Comprehending Object and Process Models: An Empirical Study

Ritu Agarwal, Prabuddha De, and Atish P. Sinha

Abstract—Although prior research has compared modeling performance using different systems development methods, there has been little research examining the comprehensibility of models generated by those methods. In this paper, we report the results of an empirical study comparing user comprehension of object-oriented (OO) and process-oriented (PO) models. The fundamental difference is that while OO models tend to focus on *structure*, PO models tend to emphasize *behavior* or processes. Proponents of the OO modeling approach argue that it lends itself naturally to the way humans think. However, evidence from research in cognitive psychology and human factors suggests that human problem solving is innately procedural. Given these conflicting viewpoints, we investigate empirically if OO models are in fact easier to understand than PO models. But, as suggested by the theory of cognitive fit, model comprehension may be influenced by task-specific characteristics. We, therefore, compare OO and PO models based on whether the comprehension activity involves: 1) only structural aspects, 2) only behavioral aspects, or 3) a combination of structural and behavioral aspects. We measure comprehension through subjects' responses to questions designed along these three dimensions. Two experiments were conducted, each with a different application and a different group of subjects. Each subject was first trained in both methods, and then participated in one of the two experiments, answering several questions relating to his or her comprehension of an OO or a PO model of a business application. The comprehension questions ranged in complexity from relatively simple (addressing either structural or behavioral aspects) to more complex ones (addressing both structural and behavioral aspects). Results show that for most of the simple questions, no significant difference was observed insofar as model comprehension is concerned. For most of the complex questions, however, the PO model was found to be easier to understand than the OO model. In addition to describing the process and the outcomes of the experiments, we present the experimental method employed as a viable approach for conducting research into various phenomena related to the efficacy of alternative systems analysis and design methods. We also identify areas where future research is necessary, along with a recommendation of appropriate research methods for empirical examination.

Index Terms—Cognitive fit, experimental method, human factors, model comprehension, object-oriented modeling, process-oriented modeling.

1 Introduction

Several systems development methods exist for analyzing, designing, implementing, and testing complex software systems. Among those, the process-oriented (PO) methods—in particular, the structured techniques—have dominated systems development efforts for over three decades. However, with the emergence of object-oriented (OO) methods as a viable and attractive alternative for systems development, many organizations have adopted and are actively utilizing those methods.

The object-oriented approach provides a powerful and effective environment for analyzing, designing, and implementing flexible and robust real-world systems, offering benefits such as *encapsulation* (information hiding), *polymorphism*, *inheritance,* and *reusability* [3], [6], [19], [30]. A major advantage of using the OO approach is that it provides a continuum of representation from analysis to design to implementation, thus engendering a seamless transition from one model to another [6], [19]. However, in spite of the purported *technical* advantages associated with OO methods, industry experience with this paradigm has been mixed. Indeed, there is substantial evidence to suggest that the incorporation of the OO approach into systems development activities has been less than ideal [4]. One plausible reason for this is that the problems with OO methods are not inherently technical in nature; rather, they are an outcome of the difficulties systems developers experience with those methods. As noted cogently by Perry et al. [27, p. 36], "we must consider the larger development picture, which encompasses organizational and social, as well as technological factors."

The OO and PO methods provide their own representational notations for constructing a set of models during the development life cycle for a given system. During the initial phases, the models developed are abstract, focusing on the external qualities of the system. They progressively become more and more detailed, as internal, implementation-related factors are taken into account. The OO and PO methods incorporate a variety of techniques and representational constructs to facilitate the process of model development. But while both provide techniques and constructs to model an information processing system—in

• *R. Agarwal is with the Department of Decision and Information Technologies, University of Maryland, College Park, MD 20742. E-mail: ragarwal@rhsmith.umd.edu.*
• *P. De and A.P. Sinha are with the Department of MIS & Decision Sciences, University of Dayton, Dayton, OH 45469. E-mail: {pde, sinha}@udayton.edu.*

Manuscript received 10 July 1998; revised 22 Feb. 1999.
Recommended for acceptance by D. Ross Jeffery.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number 109542.

terms of its data and the processes that act on the data—they do so in substantially different ways. A major difference is that OO models are built around *objects*, whereas PO models are built around *processes*.

Given this radical difference in emphasis, researchers have investigated the relative efficacy of one type of method vs. another insofar as modeling performance is concerned [1], [37], [40]. Despite the claims made by OO proponents about the superiority of OO methods, prior research has failed to establish that they are universally more powerful. For example, arguing for contingent effects on modeling performance, Agarwal et al. [1], [2] showed that factors such as the nature of the application being modeled and the type of prior modeling experience exert significant influence on an individual's ability to utilize a particular method.

With the gradual adoption of OO analysis and design methods by an increasing number of organizations, it is clearly important to conduct research examining the efficacy of those methods vis-à-vis traditional PO methods. However, although prior studies have compared modeling performance—especially as it relates to requirements analysis—using those two types of methods, there has been very little research examining how easy or difficult it is to *understand* the end products of modeling, the models themselves. The lack of attention to model comprehension is all the more disturbing as models play a crucial role in providing a communication mechanism among different stakeholders during the systems development process. For instance, systems analysts use requirements analysis models to verify with users if all their requirements have been captured, as well as to communicate those requirements to systems designers. For large software systems, teams may be geographically distributed [28] and models constitute the primary communication vehicle among developers.

Because effective communication can only result if the models themselves are easily comprehensible, and because the ultimate quality of any large-scale system is, in turn, dependent on the quality of communication among the multiple parties involved, it is imperative that we examine the issue of comprehension of models developed using different methods. In this paper, we describe the process and outcomes of two experiments conducted to examine if OO requirements analysis models are easier to comprehend than their PO counterparts. Consistent with prior research that has established the moderating influence of the characteristics of the application being modeled, we compare OO and PO models based on whether the comprehension activity involves: 1) only structural aspects, 2) only behavioral (process-oriented) aspects, or 3) a combination of both. The quality of comprehension is measured through subjects' responses to questions designed along these dimensions.

The research strategy of experimentation is presented as a useful approach for investigating the phenomena of interest in this study. As noted by Jarvenpaa [20], laboratory experimentation represents a feasible and under-utilized method for conducting research related to information systems issues. She points out that the traditional criticisms raised against experimentation—those of low external validity and low generalizability—are an outcome of poor design and application of the research method, rather than an intrinsic limitation of the method itself. Indeed, experiments are a preferred research strategy when the cumulative body of knowledge related to a specific phenomenon is limited. It increases the internal validity of the study, albeit at the expense of some realism [8]. Experimental findings can then be used to refine existing theory or develop new theory to be tested in the field. In this study, we pay careful attention to the common problems in experimental information systems research as articulated by Jarvenpaa et al. [21]: a lack of underlying theory, proliferation of measurement instruments, inappropriate research designs, and diversity of experimental tasks.

The remainder of this article is organized as follows. We first present the major theoretical themes underlying our study and review prior empirical research relating to the efficacy of OO and PO models. Next, we describe the specific research questions examined in this study and the research methodology employed to address those questions. The results of our study are presented in the following section. The article concludes with a discussion of the results and their implications for research and practice.

2 Theoretical Background

The major reference theories for examining the efficacy of alternative systems analysis and design methods come from the cognitive psychology and human factors literatures. Prior research in cognitive psychology and human factors underscores the fact that the external representation of a problem could play a major role in problem solving. In their theory of *human information processing*, Newell and Simon [25] posit a set of cognitive processes that produce the behavior of a human problem solver. Problem solving takes place within a *problem space*, which is the internal representation of the task environment used by the human subject. Two of the key determinants of the problem space are the *task* or problem itself and the specific *representation* used to describe the task.

The cognitive processes involved in solving computing and modeling-related problems have been examined in a significant body of prior research (e.g., [5], [34]). Researchers in human factors have also explored the influence of the nature of the task and the way it is represented on problem solving performance. An important notion generated from this stream of work is that of *cognitive fit* [36]. The basic model of cognitive fit views problem solving as the outcome of the relationship between the external problem representation and the problem solving task, which are both characterized by the types of information they emphasize. Cognitive fit exists when the cognitive processes used to act on the problem representation and those used to complete the task match, resulting in superior problem solving performance.

Vessey and Galletta [38] examined the effects of a match among problem solving skill, problem representation, and problem solving task on problem solving performance. They used two types of tasks, spatial and symbolic, and two types of representations, graphs and tables. Both spatial and symbolic subject skills were measured. Based on the results, the authors concluded that the effectiveness of a problem representation varied with the type of task to be solved.

In the domain of systems analysis and design, human factors researchers compared performance with alternative methods. In an experimental study comparing data flow diagrams (DFDs) and the Integrated Definition Method (IDEF0), Yadav et al. [42] found that, in general, DFDs were easier to learn and use, although no definitive statements could be made about the quality of the model produced by both techniques. Vessey and Conger [37] compared process, data, and object-oriented methodologies for requirements specification using process-tracing methods with a small sample. Their results suggest that novice systems analysts find the process-oriented methodology easiest to apply and the object-oriented methodology the most difficult. In another study, Wang [40] conducted an experiment to compare the effectiveness of the DFD method with the object-oriented analysis (OOA) method. The DFD method appeared to be easier to learn for inexperienced participants, but with further training, the OOA method led to more accurate answers.

Agarwal et al. [1] examined the effects of cognitive fit between problem solving task and problem solving tool on performance in generating a requirements model. In their study, subjects used two different systems development methods or tools—object-oriented and process-oriented—on two tasks. One task was classified as object-oriented, while the other was classified as process-oriented. The classification was done based on the inherent features of the tasks. As predicted by the theory of cognitive fit, the researchers found that superior performance resulted when the process-oriented tool was applied to the process-oriented task. For the object-oriented task, however, they did not find a significant difference in performance using the two tools.

Although prior research has compared modeling performance using different systems development methods and tools, there has been little research examining the comprehensibility of models (or representations) generated by those methods or tools. However, there has been some work investigating user comprehension of database schemas generated using different data models. For example, Kim and March [22] compared two semantic data models—extended entity-relationship (EER) and Nijssen information analysis methodology (NIAM)—with respect to modeling as well as comprehension. In another study, Shoval and Frumermann [31] compared user comprehension of diagrammatic OO and EER schemas.

In the broader area of systems analysis and design, there appears to be a paucity of work focusing on the issue of user comprehension. Moynihan [24] reports an experiment to evaluate the relative effectiveness of the functional decomposition and object-oriented methods as a means of communication between clients and developers during the early stages of system development. Effectiveness was assessed by asking the subjects to critique two equivalent analysis models developed using the two methods. The results of the experiment suggest that the functional decomposition method is more effective than the object-oriented method for early client/developer communication.

In another study, Gemino and Wand [17] conducted an experiment to compare three different representations of a business problem: text, OO diagram, and DFD. The participants in the experiment completed a series of three tests: a comprehension test, a problem solving test, and a Cloze test. The preliminary findings of this study suggest that both OO diagrams and DFDs are better than text descriptions in comprehension tests. Although OO diagrams ranked lowest in ease-of-use, they ranked highest in problem solving and comprehension.

In summary, theories of human information processing and cognitive fit strongly suggest that problem representation is a determinant of performance, from the perspective of problem solving, as well as comprehension. Our work builds upon these theories and prior empirical work by examining a relatively unexplored issue. Insofar as the task of systems analysis and design is concerned, there is a lack of conclusive evidence on the relative comprehensibility of representations or models developed using different methods and tools—a gap in the literature that this study attempts to address.

3 Research Questions

Requirements modeling constitutes the most consequential phase of the systems development life cycle [10], [37]. To ensure the ultimate success of a system, it is critical that the systems analyst develop a complete and accurate set of requirement specifications, forming the basis for systems design. The models developed as a result of requirements analysis are represented primarily in diagrammatic form, supplemented with textual descriptions for those parts that cannot be captured diagrammatically. Requirements models generated using an OO method and a PO method are expressed as object class diagrams and data flow diagrams (DFDs), respectively.

The object and process models investigated in this study provide two different representations for a given problem. Although represented differently, they are *equivalent* with respect to the *information* they capture. According to Larkin and Simon [23, p. 67], "two representations are informationally equivalent if all of the information in the one is also inferable from the other, and vice versa." Indeed, researchers in human factors have compared problem solving and comprehension performance using informationally equivalent representations such as graphs and tables [38].

Prior research in cognitive psychology and human factors indicates that the effectiveness of solving a task depends both on the task itself and its representation. For example, Vessey and Galletta [38] found that graphs resulted in faster performance on spatial tasks than tables, and tables resulted in both faster and more accurate performance on symbolic tasks than graphs. DeMarco [11] suggests that applications could be *function strong*—i.e., they contain significant amounts of processing—or *data strong*—i.e., they have significant data requirements. Agarwal et al. [1] studied performance in modeling two types of systems analysis and design tasks: structure-oriented and process-oriented. They found that, for the

PO task, performance was better using the PO tool than the OO tool, supporting the notion of a fit between task and tool. For the structure-oriented task, however, no significant difference in performance was observed.

The effects of the nature of an application or task on performance have also been studied in other phases of systems development, such as design and coding [32], [39]. In addition, there is substantial evidence outside the systems development area suggesting that problem solvers employ different processing strategies for tasks with different characteristics [12], [33], [35].

The fundamental difference between the two types of representation investigated in this study is that while OO models tend to emphasize the *structural* aspects of an application—in terms of objects, classes, and relationships—PO models tend to focus on its *behavioral* or process-oriented aspects [7], [15], [37]. Rumbaugh et al. [30, pp. 6-7] point out that, in contrast to the function-oriented methods, where "the primary emphasis is placed on specifying and decomposing system functionality... the object-oriented approach focuses first on identifying objects from the application domain, then fitting procedures around them." Fayad et al. [14, p. 110] suggest that "instead of worrying about the data, processes, and data flow among the processes," software developers who have used structured techniques "need to think in terms of objects, methods, and inheritance."

A typical real-world involves both structural aspects, although the emphases could be different [1], [3], [6]. Iivari [18] presents three dimensions for OO modeling—structure, function, and behavior—parallel to the object, functional, and dynamic models of the OMT methodology [30]. The process-oriented approach also supports multiple perspectives through diagrams, although the primary focus is on the functional aspects presented through data flow diagrams. However, while models developed using an OO method tend to form a cohesive set, the models developed using a process-oriented method lack a common underlying representation and are, therefore, *weakly connected* [6].

Proponents of the OO approach claim other advantages as well. They argue that the resulting systems tend to be more stable, because they are built around objects, which are less prone to change than procedures or functions [3], [19], [30]. Also, according to them, OO modeling lends itself naturally to the way humans think. However, evidence from research in cognitive psychology and human factors suggests otherwise: that human problem solving is innately procedural [26], [29], [41].

Given these conflicting viewpoints, we examine in this study if OO models are in fact easier to comprehend than PO models. However, as suggested by the theory of cognitive fit [36], model comprehension may be influenced by task-specific characteristics. Hence, we compare comprehension of OO and PO models based on whether the comprehension activity involves: 1) only structural aspects, 2) only behavioral aspects, or 3) a combination of structural and behavioral aspects. Note that comprehension tasks belonging to the third category are typically more *complex* than those belonging to the first two categories because understanding both structure and behavior together is likely to be more difficult than understanding one of those aspects in isolation.

In comparing the comprehension of informationally equivalent OO and PO models, we need to further consider the issue of *computational equivalence*. According to Larkin and Simon [23, p. 67], "two representations are computationally equivalent if they are informationally equivalent and, in addition, any inference that can be drawn easily and quickly from the information given explicitly in the one can also be drawn easily and quickly from the information given explicitly in the other, and vice versa." So although OO and PO models are informationally equivalent, if they are not equivalent computationally with respect to a comprehension task, we can expect to find differences in user understanding of the two types of models.

In this study, comprehension is assessed through subjects' responses to questions designed to elicit their understanding of OO and PO models along the three dimensions outlined above. Some of the comprehension questions used in the study are primarily structure-oriented, some are primarily process-oriented, while the remaining problems have both structure and process components. Because cognitive fit exists when the processes acting on the task and the representation match—i.e., when an OO model is used to answer structure-oriented questions and when a PO model is used to answer process-oriented questions—we pose the following research questions:

RQ1: Is it easier to understand structure-oriented aspects of an application represented using an OO model rather than a PO model?

RQ2: Is it easier to understand process-oriented aspects of an application represented using a PO model rather than an OO model?

To answer the more complex questions involving structure and behavior, designers must comprehend both of those aspects together. If an OO model is used, cognitive fit will exist only for the structural aspects. On the other hand, if a PO model is used, cognitive fit will exist only for the process-oriented aspects. Depending on which model provides a better fit and which of the two aspects dominates the other, one model may be computationally more efficient than the other with respect to comprehension. We state this as an exploratory research question:

RQ3: Is it easier to understand both structure-oriented and process-oriented aspects of an application together using an OO model or a PO model?

The experiments conducted to examine these questions are described next.

4 Experimental Method

We employed the research strategy of laboratory experimentation to gather empirical data for addressing the research questions posed above. In designing the experiment, careful attention was paid to the choice of experimental materials and the inclusion of experimental controls. Furthermore, to increase confidence in the findings, the main experiment was replicated using another task. The major dependent variable in the experiments was the accuracy of subject responses to comprehension questions using the OO and PO models under investigation.

4.1 Task Materials

Two business information processing tasks, represented both as an OO model and a PO model, were used as stimulus materials in the experiments for assessing comprehension. These constituted the primary treatment. The first task was for a payroll system in a corporation (henceforth referred to as the ABC case), while the second task was for a motor vehicle registration system (henceforth referred to as the Texas case). Fig. 1, Fig. 2 and Fig. 3, Fig. 4 (Fig. 5, Fig. 6 and Fig. 7, Fig. 8) present the informationally equivalent OO and PO models for the ABC case (Texas case), respectively.

An OO model of a business application is developed by identifying the domain object classes, their attributes, and structural relationships [6]. For example, the OO model shown in Fig. 5 identifies several domain classes (e.g., VEHICLE, CAR, TRUCK, TITLE, and REGISTRATION), their attributes (e.g., vehicle_number and year of VEHICLE) and methods (e.g., calculate_renewal_fee and issue_new_registration inside REGISTRATION). It also depicts the generalization relationship that exists between a subclass (e.g., CAR) and its superclass (e.g., VEHICLE) using a semicircle connector, implying that the subclass inherits the features of its superclass. Inheritance of an attribute or method by a subclass can be overriden (blocked) by placing an asterisk next to its name (e.g., number_passengers and diesel in TRAILER). The OO diagram also shows the message connections between objects by linking an object calling a method to the method being called. For example, the message connection between TITLE CLERK and issue_title shows that the TITLE CLERK object sends a message called issue_title to the TITLE object.

In addition to the graphical representation, an OO model also contains a textual specification of details relating to the methods and sequencing of the methods. The textual representation distinguishes between *atomic methods*, which are indivisible, and *meta methods*, each of which comprises a sequence of methods. An example of an atomic method is issue_title within TITLE (see Fig. 6); the method name is followed by the name of the object, REGISTRATION, from which the method needs some data. An example of a meta method is MM1, which consists of three atomic methods performed in sequence. The specification of the meta method includes the names of any documents that are sent from a source object or sent to a destination object, along with the names of the source/destination objects. The notation for doing so is {SOURCE/DESTINATION OBJECT [document_name]}, which precedes the name in the case of a source object and follows it in the case of a destination object. For example, the specification issue_renewal_notice {OWNER [renewal_notice]} in the second step of MM1

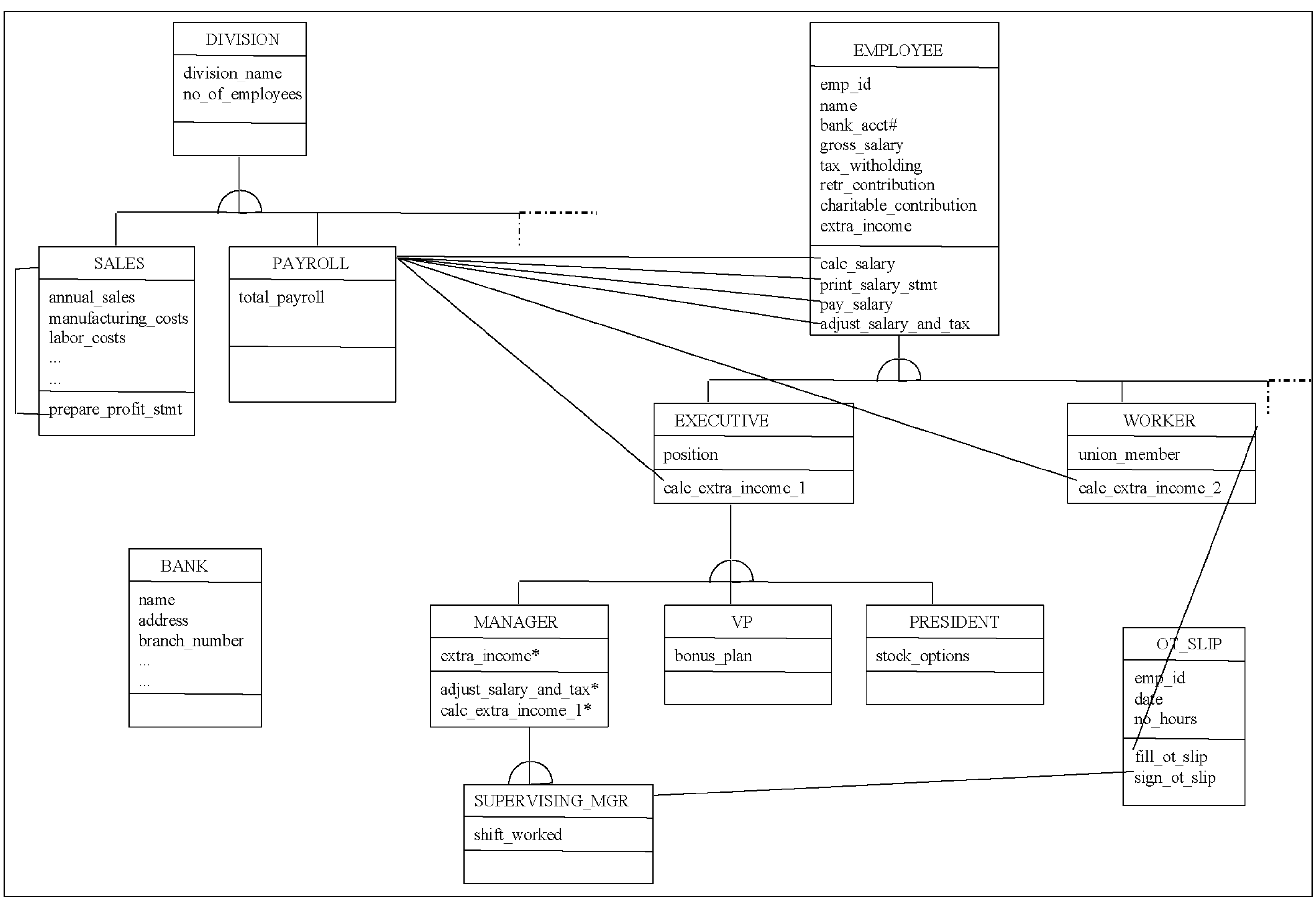

Fig. 1. ABC's payroll system: object diagram.

Atomic Methods:

prepare_profit_stmt
calc_extra_income_1 <SALES>
calc_salary
print_salary_stmt
pay_salary
adjust_salary_and_tax
calc_extra_income_2 <OT_SLIP>
fill_ot_slip
sign_ot_slip

Meta Methods:

MM1

1. calc_salary
2. pay_salary {EMPLOYEE [check] | BANK [deposit_advice]}
3. print_salary_stmt {EMPLOYEE [salary_stmt]}

MM2

1. prepare_profit_stmt {PAYROLL [profit_statement]}
2. calc_extra_income_1
3. adjust_salary_and_tax

MM3

1. MM4
2. calc_extra_income_2
3. adjust_salary_and_tax

MM4

1. fill_ot_slip {SUPERVISING_MGR [ot_slip]}
2. sign_ot_slip {PAYROLL [ot_slip]}

Fig. 2. ABC's payroll system: methods.

indicates that the issue_renewal_notice method sends the renewal_notice document to the OWNER object.

The corresponding PO model for the Texas case (see Fig. 7) is shown as a data flow diagram (DFD), which represents the flow of data and processing in the system [10]. A rectangle, such as OWNER, represents an *external entity* that interacts with the system. An oval, such as "issue new registration," represents a *process* that transforms data within the system. Additional contextual information such as the name of the individual or organization responsible for invoking the process is often located in the bottom part of the oval. For example, R.CLERK (registration clerk) is responsible for completing four of the processes in the Texas system. The open-ended rectangles represent *data stores* for storage of some type of data. REGISTRATION INFO. and VEHICLE INFO. are examples of data stores that store registration-related information and vehicle-related information, respectively.

Processes, data stores, and external entities are connected in a DFD by directed arrows representing *data flows*. A data flow from a data store to a process indicates that the process is using the data associated with that data store. A double-headed arrow between a process and a data store indicates that the process is using as well as modifying the data in that data store.

As with OO models, the graphical DFD of a PO model is supplemented with textual descriptions of additional information that cannot be captured directly in the DFD. A DFD is accompanied by a textual *data dictionary*, which contains additional details relating to how data is organized within the system. It usually contains an entry corresponding to each data store and describes the attributes within the store. For example, the data dictionary for the Texas case shown in Fig. 8 documents the relevant entities (e.g., VEHICLE and REGISTRATION) and their attributes (e.g., vehicle_number and year of VEHICLE). If an entity could be one of different possible types (e.g., types CAR, TRUCK, etc. of VEHICLE entity), that information is specified in the data dictionary by separating each type with a vertical bar. If there are certain attributes that do not apply to an entity, those types are placed after the attributes within { }. For example, the "number_passengers" attribute does not apply to the VEHICLE type called TRAILER. On the other hand, if there are certain attributes that apply only to some of the types, but not all of them, those attributes are enclosed in ( ). For example, the "temp_gross_wt" attribute belongs only to TRUCK and not to the other VEHICLE objects.

A total of eight questions were developed for each task, ranging in complexity from relatively simple questions involving only structure or behavior to more complex ones involving both aspects. An example of a question involving only *structure* is Question 2 from the Texas case:

Describe the hierarchy of vehicles, i.e., the different types of vehicles and how they are related.

An example of a question involving only *behavior* is Question 4 from the same case:

List all the activities that are involved in the title issuance for a trailer.

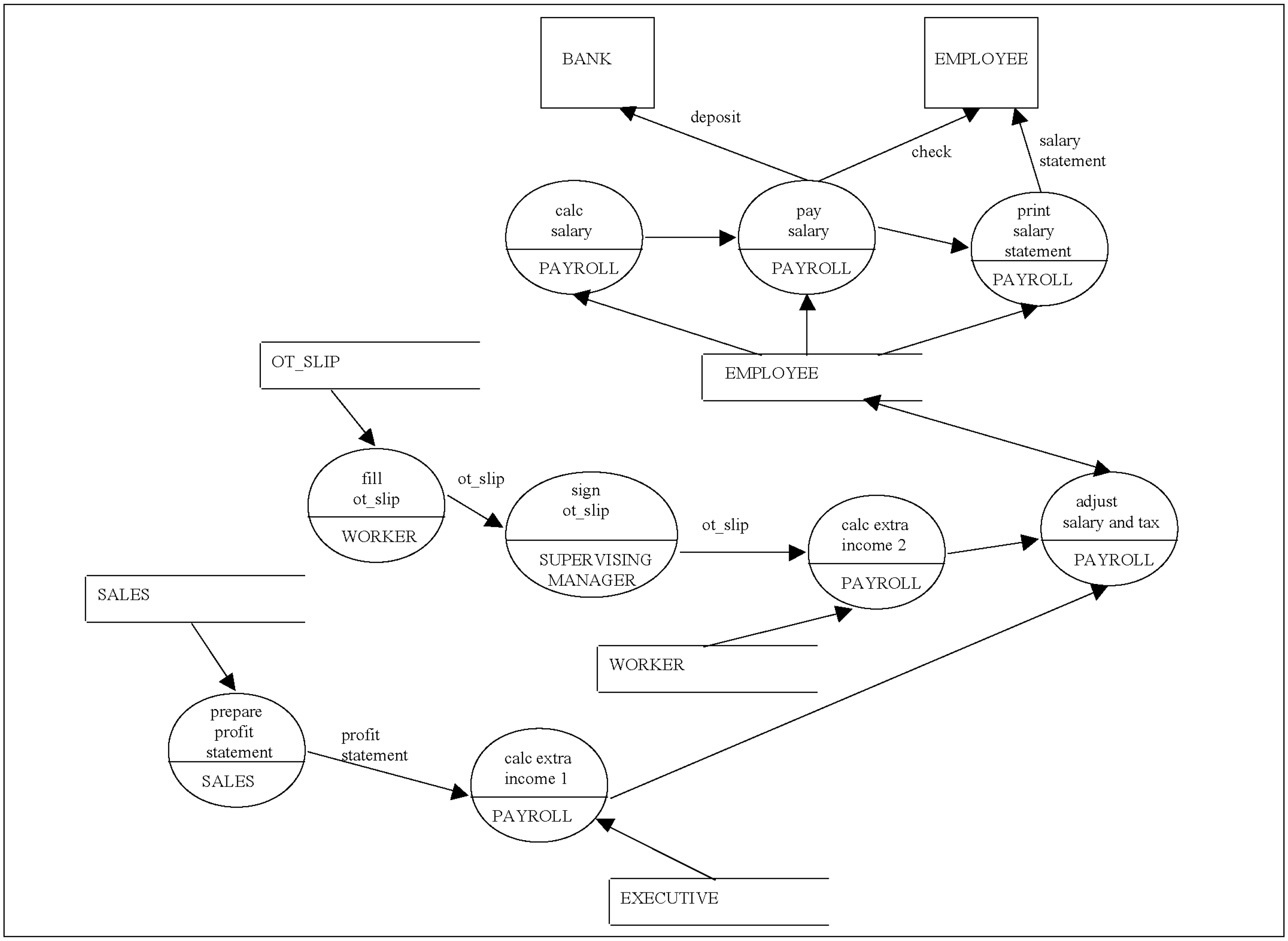

Fig. 3. ABC's payroll system: data flow diagram.

EMPLOYEE =	emp_type [EXECUTIVE [MANAGER [SUPERVISING_MANAGER] | VP | PRESIDENT] | WORKER] + emp_id + name + bank_acct# + gross_salary + tax_ witholding + retr_contribution + charitable_contribution + extra_income {MANAGER} + position (EXECUTIVE) + union_member (WORKER) + bonus_plan (VP) + stock_options (PRESIDENT) + shift_worked (SUPERVISING_MANAGER)
OT_SLIP =	emp_id + date + no_hours
SALES =	annual_sales + manufacturing_costs + labor_costs + ...

Fig. 4. ABC's payroll system: data dictionary.

An example of a *complex* question involving both structure and behavior is Question 8 from the ABC case:

Consider two major functions of the system—payment of regular salary and payment of extra income. List the data sources that are common to the activities associated with both these functions.

The questions for the ABC case and the Texas case are listed in Appendix A and Appendix B, respectively. For both cases, Questions 1 and 2 are structure-oriented, while Questions 4 and 7 are process-oriented. The remaining questions—Questions 3, 5, 6, and 8—involve both structure and behavior.

4.2 Subjects

The subjects in this study were undergraduate students majoring in information systems at a large state university. They had previously taken courses in systems analysis and design and database management. Apart from a few who had some internship experience, the subjects did not have any real-world systems analysis and design experience. The subjects had some prior experience with process-oriented modeling; this is precisely the population we sought to generalize to. The existing information systems workforce in industry today has prior experience in PO modeling, and it is important to gain insights into the relative performance of such individuals with the newer OO models. A total of 71 subjects participated in the two experiments. Depending on

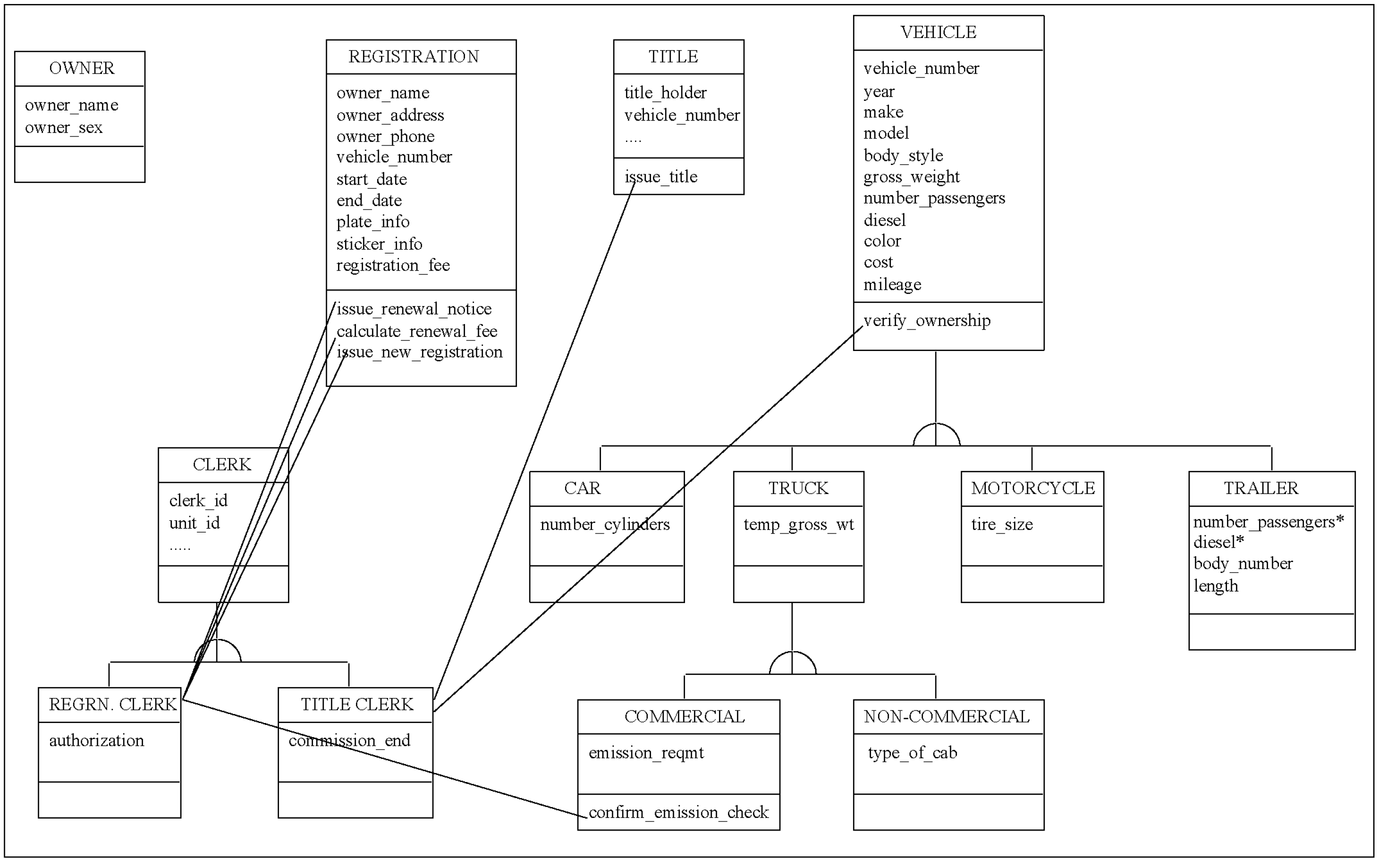

Fig. 5. Registration and title system in Texas: object diagram.

Atomic Methods:

issue_new_registration <VEHICLE>
issue_renewal_notice
calculate_renewal_fee <VEHICLE>
issue_title <REGISTRATION>
verify_ownership
confirm_emission_check

Meta Methods:

MM1

1. calculate_renewal_fee
2. issue_renewal_notice {OWNER [renewal_notice]}
3. {OWNER [renewal_notice + payment]} issue_new_registration {OWNER [renewed_registration]}

MM2

1. calculate_renewal_fee
2. issue_renewal_notice {OWNER [renewal_notice]}
3. confirm_emission_check
4. {OWNER [renewal_notice + payment]} issue_new_registration {OWNER [renewed_registration]}

MM3

1. {OWNER [proof_of_ownership + title fee]} verify_ownership
2. issue_title {OWNER [title]}

Fig. 6. Registration and title system in Texas: methods.

the performance on the experimental task, each subject received credit, which was factored into the final grade for the course. Hence, a significant concern in experiments involving student subjects, that of subject motivation, was addressed.

4.3 Experimental Design and Controls

Two experiments were carried out using the two cases described above. For both experiments, the subjects were randomly assigned to one of two groups: one group received the OO model while the other received the PO model. Subjects in both groups had to answer questions based on the model provided. To counteract the possibility of sequence effects in the comprehension test, the eight

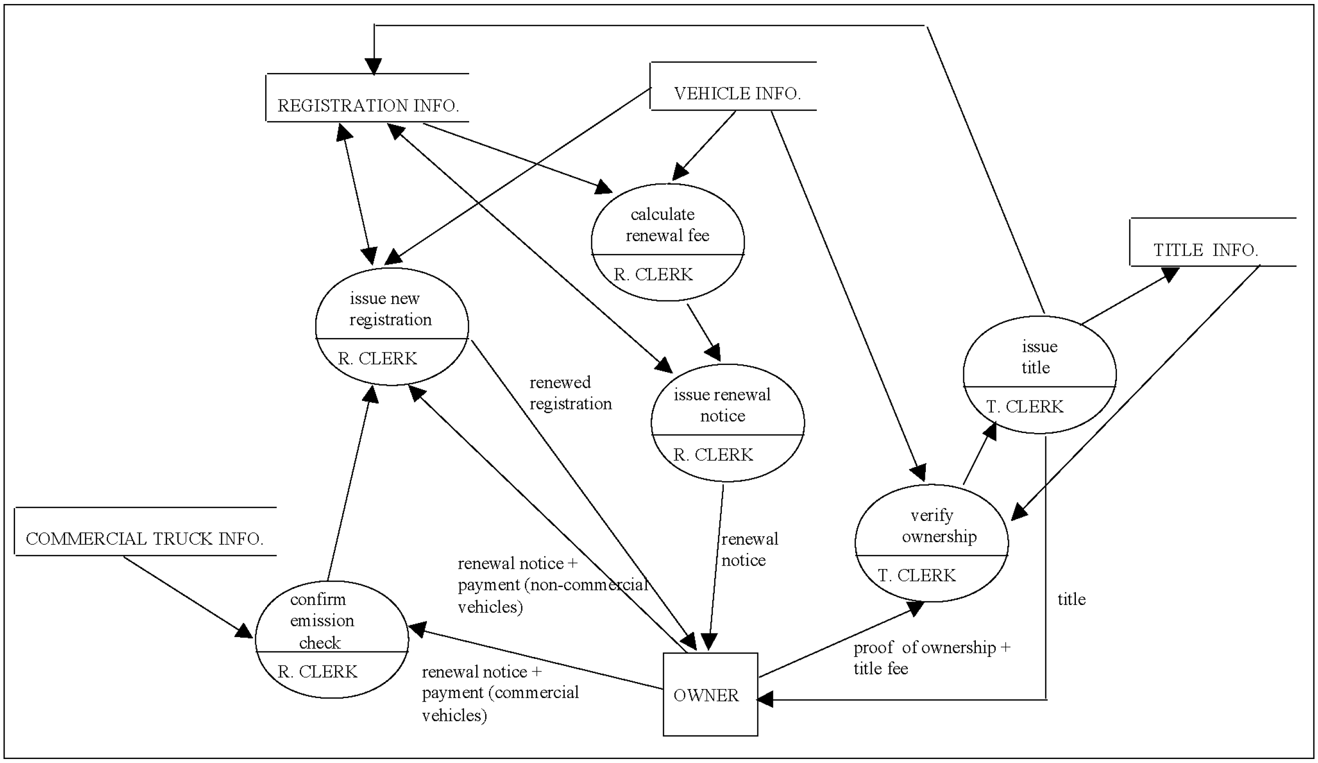

Fig. 7. Registration and title system in Texas: data flow diagram.

VEHICLE =	vehicle_type [CAR | TRUCK [COMMERCIAL | NON-COMMERCIAL] | MOTORCYCLE | TRAILER] + vehicle_number + year + make + model + body_style + gross_weight + number_passengers + diesel + color + cost + mileage + number_cylinders (CAR) + temp_gross_wt (TRUCK) + tire_size (MOTORCYCLE) + number_passengers {TRAILER} + diesel {TRAILER} + body_number (TRAILER) + length (TRAILER) + emission_reqmt (COMMERCIAL) + type_of_cab (NON_COMMERCIAL)
REGISTRATION =	owner_name + owner_address + owner_phone + vehicle_number + start_date + end_date + plate_info + sticker_info + registration_fee
TITLE =	title_holder + vehicle_ numner + ...
CLERK =	clerk_type [T. CLERK | R. CLERK] + clerk_id + unit_id + ...

Fig. 8. Registration and title system in Texas: data dictionary.

questions were randomly ordered for each subject. Thus, no two subjects received the same question sequence. The two levels of random assignment further helped ensure the internal validity of the experiment; i.e., extraneous variance related to between-subject factors was reduced [8]. In the first experiment, 18 subjects received the OO model and 18 others received the PO model for the ABC case. In the second experiment, 18 subjects received the OO model and 17 subjects received the PO model for the Texas case. The second experiment was a replication of the first with a different set of subjects and a different task to ensure that the results were not biased by any task-specific characteristics. Each subject participated in one of the two experiments and answered all eight questions for the given case using the OO model or the PO model provided.

4.4 Procedure

Prior to the two experiments, a pilot test was conducted with a different set of students. A total of 18 students answered comprehension questions for both cases, using the OO model for one case and the PO model for the other. The order of presentation of the cases and the model representations was counterbalanced. As a result of the pilot test, we determined that, for the actual experiments, it would be better, from a time standpoint, for each subject to work on only one type of model for a given case. This would also eliminate the possibility of any learning effects from a repeated-measures design. Doing so allowed us to use eight questions for each case instead of six questions as in the pilot.

For the main experiments, subjects were trained in object-oriented modeling concepts, and their prior knowledge in process-oriented modeling was refreshed through a review session. To guard against any possible bias, the same instructor provided the training for both modeling methods. The training materials included handouts that described in detail the fundamental concepts underlying those methods and the techniques for applying them. As part of their training, the participants had to solve several systems analysis and design problems using both methods. After the training session, each subject participated in one of the two experiments.

4.5 Dependent Variable

The dependent variable investigated in this study is the accuracy of comprehension. This was assessed by first assigning a discrete score (0, 0.5, or 1) to each part of an answer, depending on whether that part was answered fully and correctly. For example, the answer to question 8 of the ABC case, which has two parts—WORKER and EXECUTIVE—was graded individually on each of those parts. Those individual scores were then added to reflect the total comprehension score for that question.

In addition, a second grading scheme was employed to assign an overall comprehension score on a scale of 1 to 7 for an answer to a question. This score is important because it evaluates the notion of comprehension accuracy on a holistic basis, and is similar to some of the scoring schemes employed successfully in past human factors research (see, for example, [1]). Note that both grading schemes were intended to measure the same dependent variable and, therefore, their outcomes should be consistent with one another [9].

As noted earlier, considerable care was taken to ensure the reliability and validity of the dependent variable. A scoring guide was developed in an iterative manner. First, two graders were used to assess the accuracy of the answers for 16 subjects (8 from the OO treatment, and 8 from the PO treatment) using an initial scoring guide for each case. The inter-rater reliability of the scores assigned by the two graders on that set was over 90 percent, indicating a high degree of agreement. Disagreements were resolved through discussion, which led to a modest refinement of the scoring guide. The final scoring guide was utilized by two graders to assess the performance of all subjects. Only a handful of disagreements remained, which were resolved through discussion, resulting in one consensus set of scores.

In summary, the experiments were designed and conducted so as to ensure their methodological rigor. We selected a target population to which we wish to generalize the results of this study, viz., analysts and designers with some prior experience in process-oriented modeling [8]. We then used the principle of randomization to eliminate the possible confounding effects of nuisance variables. The two treatments, i.e., the OO and PO models, were informationally equivalent. Replication helped address the criticism of low generalizability leveled against experiments, and contributed to the external validity of the study. Finally, valid and reliable dependent variables were used to assess the outcomes of the experiment.

TABLE 2
Descriptive Statistics for Experiment 2

Texas Case				
	PO Model		**OO Model**	
Question	**Mean**	**SD**	**Mean**	**SD**
Q1	5.64	1.11	6.00	1.24
Q2	6.06	1.43	6.44	1.54
Q3	5.71	1.10	4.78	1.31
Q4	6.00	1.17	5.94	1.55
Q5	5.24	0.97	3.50	1.47
Q6	5.76	1.64	3.83	2.09
Q7	6.23	0.83	6.06	1.47
Q8	6.00	1.37	3.56	1.95

Notes:
1. The overall score for each question is reported.
2. Scores were assigned on a scale of 1-7.

TABLE 1
Descriptive Statistics for Experiment 1

ABC Case				
	PO Model		**OO Model**	
Question	**Mean**	**SD**	**Mean**	**SD**
Q1	4.39	1.14	4.56	1.62
Q2	3.78	2.13	5.50	2.15
Q3	5.22	1.48	3.44	1.19
Q4	5.11	1.40	4.78	1.48
Q5	4.89	1.23	4.56	2.14
Q6	6.00	0.91	4.94	1.86
Q7	4.94	1.89	5.11	1.78
Q8	3.17	2.36	1.56	1.34

Notes:
1. The overall score for each question is reported.
2. Scores were assigned on a scale of 1-7.

5 Results

5.1 Statistical Testing

Descriptive data for the two experiments are presented in Table 1 and Table 2. We ran *t* tests on the comprehension scores for the individual questions to determine if an OO model or a PO model is easier to understand. The results of the tests are summarized in Table 3 and Table 4. As the total score—generated by adding together the individual scores on the parts—and the overall score for each question produced the same results, only the overall score results are reported.[1] Except for Question 2 of the ABC case, there was no significant difference in accuracy of the answers to the structure-oriented questions. For question 2 of the ABC case, using an OO model resulted in more accurate answers than using a PO model. Research Question 1 is, therefore answered in the positive for only one of the four simple, structure-oriented questions. None of the four process-oriented questions showed any differences in terms of comprehension accuracy. Research Question 2, therefore, was answered in the negative.

For the more complex questions, which involved both structure and process-oriented aspects, significant differences were observed for seven out of the eight questions—there was no difference in accuracy for Question 5 of the ABC case. For all the seven questions that showed a difference, using the PO model resulted in better comprehension than using the OO model, suggesting that the former was better than the latter in terms of interpreting complex questions. Research Question 3, therefore, was answered in the affirmative, indicating strongly that PO models are easier to comprehend than OO models for more complex questions.

1. The fact that both scores produced identical results further points to the reliability of the measures.

TABLE 3
Comprehension Performance Using the OO Model vs. the PO Model: Results for Experiment 1 (ABC Case)

Question Number/Type		t-value	p-value
Q1	(structure oriented)	0.36	.72
Q2	(structure oriented)	2.42	**.02**
Q3	(both structure and process)	-3.96	**.00**
Q4	(process oriented)	-0.93	.36
Q5	(both structure and process)	-0.57	.57
Q6	(both structure and process)	-2.16	**.04**
Q7	(process oriented)	0.27	.79
Q8	(both structure and process)	-2.52	**.02**

Notes
1. Sample size was 18 for the OO model and 18 for the PO model.
2. 2-tailed p-values are reported.
3. Mean differences were computed as OO model − PO model; therefore, a positive t-value indicates superior performance with the OO model.

TABLE 4
Comprehension Performance Using the OO Model vs. the PO Model: Results for Experiment 2 (Texas Case)

Question Number/Type		t-value	p-value
Q1	(structure oriented)	0.89	.38
Q2	(structure oriented)	0.76	.45
Q3	(both structure and process)	-2.26	**.03**
Q4	(process oriented)	-0.12	.90
Q5	(both structure and process)	-4.10	**.00**
Q6	(both structure and process)	-3.03	**.00**
Q7	(process oriented)	-0.44	.66
Q8	(both structure and process)	-4.27	**.00**

Notes
1. Sample size was 18 for the OO model and 17 for the PO model.
2. 2-tailed p-values are reported.
3. Mean differences were computed as OO model − PO model; therefore, a positive t-value indicates superior performance with the OO model.

While the results of the statistical tests provide valuable initial insights into the relative efficacy of OO and PO methods for model comprehension, we sought to gain a more complete understanding of why subjects performed the way they did. To that end, we conducted a qualitative pattern analysis of scores received by subjects who used the OO model vs. those who used the PO model.

5.2 Qualitative Pattern Analysis

For each of these two subject groups, we calculated the total of the scores received by the subjects in every individual part of an answer to a question. We examined those individual parts for which the two total scores, one for each subject group, were found to be significantly different, implying a noticeable difference in comprehension. The individual parts of all the answers for the ABC case and the Texas case are listed in Appendix C and Appendix D, respectively.

Recall that the score for an individual part of an answer could be 0, 0.5, or 1. Recall also that, for the ABC case, there were 18 subjects for both the OO model and the PO model and that, for the Texas case, there were 18 subjects for the OO model and 17 for the PO model. We used the following heuristic to identify individual parts of an answer for which the two total scores mentioned above were significantly different: For the ABC case, all parts for which the OO total differed from the PO total by at least 5 were selected. For the Texas case, all parts for which either the OO total exceeded the PO total by at least 6 or the PO total exceeded the OO total by at least 4 were selected. Table 5 and Table 6 show all answer parts so selected; they also indicate which total score, OO or PO, was found to be higher in each case. The following observations can be made based on these results.

5.2.1 *ABC Case*

Q2a, b, c: The hierarchy was very evident and highly salient in the OO diagram. Subjects using the PO model, however, needed to examine the data dictionary carefully to identify all aspects of the hierarchy.

Q3b, e, f, g, h: Visual examination of the PO model immediately highlighted the external entities and data stores that interacted with any given process. For the OO representation, however, it was necessary to scan both the graphical and textual descriptions in order to

TABLE 5
Pattern Analysis for Experiment 1

ABC Case	
Question Number	**Pattern**
Q2a	OO > PO
Q2b	OO >PO
Q2c	OO > PO
Q3b	PO > OO
Q3e	PO > OO
Q3f	PO > OO
Q3g	PO > OO
Q3h	PO > OO
Q5a	OO > PO
Q5b	OO > PO
Q5d	PO > OO
Q6a	PO > OO
Q6b	PO > OO
Q8a	PO > OO
Q8b	PO > OO

Notes:
1. OO = object oriented, PO = process oriented.

TABLE 6
Pattern Analysis for Experiment 2

Texas Case	
Question Number	**Pattern**
Q3b	PO > OO
Q3d	PO > OO
Q5a	PO > OO
Q5b	PO > OO
Q6a	PO > OO
Q6b	PO > OO
Q6d	PO > OO
Q8b	PO > OO

Notes:
1. OO = object oriented, PO = process oriented.

comprehend how the methods and the objects interacted, thereby increasing cognitive complexity.

Q5a, b: The graphical inheritance structure of the OO model facilitated recognition of the fact that lower-level objects in a generalization hierarchy are specializations of a higher-level object. In the PO model, although the graphical description clearly showed that calc_extra_income_1 and calc_extra_income_2 used information from EXECUTIVE and WORKER, inheritance was obfuscated in the data dictionary, and apparently subjects were unable to recognize that EXECUTIVE and WORKER were employees, too.

Q6a, b: Significant performance difference was observed in instances where there were multiple outflows of documents and multiple destinations originating from a single activity, increasing the complexity of the OO representation considerably.

Q8a, b: To respond to this question using the PO model, subjects needed to access only the graphical representation. With the OO model, however, they had to follow a complex chain beginning with the meta methods (to identify the methods involved in the two major functions), moving on to the graphical representation (to determine what objects the methods belonged to), and then to the atomic methods (to determine what additional objects were involved). Finally, they needed to intersect the two sets of objects. Here the number of accesses required seemed to introduce cognitive overload, resulting in an increased propensity to make an error.

5.2.2 Texas Case

Q3b: Apparently, subjects missed this item in the OO representation because there was no direct connection between OWNER and the relevant methods in the graphical model; this information had to be derived from the meta methods.

Q3d: A plausible reason for why subjects missed this item in the OO representation could be that this was an additional object listed in the atomic methods section; there was no indication of a link between the methods involved in title issuance and the REGISTRATION object in the graphical representation.

Q5a, b: In both these parts, methods beyond the ones directly associated with the VEHICLE object, which would use or modify information about vehicles, had to be inferred by scanning the atomic method specifications. Subject comprehension seemed to suffer when more than one object was associated with a method in the OO representation.

Q6a, b, d: In the OO representation, there was no direct connection between the OWNER object and any other object. The involvement of OWNER was only evident through scanning the meta methods as OWNER was the source and the destination of several documents. Further, the fact that commercial truck is a type of vehicle, and therefore, all documents sent or received by the owner of a vehicle would apply to the owner of a commercial truck was perhaps not immediately obvious to many subjects.

In summary, the statistical tests and the qualitative analysis appear to collectively suggest that, for relatively complex questions, the subjects in this sample found PO models easier to comprehend than OO models. We now discuss these findings and identify the future research directions.

6 Discussion and Conclusions

Among the simpler questions, the effects of cognitive fit were manifest in only one of the four structure-oriented questions, where using an OO model resulted in more accurate answers than using a PO model. We conjecture that comprehending simply structure or behavior is a relatively simple task compared to understanding both of them together; in other words, the tasks were not complex enough for differences in comprehension performance to surface.

Although the OO and PO models used for each of the two tasks are *informationally* equivalent, from our experiments it appears that a PO model is *computationally* more efficient than the corresponding OO model, but only for complex questions involving both structure and behavior. This pattern was also observed in the qualitative analysis where we noted decreased response accuracy for the questions involving the perusal of several different components of the model. One possible interpretation is that humans may find it easier to understand a model that emphasizes processes rather than data. The processes in a DFD may be better in terms of guiding shifts in attention from one portion of the model to another than the objects in an OO class diagram, giving rise to computational inequivalence between the two representations. As Larkin and Simon [23, pp. 67-68] note, the nature of the *attention management system* that "determines what portion of the data structure is currently attended to... depends crucially on the linkages provided in the data structure."

Several avenues for future research remain. We believe that there is a compelling need for systematic investigation into the relative strengths and weaknesses of alternative systems analysis and design methods—not only from a technical standpoint, but also from the perspective of their usability by systems designers and other key stakeholders. Beginning with replication using lab experiments of the kind described here, the next logical step would be to extend such work into the field by using systems developers as subjects. For such studies, protocol analysis could be employed as an additional data-gathering strategy, because it provides richer insights than those typically obtained through factor-oriented studies. Field experiments might be complemented with in-depth case studies that incorporate additional contextual information and shed further light on the barriers to usability.

OO proponents might argue that models developed using the Unified Modeling Language (UML)—a language that was recently endorsed as a standard for OO modeling [13], [16]—are easier to understand than the OO models investigated in this study because of UML's emphasis on behavioral aspects, through use-case diagrams at a high level and sequence diagrams at a more detailed level. Future research studies could fruitfully examine subject performance using models expressed in UML or in other OO notations.

We used the theoretical paradigm of cognitive fit to frame our research questions. This paradigm has proved to be powerful for explaining the underlying *processes* in problem solving in several research studies (e.g., [1], [2], [32]). There are, however, alternative theories that might be used to extend our understanding of human information processing in the context of model comprehension. The use of multiple theories will allow us to investigate the phenomenon from different perspectives, thereby extending our cumulative knowledge.

From the standpoint of practice, model comprehension is critical during requirements analysis because it helps maintain the flow of communication between analysts and users. The results of this study point to areas around which future training should be devoted so as to promote better comprehension and communication, resulting in improved system quality.

Prior training in PO modeling could also be one of the factors contributing to better performance with a PO model. However, this situation is representative of the present-day workforce where most systems analysts and designers have a background in traditional PO modeling. Changing their mindset with a view toward improving comprehension and communication using OO models is a major challenge that needs to be addressed from a training standpoint.

Appendix A
Questions for the ABC Case

1. List all the attributes that describe a supervising manager.
2. Describe the hierarchy of employees, i.e., the different types of employees and how they are related.
3. List the individuals and/or organizational units that are involved in the calculation of extra income for employees, as well as the data sources that are used or modified in the relevant activities.
4. List all the activities that are involved in the processing and disbursing of salary.
5. List all the activities that directly use or modify the information about employees.
6. List all the documents that flow in the system, as well as their destinations (destinations could be individuals or organizational units).
7. List all the activities that are involved in the adjustment of salary for a VP due to extra income.
8. Consider two major functions of the system—payment of regular salary and payment of extra income. List the data sources that are common to the activities associated with **both** these functions.

Appendix B
Questions for the Texas Case

1. In two separate lists, show all the attributes that describe a trailer and those that describe a commercial truck.
2. Describe the hierarchy of vehicles, i.e., the different types of vehicles and how they are related.
3. List the individuals who are involved in the issuance of a new vehicle title, as well as the data sources that are used or modified in the relevant activities.
4. List all the activities that are involved in the title issuance for a trailer.
5. List all the activities that directly use or modify the information about vehicles.
6. List all the documents that the owner of a commercial truck receives or sends, as well as their sources and destinations, respectively, in connection with the annual registration activity (sources and destinations could be individuals or organizational units).
7. List all the activities that are involved in the registration renewal for a car.

8. Consider two major functions of the system–registration renewal and title issuance. List the data sources that are common to the activities associated with **both** these functions.

Appendix C

Answer Parts for the ABC Case

1. List all the attributes that describe a supervising manager.
   a. all employee attributes
   b. position from executive
   c. extra_income does not exist
   d. shift_worked exists
2. Describe the hierarchy of employees, i.e., the different types of employees and how they are related.
   a. EMPLOYEE and EXECUTIVE; WORKER
   b. EXECUTIVE and MANAGER; VP; PRESIDENT
   d. MANAGER and SUPERVISING MANAGER
3. List the individuals and/or organizational units that are involved in the calculation of extra income for employees, as well as the data sources that are used or modified in the relevant activities.
   a. WORKER
   b. SUPERVISING MANAGER
   c. PAYROLL
   d. SALES
   -
   e. OT_SLIP
   f. SALES
   g. WORKER
   h. EXECUTIVE
4. List all the activities that are involved in the processing and disbursing of salary.
   a. calc_salary
   b. pay_salary
   c. print_salary_statement
5. List all the activities that directly use or modify the information about employees.
   a. calc_extra_income_1
   b. calc_extra_income_2
   c. calc_salary
   d. pay_salary
   e. print_salary_statement
   f. adjust_salary_and_tax
6. List all the documents that flow in the system, as well as their destinations (destinations could be individuals or organizational units).

a. check	EMPLOYEE
b. deposit_advice	BANK
c. salary_stmt	EMPLOYEE
d. profit_stmt	PAYROLL
e. ot_slip	SUPERVISING_MGR
f. ot_slip	PAYROLL

7. List all the activities that are involved in the adjustment of salary for a VP due to extra income.
   a. prepare_profit_statement
   b. calc_extra_income_1
   c. adjust_salary_and_tax
8. Consider two major functions of the system—payment of regular salary and payment of extra income. List the data sources that are common to the activities associated with both these functions.
   a. WORKER
   b. EXECUTIVE

Appendix D

Answer Parts for the Texas Case

1. In two separate lists, show all the attributes that describe a trailer and those that describe a commercial truck.
   a. all vehicle attributes;
   b. number_passengers and diesel do not exist;
   c. body_number and length exist.
   -
   d. all vehicle attributes
   e. temp_gross_wt exists
   f. emission_reqmt exists
2. Describe the hierarchy of vehicles, i.e., the different types of vehicles and how they are related.
   a. VEHICLE and CAR; TRUCK; MOTORCYCLE; TRAILER
   b. TRUCK and COMMERCIAL; NON-COMMERCIAL
3. List the individuals who are involved in the issuance of a new vehicle title, as well as the data sources that are used or modified in the relevant activities.
   a. TITLE CLERK
   b. OWNER
   -
   c. TITLE
   d. REGISTRATION
   e. VEHICLE
4. List all the activities that are involved in the title issuance for a trailer.
   a. verify_ownership
   b. issue_title
5. List all the activities that directly use or modify the information about vehicles.
   a. issue_new_registration
   b. calculate_renewal_fee
   c. verify_ownership
   d. confirm_emission_check
6. List all the documents that the owner of a commercial truck receives or sends, as well as their sources and destinations, respectively, in connection with the annual registration activity (sources and destinations could be individuals or organizational units).

a. sends	renewal_notice to	R. CLERK
b. sends	payment to	R. CLERK
c. receives	renewed_registration from	R. CLERK
d. receives	renewal_notice from	R. CLERK

7. List all the activities that are involved in the registration renewal for a car.
   a. calculate_renewal_fee
   b. issue_renewal_notice
   c. issue_new_registration

8. Consider two major functions of the system—registration renewal and title issuance. List the data sources that are common to the activities associated with both these functions.
   a. VEHICLE
   b. REGISTRATION

ACKNOWLEDGMENTS

The authors would like to thank the editors of this special issue and two anonymous reviewers for their helpful comments.

REFERENCES

[1] R. Agarwal, A.P. Sinha, and M. Tanniru, "Cognitive Fit in Requirements Modeling: A Study of Object and Process Methodologies," *J. Management Information Systems,* vol. 13, no. 2, pp. 137–162, 1996.

[2] R. Agarwal, A.P. Sinha, and M. Tanniru, "The Role of Prior Experience and Task Characteristics in Object-Oriented Modeling: An Empirical Study," *Int'l J. Human Computer Studies,* vol. 45, pp. 639–667, 1996.

[3] G. Booch, *Object-Oriented Analysis and Design with Applications,* second ed., Redwood City, Calif.: Benjamin/Cummings, 1994.

[4] F.P. Brooks, *The Mythical Man Month: Essays on Software Engineering.* Reading, Mass.: Addison-Wesley Longamn, 1995.

[5] S.N. Cant, D.R. Jeffrey, and B. Henderson-Sellers, "A Conceptual Model of Cognitive Complexity of Elements of the Programming Process," *Information and Software Technology,* vol. 37, no. 7, pp. 351–62, 1995.

[6] P. Coad and E. Yourdon, *Object-Oriented Analysis,* second. ed., Englewood Cliffs, N.J.: Yourdon Press/Prentice Hall 1991.

[7] L.L. Constantine, "Object-Oriented and Structured Methods: Toward Integration," *American Programmer,* vol. 2, no. 7/8, pp. 34–40, Aug. 1989.

[8] T.D. Cook and D.T. Campbell, *Quasi Experimentation: Design and Analysis Issues for Field Settings.* Boston, Mass.: Houghton Mifflin 1979.

[9] L.J. Cronbach, *Essentials of Psychological Testing,* third ed., New York: Harper and Row, 1970.

[10] T. DeMarco, *Structured Analysis and System Specification.* Englewood Cliffs, N.J.: Prentice Hall, 1978.

[11] T. DeMarco, *Controlling Software Projects: Management, Measurement, and Estimation.* New York: Yourdon Press, 1982.

[12] H.J. Einhorn and R.M. Hogarth, "Behavioral Decision Theory: Processes of Judgment and Choice," D.E. Bell, H. Raiffa, and A. Tversky, eds., *Decision Making: Descriptive, Normative, and Prescriptive Interactions.* Cambridge, England: Cambridge Univ. Press pp. 113–146, 1988.

[13] H. Eriksson and M. Penker, *UML Toolkit.* New York: John Wiley & Sons, 1998.

[14] M.E. Fayad, W. Tsai, and M.L. Fulghum, "Transition to Object-Oriented Development," *Comm. ACM,* vol. 39, no. 2, pp. 108–121, 1996.

[15] R.G. Fichman and C.F. Kemerer, "Object-Oriented and Conventional Analysis and Design Methodologies: Comparison and Critique," *Computer,* vol. 25, no. 10, pp. 22–39, Oct. 1992.

[16] M. Fowler, *UML Distilled: Applying the Standard Object Modeling Language.* Reading, Mass.: Addison-Wesley, 1997.

[17] W. Gemino and Y. Wand, "Empirical Comparison of Object-Oriented and Dataflow Models," *Proc. 18th Int'l Conf. Information Systems,* pp. 446-447, Dec. 1997.

[18] J. Iivari, "Object-Orientation as Structural, Functional and Behavioural Modelling: A Comparison of Six Methods for Object-Oriented Analysis," *Information and Software Technology,* vol. 37, no. 3, pp. 155–163, 1995.

[19] I. Jacobson, *Object-Oriented Software Engineering: A Use Case Driven Approach.* Reading, Mass.: Addison-Wesley, 1992.

[20] S.L. Jarvenpaa, "The Importance of Laboratory Experimentation in IS Research," *Comm. ACM,* vol. 31, no. 12, pp. 1,504–1,507, 1988.

[21] S.L. Jarvenpaa, G.W. Dickson, and G. DeSanctis, "Methodological Issues in Experimental IS Research: Experiences and Recommendations," *MIS Quarterly,* vol. 9, no. 2, pp. 141–56, 1985.

[22] Y. Kim and S. March, "Comparing Data Modeling Formalisms," *Comm. ACM,* vol. 38, no. 6, pp. 103–115, 1995.

[23] J.H. Larkin and H.A. Simon, "Why a Diagram Is (Sometimes) Worth Ten Thousand Words," *Cognitive Science,* vol. 11, pp. 65–99, 1987.

[24] T. Moynihan, "An Experimental Comparison of Object-Orientation and Functional Decomposition as Paradigms for Communicating System Functionality to Users," *J. Systems and Software,* vol. 33, pp. 163–169, 1996.

[25] A. Newell and H.A. Simon, *Human Problem Solving.* Englewood Cliffs, N.J.: Prentice Hall, 1972.

[26] S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas.* New York: Basic Books, 1980.

[27] D.E. Perry, N.A. Staudenmayer, and L.G. Votta, "People, Organizations, and Process Improvement," *IEEE Software,* vol. 11, pp. 36–45, July 1994.

[28] A. Porter and L. Votta, "What Makes Inspections Work?" *IEEE Software,* vol. 14, pp. 99–102, Nov./Dec. 1997.

[29] B. Ratcliffe and J.I.A. Siddiqui, "An Empirical Investigation into Problem Decomposition Strategies Used in Program Design," *Int'l J. Man-Machine Studies,* vol. 22, no. 1, pp. 77–90, 1985.

[30] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-Oriented Modeling and Design.* Englewood Cliffs, N.J.: Prentice Hall, 1991.

[31] P. Shoval and I. Frumermann, "OO and EER Conceptual Schemas: A Comparison of User Comprehension," *J. Database Management,* vol. 5, no. 4, pp. 28–38, 1994.

[32] A.P. Sinha and I. Vessey, "Cognitive Fit: An Empirical Study of Recursion and Iteration," *IEEE Trans. Software Eng.,* vol. 18, no. 5, pp. 368-379, May 1992.

[33] P. Slovic and S. Lichtenstein, "Preference Reversals: A Broader Perspective," *American Economic Review,* vol. 73, no. 4, pp. 596–605, 1983.

[34] E. Soloway, J. Bonar, and K. Ehlrich, "Cognitive Strategies and Looping Constructs: An Empirical Study," *Comm. ACM,* vol. 26, no. 11, pp. 485–492, 1983.

[35] A. Tversky, S. Sattath , and P. Slovic, "Contingent Weighting in Judgment and Choice," *Psychological Review,* vol. 95, no. 3, pp. 371–384, 1988.

[36] I. Vessey, "Cognitive Fit: A Theory-Based Analysis of the Graphs versus Tables Literature," *Decision Sciences,* vol. 22, no. 2, pp. 219–240, 1991.

[37] I. Vessey and S. Conger, "Requirements Specification: Learning Object, Process, and Data Methodologies," *Comm. ACM,* vol. 37, no. 5, pp. 102–113, 1994.

[38] I. Vessey and D. Galletta, "Cognitive Fit: An Empirical Study of Information Acquisition," *Information Systems Research* vol. 2, no. 1, pp. 63–84, 1991.

[39] I. Vessey and R. Weber, "Structured Tools and Conditional Logic: An Empirical Investigation," *Comm. ACM,* vol. 29, no. 1, pp. 48–57, 1986.

[40] S. Wang, "Two MIS Analysis Methods: An Experimental Comparison," *J. Education for Business,* pp. 136–141, Jan./Feb. 1996.

[41] C. Welty and D.W. Stemple, "Human Factors Comparison of a Procedural and a Non-Procedural Query Language," *ACM Trans. Database Systems,* vol. 6, no. 4, pp. 626–649, 1981.

[42] S.B. Yadav, R.R. Bravocco, A.T. Chatfield, and T.M. Rajkumar, "Comparison of Analysis Techniques for Information Requirement Determination," *Comm. ACM,* vol. 31, no. 9, pp. 1,090–1,097, 1988.

Ritu Agarwal received her MS degree in computer science and her PhD degree in management information systems (MIS), both from Syracuse University, and an MBA from the Indian Institute of Management, Calcutta. She is currently an associate professor of MIS in the Robert H. Smith School of Business at the Uniersity of Maryland. Prior to joining the University of Maryland, Dr. Agarwal taught in the School of Business Administration at Florida State University and at the University of Dayton, and also in the Stern School of Businesss at New York University. She has published widely on the topic of information technology management in journals such as *Information Systems Research*, *MIS Quarterly*, *Journal of Management Information Systems*, *Decision Sciences*, *IEEE Transactions*, *Decision Support Systems*, *Information and Management*, and *Knowledge-Based Systems*. She has also made presentations at a variety of national and international conferences. Her current research focuses on individual learning and organizational adoption and diffusion of new information technologies, as well as object-oriented technologies. She serves as an associate editor for *MIS Quarterly* and the *International Journal of Human-Computer Studies*. Dr. Agarwal is a member of the ACM, INFORMS, and the Academy of Management.

Prabuddha De received his PhD degree in industrial administration from Carnegie Mellon University in 1979. Dr. De is Sherman-Standard Register Chair in management information systems at the University of Dayton. Previously, he taught at Ohio State University and Cornell University. He currently serves as an associate editor, on the advisory board, or on the editorial board for a number of major journals in the MIS area, including *Information Systems Research*, *Management Science*, *Decision Sciences*, *Journal of MIS*, and *Decision Support Systems*. He is president of the Workshop on Information Technologies & Systems (WITS) for 1998–2000 and program chair for the International Conference on Information Systems (ICIS) for 1999.

Atish P. Sinha received his PhD degree in business, with a concentration in artificial intelligence, from the University of Pittsburgh. Dr. Sinha is an associate professor of management information systems in the School of Business Administration at the University of Dayton. His current research interests are in the areas of case-based reasoning, expert systems, data mining, object-oriented modeling, and human factors in software engineering.