the suggestion by Gemino to remove the diagram (24), but in contrast to the suggestion by Parsons and Cole not to remove the diagrams (51). While this is appropriate to assess *communication* aspects of conceptual models, our study requires that the effect of conceptual models on *understanding* not be confounded or overshadowed by their communicative ability. Removing the diagram ensures that the communication function of the model is completed and problem solving is based entirely on domain understanding gained from it. Finally, the post-test questionnaire for control variables was administered.

Statistical Results

Scale Reliabilities and Validities. The analysis shows excellent scale reliabilities for the Ease of Interpretation (a_{CR}=.9456, a_{OP}=.9234) and Usefulness (a_{CR}=.9176, a_{OP}=.9243) measures. Convergent and discriminant validity were assessed using factor analysis by computing a two-factor solution. Theoretically, usefulness of information and ease of interpretation should be unrelated. We therefore used a varimax orthogonal factor rotation.[7] For the car rental and order processing domains, the proportion of variance explained by both factors was 0.690 and 0.666, respectively. Both factors show excellent discriminant and convergent validity with a clear factor loading structure. Cross-loadings were generally below 0.35 and on-factor loadings generally higher than 0.65.

Inter-rater Reliabilities. The problem solving questions ("Prob") were open-ended questions; subjects were asked to provide as many answers as they could think of. Hence, there were no correct answers against which to check the responses. Two PhD students in MIS with general knowledge of both domains independently rated the responses as correct or incorrect, and assessed the total number of correct responses for each question for each subject. The average inter-rater agreement (Cohen's Kappa) was 0.9138 with a minimum of 0.7872 and a maximum of 1.000, showing excellent agreement between the two raters.

Hypothesis Testing. Figure 1 shows the interaction of rule conformance and domain on problem solving performance, measured by the average number of correct answers for the problem solving questions. Both domains follow the same pattern. Figure 2 shows the interaction of rule conformance and subject group. The first group of computer science subjects did not follow the pattern of the business students and the second group of computer science students.

These visually suggested effects were assessed statistically by an ANCOVA procedure, including the factors "Rules," "Group" and "Domain," as well as their first order interactions, and further including the following continuous covariates without interaction effects (Table 4):[8]

- Total score on the UML assessment questions
- Self-assessed UML knowledge (7-Point Likert)
- Self-assessed domain knowledge (7-point Likert)
- Time taken for problem solving (Time)
- Average comprehension score (Comp)
- Ease of Interpretation factor scores
- Usefulness factor scores

There were significant main and interaction effects of rule conformance and subject group ($p < .05$). The goodness of fit of the ANCOVA model was R^2=.4505, suggesting adequate explanatory power. The estimated improvement in problem solving performance for the rule conforming diagrams was 0.388 (average correct answers per question). This represents a 26% increase in problem solving performance. While the effect of UML knowledge (UML.TTL) on problem solving performance was statistically significant, the small magnitude of the parameter (.089) made it practically irrelevant.

Equivalence of Diagrams. No effect of rule conformance or domain on usefulness was found. This justifies the assumption of informational equivalence of the diagrams for the purposes of the given problem-solving task. Moreover, by the method of diagram construction, we guaranteed informational equivalence between conditions "R" and "R2" (described above).

DISCUSSION

Overall, the results of the experiment confirm the hypothesis and underline the benefits of the ontologically derived modeling rules. We conclude that conceptual modeling guided by ontologically derived rules can lead to a practically significant increase in domain understanding.

The significant difference between the performance on diagrams that completely conform to ontological rules ("R") and diagrams that conform to all but one rule ("R2") cannot be explained by diagram complexity or readability (38). Diagrams in both conditions use an equal number of constructs. The difference indicates that benefits may not be linear in the level of rule conformance, i.e. there may be certain threshold levels of rule conformance. In terms of the cognitive network theory, this might suggest that the model may have to match the mental network in certain important aspects in order to be properly understood. This opens up interesting future research on the relative importance of the rules and the constrained language constructs. For example, the critical rule violated in the third experimental condition is related to associations. Associations are recognized as a central and important language construct (65, 68). The importance of the construct may suggest the rule is also important and hence its violoation might lead to pronounced effects.

UML knowledge, as assessed by the multiple-choice questions, plays a partial role in explaining the derived benefits, but self-assessed UML knowledge does not have any effect. This suggests the two may be different and justifies inclusion of both measures of modeling knowledge.

No effect of rule conformance on the usefulness of information could be found, confirming *task-relative* information equivalence. However, there was an effect of rule conformance on ease of interpretation ($p < 0.01$): Diagrams fully conforming to the rules (R) were easier to interpret than diagrams partially conforming to the rules (R2). The latter were in turn easier to interpret than rule non-conforming diagrams (N). This result confirms the theoretical predictions of this research, namely, that fully conforming diagrams would lead to better domain understanding because they conform to existing cognitive structures and are therefore easier to integrate into existing knowledge. The effect of rule conformance on ease of interpretation may be argued to substantiate this effect.

While differences between computer science and business student subjects may be expected, differences arose between two groups of computer science student subjects. This is all the more surprising as the two groups had received the same instruction, were taught by the same instructor, in the same context. The second group, which did not respond to differences in rule-conformance, was taken from a summer semester course. As the university does not generally offer summer courses, it is possible that these were highly motivated students aiming to complete their studies quickly. On the other hand, they may also have been low-motivation students, who repeated a failed course. A test for an effect of subject group on the time spent shows that subjects in the first group spent about 2.5 minutes less on the overall task. However, this effect did not in turn translate to a significant effect of time on problem solving performance. A post-study discussion with the instructor of the course from which the subjects were sampled also suggested motivational differences between the two groups. However, there were no differences in UML knowledge between the two groups.

Care must be taken when generalizing from the sample to the target population. The sample frame was defined to match characteristics of business analysts, possessing some IT and modeling knowledge and wide business knowledge. The two domains were also chosen to not be outside the conceivable area of knowledge of the sample. Thus, while the subjects may not have been business experts per se, the domains did not require such expertise as they should have been reasonably familiar to subjects but not to the extent that specific domain knowledge dominated the problem-solving task.

FIGURE 1
Interaction Effects of Rule Conformance and Domain on Problem Solving (Mean of Correct Answers for Problem Solving Questions)

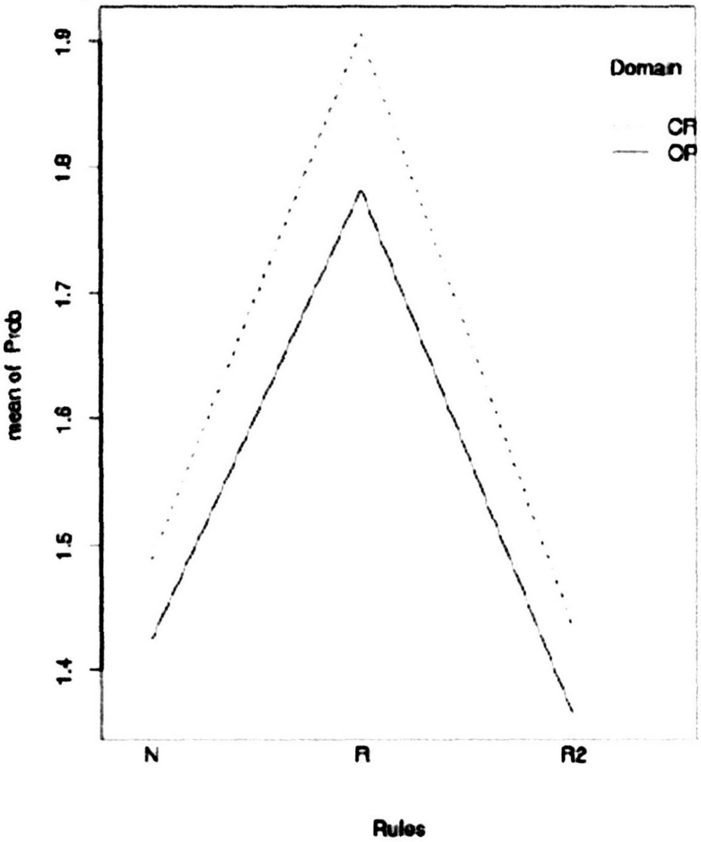
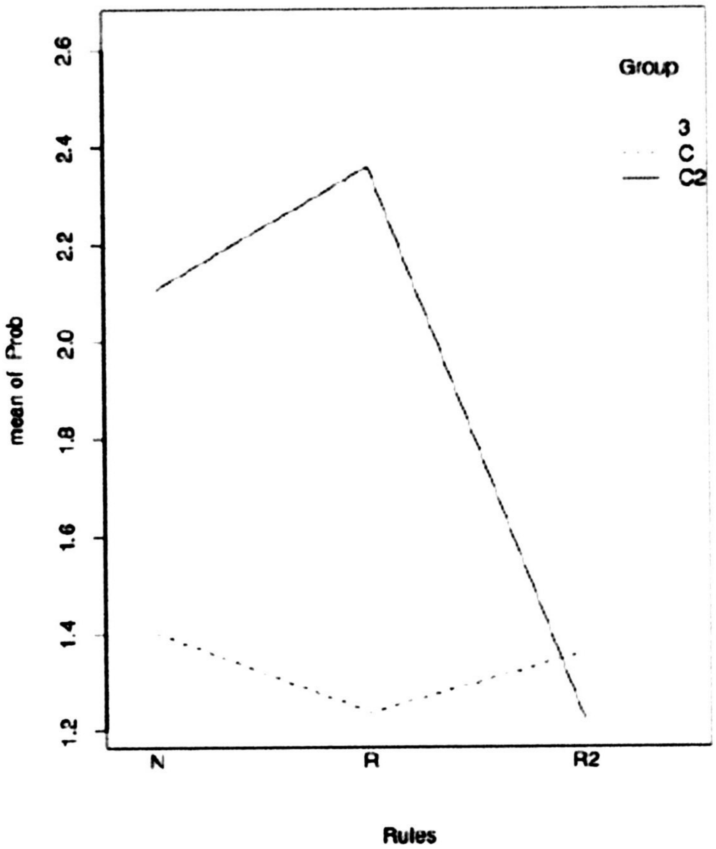


TABLE 4
Main Effects of Rule Conformance, Domain and Subject Group

	Rule Conformance				Domain			Subject Group			
	No Rules (N)	Rules (R)	Partial Rules (R2)	Results	Car Rental (CR)	Order Proc. (OP)	Results	Business Students (3)	Comp Science Students 1 (C)	Comp Science Students 2 (C2)	Results
	Means (SD) n=16	Means (SD) n=17	Means (SD) n=20	F (sig.)	Means (SD) n=53	Means (SD) n=53	F (sig.)	Means (SD) n=14	Means (SD) n=26	Means (SD) n=13	F (sig.)
Problem Solving Performance[9]	1.4558 (.6720)	1.8437 (1.003)	1.3996 (.6439)	4.0488 (.021)*	1.6019 (.8648)	1.5162 (.7388)	.4411 (.508)	1.9250 (.9406)	1.3313 (.5736)	1.6203 (.9002)	6.6711 (.002) **
Comprehension Performance[10]	0.8958 (.0964)	0.7944 (.1545)	0.9194 (.0783)	11.772 (.000)**	0.8679 (.1326)	0.8765 (.1170)	0.1490 (.7004)	0.8582 (.1263)	0.8772 (.1199)	0.8772 (.1353)	0.3716 (0.6907)

FIGURE 2
Interaction Effects of Rule Conformance and Subject Group on Problem Solving
(Mean of Correct Answers for Problem Solving Questions)



Similarly, care must also be taken when generalizing to other domains. While car rental and order processing may be typical business contexts, the models used in this study were necessarily simpler than models in real development projects, as they were derived from teaching models in textbooks. While readability is limited by model size (3), improvements in domain understanding may well scale with additional model size, and thus partially compensate for this. In this respect, Gemino and Wand (26) have discussed the trade-off between model complexity and local clarity.

CONCLUSION

Previous research has proposed semantics and rules for using software description languages for conceptual modeling. To obtain business semantics for the language constructs, they are mapped to an ontology. These semantics lead to modeling rules that constrain the possible ways to create a conceptual model of a given domain. This experimental study of a proposed set of modeling rules shows them to be beneficial in enhancing domain understanding, an important prerequisite for successful IS development.

While UML is currently the de-facto standard in object-oriented IS development, other languages are also being used for conceptual modeling. Prominent among these are ER diagrams and the ARIS language (54), both of which have been mapped to an ontology (30, 68). The apparent success of developing effective modeling rules for UML should encourage researchers, based on the existing mappings, to also develop modeling rules that can guide the business analyst in the application of these other languages.

In the wider context of modeling quality, this study has tested a set of rules intended to ensure, in terms of Schütte and Rotthowe (55), the semantic correctness or construction adequacy of the modeling language and resulting models. This is but one important quality criterion. The conflicting relationship between this criterion and issues such as model complexity and the economics of time and money spent on modeling is widely recognized. Future studies need to evaluate the set of proposed rules with respect to other quality dimensions and potential quality criteria trade-offs.

ENDNOTES

[1]Rules and corollaries are numbered here to be consistent with the complete list in Appendix I.

[2]As the construct set and their meanings are not varied, we argue this is an intra-grammar study. However, when one characterizes a grammar as the symbol set and associated wel-formedness rules, this may be an inter-grammar comparison (UML with regular well-formedness rules, UML with additional ontologically derived well-formedness rules).

[3]The names of diagram elements.

[4]As discussed above, we expect differences in information

content, due to the nature of the rules. We have objectively ensured informational equivalence between conditions "R" and "R2."

[5]Indeed, the notion of informational equivalence is not unproblematic. Informational equivalence as applied by Gemino and Wand (25, 29) differs from that applied by Siau (57), who states that "Two representations are informationally equivalent if all of the information in one is also *inferrable* from the other" (57, p. 77, emphasis added). Testing whether the information is *contained* in a diagram by means of comprehension questions is not sufficient to establish information equivalence in Siau's sense. Indeed, with Siau's understanding of informational equivalence, the problem solving tasks of Gemino and Wand (25, 29) specifically assume *non*-equivalence of information. The *inferred* information, as evidenced by problem solving capacity, is hypothesized to be different.

[6]This effect of the rules has also been confirmed in the case study where it was evaluated positively (described above). Pilot test and final study results of diagram comprehension questions also showed expected differences (below).

[7]A subsequent promax oblique solution yielded similar results.

[8]The variables UML-use (UML.B), car rental experience (CR-2, CR-3) and order processing experience (OP-2) showed no variability and were excluded from further analysis.

[9]Mean number of correct answers for problem solving questions.

[10]Mean number of correct answers for comprehension questions.

REFERENCES

1. Anderson, J. **Cognitive Psychology and Its Implications,** 4th ed. New York: Freeman, 1995
2. Angeles, P. **Dictionary of Philosophy.** New York: Harper Perennial, 1981.
3. Bajaj, A. "The Effect of the Number of Concepts on the Readability of Schemas: An Empirical Study with Data Models," **Requirements Engineering,** 9:4, 2004, pp. 261-270.
4. Batra, D., J. Hoffer, and R. Bostrom. "Comparing Representations with Relational and EER Models," **Communications of the ACM,** 33:2, 1990, pp. 126-139.
5. Bodart, F., A. Patel, M. Sim, and R. Weber. "Should Optional Properties be Used in Conceptual Modeling? A Theory and Three Empirical Tests," **Information Systems Research,** 12:4, 2001, pp. 384-405.
6. Boehm, B. "Understanding and Controlling Software Costs," **IEEE Transactions on Software Engineering,** 14:10, 1988, pp. 1462-1477.
7. Bunge, M. **Treatise on General Ontology, Ontology I – The Furniture of the World.** Dordrecht, Netherlands: D. Reidel, 1977.
8. Bunge, M. **Treatise on General Ontology, Ontology II – A World of Systems.** Dordrecht, Netherlands: D. Reidel, 1979.
9. Burton-Jones, A. and P. Meso. "How Good are these UML Diagrams: An Empirical Test of the Wand and Weber Good Decomposition Model," **Proceedings of the International Conference on Information Systems,** Barcelona, December 2002, pp. 104-114.
10. Collins, A. and E. Loftus. "A Spreading Activation Theory of Semantic Processing," **Psychological Review,** 82, 1975, pp. 407-428.
11. Collins, A. and M. Quillian. "Facilitating Retrieval from Semantic Memory: The Effect of Repeating Part of an Inference," **Acta Psychological,** 33, 1970, pp. 304-314.
12. Collins, A. and M. Quillian. "Retrieval Time from Semantic Memory," **Journal of Verbal Learning and Verbal Behavior,** 8, 1969, pp. 240-248.
13. Davis, F. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology," **MIS Quarterly,** 13, 1989, pp. 319-340.
14. DeMarco, T. **Structured Analysis and System Specification.** Englewood Cliffs, NJ: Yourdon, 1978.
15. Dobing, B. and J. Parsons. "How UML is Used," **Communications of the ACM,** 49:5, 2006, pp. 109-113.
16. Dussart, A., B. Aubert, and M. Patry. "An Evaluation of Inter-organizational Workflow Modeling Formalisms," **Journal of Database Management,** 15:2, 2004, pp. 74-104.
17. Evermann, J. "Thinking Ontologically: Conceptual versus Design Models in UML," **Ontologies and Business System Analysis.** Hershey, PA: Idea Group, 2006, pp. 82-104.
18. Evermann, J. and Y. Wand. "An Ontological Examination of Object Interaction in Conceptual Modeling," **Proceedings of the Workshop on Information Technologies and Systems WITS'01,** New Orleans, LA, December 15-16, 2001, pp. 91-96.
19. Evermann, J. and Y. Wand. "Ontology based Object-oriented Domain Modeling: Fundamental Concepts," **Requirements Engineering,** 10:2, 2005, pp. 146-160.
20. Evermann, J. and Y. Wand. "Toward Ontologically Based Semantics for UML Constructs," **Proceedings of the 20th International Conference on Conceptual Modeling,** Yokohama, Japan, November 27-30, 2001, pp. 354-367.
21. Evermann, J. and Y. Wand. "Towards Formalizing Domain Modeling Semantics in Language Syntax," **IEEE Transactions on Software Engineering,** 31:1, 2005, pp. 21-37.
22. Fowler, M. and S. Kendall. **UML Distilled: A Brief Guide to the Standard Object Oriented Modeling Language.** Reading, MA: Addison Wesley, 2000.
23. Gane, C. and T. Sarson. **Structured System Analysis: Tools and Techniques.** New York: IST, 1979.
24. Gemino, A. "Empirical Comparisons of Systems Analysis Modeling Techniques," Ph.D. Dissertation, University of British Columbia, Canada, 1999.
25. Gemino, A. and Y. Wand. "Comparing Object Oriented with Structured Analysis Techniques in Conceptual Modeling," Draft Manuscript, August 2001.
26. Gemino, A. and Y. Wand. "Complexity and Clarity in Conceptual Modeling: Comparison of Mandatory and Optional Properties," **Data & Knowledge Engineering,** 55:3, 2005, pp. 301-326.
27. Gemino, A. and Y. Wand. "Dimensions in Experimental Evaluation of Conceptual Modeling Techniques," **Proceedings of the 3rd Symposium on Research in Systems Analysis and Design**, St. John's, NL, June 2004.
28. Gemino, A. and Y. Wand. "Evaluating Modeling Techniques Based on Models of Learning," **Communications of the ACM,** 46:10, 2003, pp. 79-84.
29. Gemino, A. and Y. Wand. "A Framework for Empirical Evaluation of Conceptual Modeling Techniques," **Requirements Engineering,** 9:4, 2004, pp. 248-260.
30. Green, P. and M. Rosemann. "Integrated Process Modeling: An Ontological Analysis," **Information Systems,** 25:2, 2000, pp. 73-87.
31. Harl, D. and B. Rumpe. "Meaningful Modeling: What's the

Semantics of 'Semantics'?" **IEEE Computer,** 37:10, 2004, pp. 64-72.
32. Hufnagel, E. and C. Conca. "User Response Data: The Potential for Errors and Biases," **Information Systems Research,** 5:1, 1994, pp. 48-73.
33. Hughes, C. and M. Gibson. "Students as Surrogates for Managers in a Decision-making Environment: An Experimental Study," **Journal of Management Information Systems,** 8:2, 1991, pp. 153-166.
34. Hungerford, B., A. Hevner, and R. Collins. "Reviewing Software Diagrams: A Cognitive Study," **IEEE Transactions on Software Engineering,** 30:2, 2004, pp. 82-96.
35. Iscoe, N., G. Williams, and G. Arango. "Domain Modelling for Software Engineering," **Proceedings of the 13th International Conference on Software Engineering,** Austin, Texas, 1991, pp. 340-343.
36. Jackson, M.A. **Principles of Program Design.** London: Academic, 1975.
37. Jackson, M.A. "The World and the Machine," **Proceedings of the 17th International Conference on Software Engineering,** Seattle, WA, 1995, pp. 283-292.
38. Kim, J., J. Hahn, and H. Hahn. "How do we Understand a System with (So) Many Diagrams? Cognitive Integration Processes in Diagrammatic Reasoning," **Information Systems Research,** 11:3, 2000, pp. 284-303.
39. Kung, C. and A. Solvberg. "Activity Modeling and Behavior Modeling," **Information System Design Methodologies: Improving the Practice.** Amsterdam: Addison-Wesley, 1986.
40. Lim, K. and I. Benbasat. "The Effect of Multimedia on Perceived Equivocality and Perceived Usefulness of Information Systems," **MIS Quarterly,** 24:3, 2000, pp. 449-471.
41. Lu, S. and J. Parsons. "Enforcing Ontological Rules in UML-based Conceptual Modeling: Principles and Implementation," **Proceedings of the 10th Information Workshop on Exploring Modeling Methods in Systems Analysis and Design,** Porto, Portugal, June 2005.
42. Mayer, R.E. "Models for Understanding," **Review of Educational Research,** 59:1, 1989, pp. 43-64.
43. Miller, L. **Instructor's Resource Manual for Modern Systems Analysis and Design.** Upper Saddle River, NJ: Pearson Education, 2002.
44. Moore, G. and I. Benbasat. "Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation," **Information Systems Research,** 23:3, 1991, pp. 192-222.
45. Mylopoulos, J. "Conceptual Modeling and Telos," **Conceptual Modeling, Databases and Cases.** New York: John Wiley, 1992.
46. Newell, A. and H. Simon. **Human Problem Solving.** Englewood Cliffs, NJ: Prentice Hall, 1972.
47. Offen, R. "Domain Understanding is the Key to Successful System Development," **Requirements Engineering,** 7:3, 2002, PP. 172-175.
48. Opdahl, A. and B. Henderson-Sellers. "Grounding the OML Metamodel in Ontology," **Journal of Systems and Software,** 57:2, 2001, pp. 119-143.
49. Opdahl, A. and B. Henderson-Sellers. "Ontological Evaluation of the UML using the Bunge-Wand-Weber Model," **Software and Systems Modeling,** 1:1, 2002, pp. 43-67.
50. Parsons, J. "Effects of Local versus Global Schema Diagrams on Verification and Communication in Conceptual Data Modeling," **Journal of Management Information Systems,** 19:3, 2002, pp. 155-183.
51. Parsons, J. and L. Cole. "What do the Pictures Mean? Guidelines for Experimental Evaluation of Representation Fidelity in Diagrammatical Conceptual Modeling Techniques," **Data and Knowledge Engineering,** 55:3, 2005, pp. 327-342.
52. Parsons, J. and Y. Wand. "Using Objects for Systems Analysis," **Communications of the ACM,** 40:12, 1997, pp. 104-110.
53. **Prentice Hall Test Manager for Hoffer, George Valacich – Modern Systems Analysis and Design,** 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 2001.
54. Scheer, A.-W. **ARIS-Business Process Modeling**. Berlin: Springer-Verlag, 1999.
55. Schütte, R. and T. Rotthowe. "The Guidelines of Modeling- An Approach to Enhance the Quality in Information Models," **Proceedings of the 1998 International Conference on Conceptual Modeling,** Singapore, November 1998, pp. 240-254.,
56. Siau, K. "Information Modeling and Method Engineering: A Psychological Perspective," **Journal of Database Management,** 10:4, 1999, pp. 44-50.
57. Siau, K. "Informational and Computational Equivalence in Comparing Information Modeling Methods," **Journal of Database Management,** 15:1, 2004, pp. 73-86.
58. Siau, K. and Q. Cao. "Unified Modeling Language (UML)- A Complexity Analysis," **Journal of Database Management,** 12:1, 2001, pp. 26-34.
59. Siau, K., J. Erickson, and L.Y. Lee. "Theoretical vs. Practical Complexity: The Case of UML," **Journal of Database Management,** 16:3, 2005, pp. 40-57.
60. Siau, K. and L. Lee. "Are Use Case and Class Diagrams Complementary in Requirements Analysis? An Experimental Study on Use Case and Class Diagrams in UML," **Requirements Engineering,** 9:4, 2004, pp. 229-237.
61. Siau, K. and P.P. Loo. "Identifying Difficulties in Learning UML," **Information Systems Management,** 23:3, Summer 2006, pp. 43-51.
62. Siau, K. and Z. Shen. "Analyzing Unified Modeling Language Using Concept Mapping," **Proceedings of the 2002 America's Conference on Information Systems (AMCIS 2002),** Dallas, TX, August 3-5, 2002, pp. 633-637.
63. Siau, K. and Y. Tian. "Analyzing Unified Modeling Language Using GOMS," **Proceedings of the 12th Workshop in Information Technology and Systems (WITS '02),** Barcelona, Spain, December 14-15, 2002, pp. 7-12.
64. Siau, K., Y. Wand, and I. Benbasat. "The Relative Importance of Structural Constraints and Surface Semantics in Information Modeling," **Information Systems,** 22:2, 1997, pp.l 155-170.
65. Storey, V. "Comparing Relationships in Conceptual Modeling: Mapping to Semantic Classifications," **IEEE Transactions on Data and Knowledge Engineering,** 17:11, 2005, pp. 1478-1489.
66. Topi, H. and V. Ramesh. "Human Factors Research on Data Modeling: A Review of Prior Research, An Extended Framework and Future Research Directions," **Journal of Database Management,** 13:2, 2002, pp. 3-19/
67. Wand, Y., V. Storey, and T. Weber. "An Ontological Analysis of the Relationship Construct in Conceptual Modeling," **ACM Transactions on Database Systems,**

24:4, 1999, pp. 494-528.
68. Wand, Y. and R. Weber. "On the Ontological Expressiveness of Information Systems Analysis and Design Grammars," **Journal of Information Systems,** 3, 1993, pp. 217-237.
69. Yourdon, E. **Modern Structured Analysis.** Englewood Cliffs, NJ: Prentice Hall, 1989.

APPENDIX I
List of Rules and Corollaries (20)

Rule 1	*Only substantial entities in the world are modeled as objects.*
Rule 2	*Ontological properties of things must be modeled as UML-attributes.*
Corollary 1	*Attributes in a UML-description of the real world cannot refer to substantial things.*
Rule 3	*Sets of mutual properties must be represented as attributes of association classes.*
Corollary 2	*An association class must not possess methods or operations.*
Rule 4	*If mutual properties can change quantitatively, methods and operations that change the values of attributes of the association class must be modeled for one or more of the classes participating in the association, objects of which can effect the change, not for the associations class.*
Corollary 3	*An association class must possess at least one attribute.*
Corollary 4	*An association class must not be associated with another class.*
Corollary 5	*An association class must not participate in generalization relationships.*
Rule 5	*An association class represents a set of mutual properties arising out of the same interaction.*
Rule 6	*Classes of objects that exhibit additional behaviour, additional attributes or additional association classes with respect to other objects of the same class, must be modeled as specialized sub-classes.*
Rule 7	*Every UML-aggregate object must consist of at least two parts.*
Rule 8	*Every UML-aggregate must possess at least one attribute or participate in an association.*
Rule 9	*All UML-classes must possess at least one attribute or participate in an association.*
Rule 10	*Object ID's must not be modeled as attributes.*
Rule 11	*The set of attribute values (representing mutual and intrinsic properties) must uniquely identify an object.*
Rule 12	*A specialized class must define more attributes, more operations or participate in more associations than the general class.*
Rule 13	*Every ordinary association (i.e. not aggregation or composition) must be an association class.*
Corollary 6	*Every object must have at least one operation.*
Rule 14	*An object must exhibit additional operations expressing qualitative changes, if a super- or sub-class is defined and instances can undergo changes of class to the super- or sub-class.*

APPENDIX 2
Experimental Conditions

FIGURE 3
First Experimental Condition, Order Processing Domain (22)

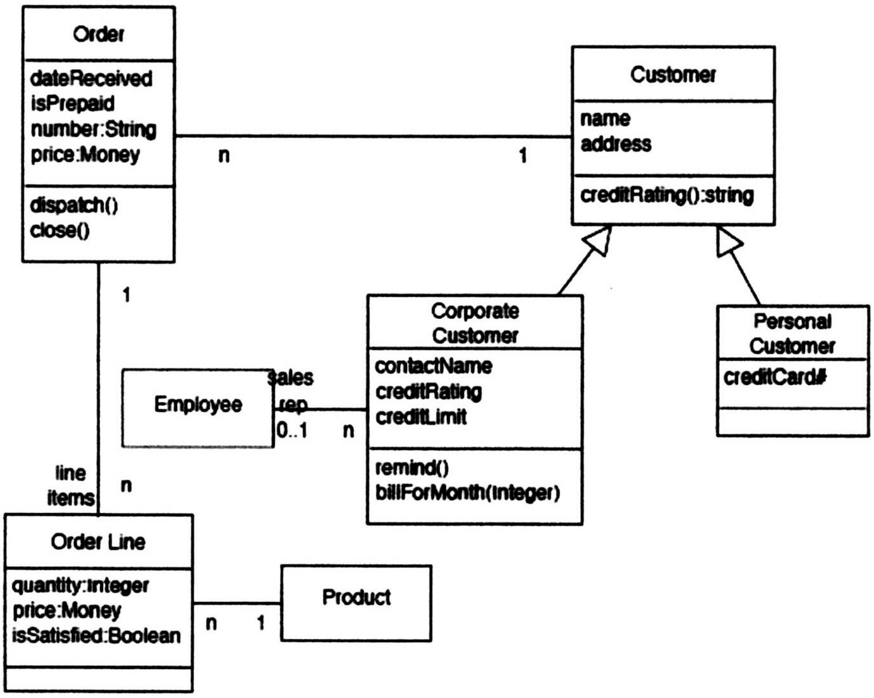


FIGURE 4
First Experimental Condition, Car Rental Domain (43)

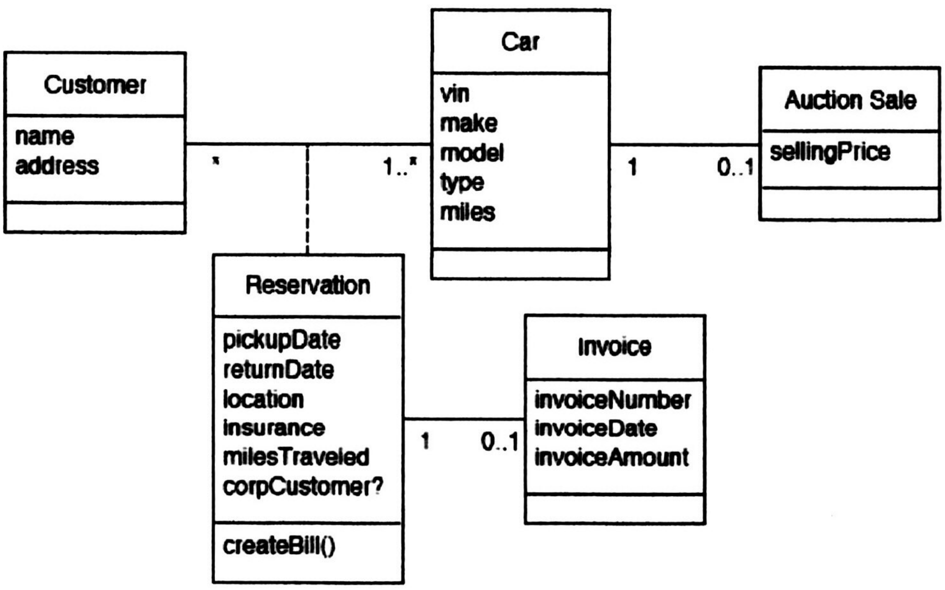


FIGURE 5
Second Experimental Condition, Order Processing Domain

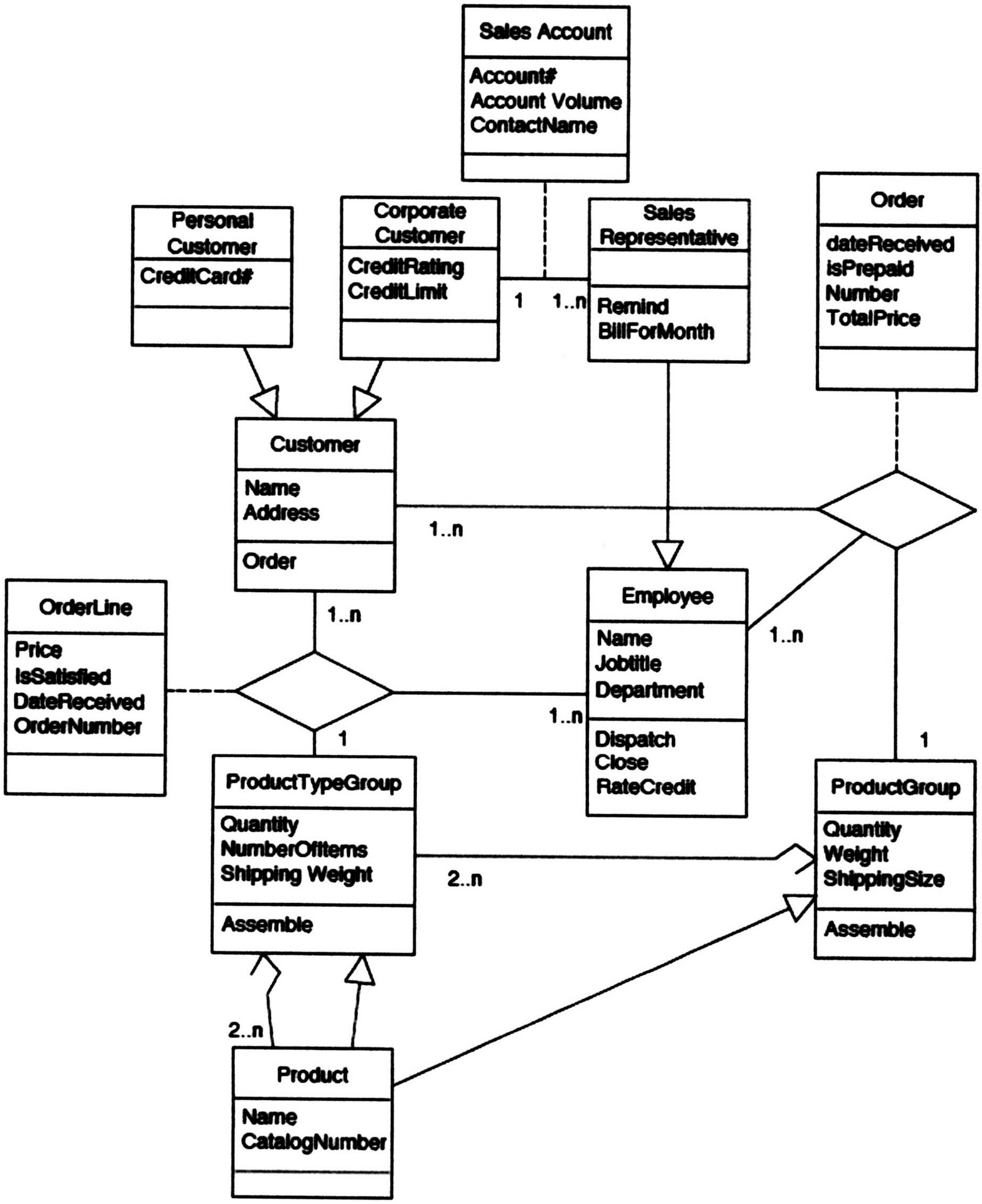


FIGURE 6
Second Experimental Condition, Car Rental Domain

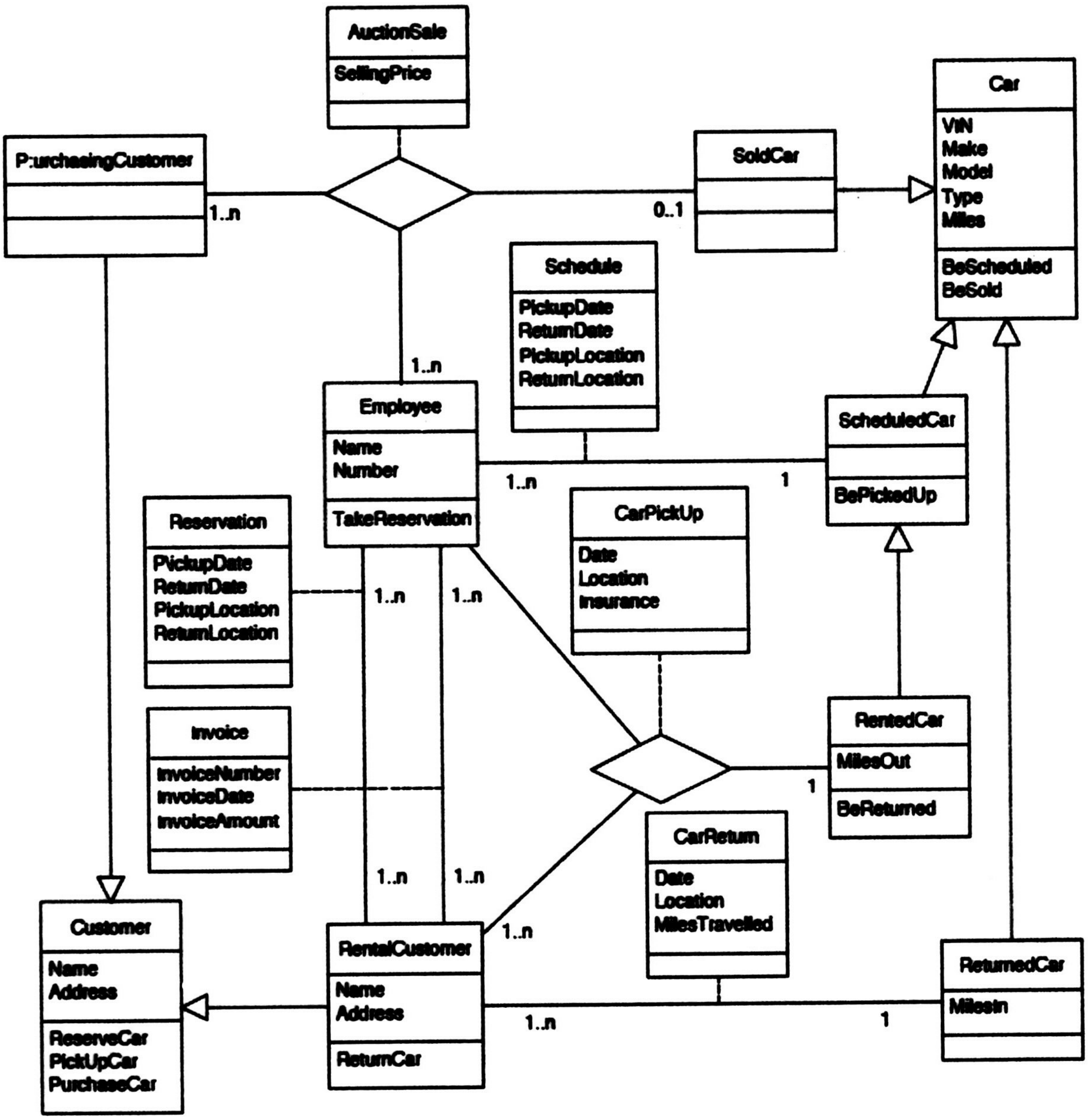


FIGURE 7
Third Experimental Condition, Order Processing Domain

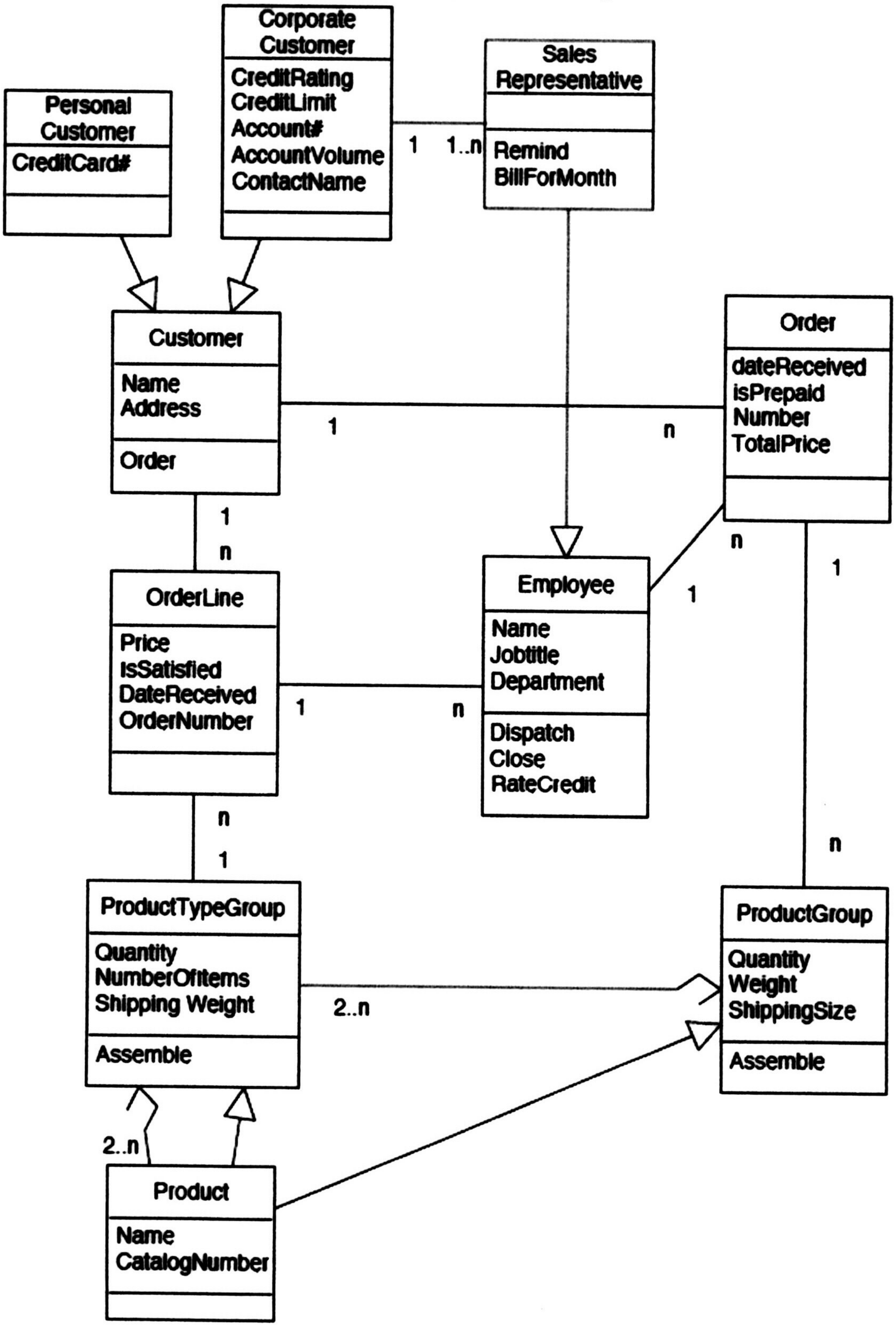


FIGURE 8
Third Experimental Condition, Car Rental Domain