THE HOLE IN THE WHOLE: A RESPONSE TO ALLEN AND MARCH

Graeme Shanks
Department of Computing and Information Systems, University of Melbourne,
Parkville, Victoria, AUSTRALIA 3052 {gshanks@unimelb.edu.au}

Ron Weber
Faculty of Information Technology, Monash University,
Caulfield East, Victoria, AUSTRALIA 3145 {Ron.Weber@monash.edu}

Allen and March provide a critique of one of our papers in which we argue composites should be represented as entities/objects in a conceptual model rather than relationships/associations (Shanks et al. 2008). They contend we have addressed a non-issue. Furthermore, they argue our theoretical rationale and empirical evidence have flaws. In this paper, we provide a response to their arguments. We show that the issue we address is substantive. We show, also, that our theoretical analysis and empirical results are robust. We find, instead, that Allen and March’s theoretical arguments and empirical evidence have flaws.

Keywords: Conceptual modeling, empirical research, ontology, information systems development, aggregation, composition, UML, entity–relationship model

Introduction

In this paper, we provide a response to a critique by Allen and March (2012) of a paper we published in the MIS Quarterly in which we argued composites should be represented as entities or objects in conceptual models rather than relationships or associations (Shanks et al. 2008). Although Allen and March agree with us that conceptual modelers should not represent composites as relationships, they argue our work addresses a non-issue and is flawed both theoretically and empirically.

We respond to a number of matters raised by Allen and March. In the interests of brevity, we have not addressed all of the matters they have raised. Rather, we have focused on those we regard as more substantive. The structure of our response follows the major criticisms Allen and March have levied at our work.

1Dale L. Goodhue was the accepting senior editor for this paper. Sandeep Purao served as the associate editor.
issue relating to the merits of providing a theoretical justification for different conceptual modeling practices.

Appealing to the Authors: A Valid Criticism?

We argue that asking the authors of these diagrams about the meaning they ascribed to them and then claiming our research is problematic because our interpretation of the authors’ examples does not match the authors’ intended meaning does not provide a valid basis for debunking our work. In matters of communication, one should not assume the “listener” must always be wrong if they do not understand the “speaker.” Moreover, as researchers we are ever skeptical about wisdom in hindsight (e.g., claiming the intended semantics of a conceptual model ought to have been clear to an informed reader when the reader points out the model can be interpreted in multiple ways).

In any event, the primary motivation for our work is to help the creators of conceptual models make clear the semantics of a domain they are intending to convey to the readers of a conceptual model. Even when a conceptual modeling grammar is used in a highly disciplined way to try to make a domain’s semantics clear (the denotational semantics), for many reasons substantial variations in the ways scripts are interpreted (the connotational semantics) might still occur (Burton-Jones et al. 2009; Burton-Jones et al. 2010). To the extent readers of conceptual models have more degrees of freedom to appropriate the meaning of a conceptual model, the consequences for subsequent development, maintenance, and use of a system become more deleterious. Creators of conceptual models need to take great care, therefore, that they convey the underlying meaning they ascribe to a domain as unambiguously as possible.

The Two Examples

Aside from contacting the authors of the two examples we used in our paper to obtain the “correct” interpretation of the conceptual models, Allen and March also argue their own interpretation of the models makes it clear our interpretations are misplaced.

Consider, first, Elmasri and Navathe’s (2004) conceptual model of a relationship class named Committee linking an entity class named Faculty and an entity class named Grad_Student.2 [SE Note #1: This model from Elmasri and Navathe (2006, page 102, Figure 4.9) was first included in Shanks et al. 2008 as their Figure 1, and then reproduced in Allen and March’s 2012 research note as their Figure 1.]

For many years, conceptual modelers have been admonished to use a singular noun to label entity classes and a verb or verb phrase to label relationship classes (see, for example Connolly and Begg 2010, pp. 324, 326; Hoffier et al. 2007, pp. 98, 121). How, then, should a reader of a conceptual model who is familiar with this convention interpret a relationship class that is given an entity class label? Is it representing a mutual property of two things? Is it representing a thing? Allen and March (p. 948) concede Elmasri and Navathe’s (2004) choice of the label for the relationship named Committee is poor.

[SE Note #2: In reference to “Figure 1,” Allen and March observe that]

the cardinality constraints, M and N in the diagram indicate that one faculty member may be associated with many graduate students and one graduate student may be associated with many faculty members....[and] all that can be known from the diagram is that faculty members and graduate students are associated with each other and that “committee” is deemed an appropriate, although potentially misleading, name for that relationship.

They go on to talk about the common meaning.]

Nonetheless, they then argue (p. 948):

If the common meaning of academic committees is assumed (i.e., each student has one committee and each faculty member can serve on many committees) then Figure 1 does allow for reasoning about committees [our emphasis on the words in italics except for “reasoning”].

We wonder what they mean by the words “common meaning.” It is true that some advisory relationships exist where some faculty members serve on multiple advisory committees and a graduate student is a member of only one advisory committee.3 Our experience, however, is that other forms of advisory committees are “common.” For instance, some students work on a set of related research projects as

3Indeed, we show these semantics represented explicitly in Figure 3 of our paper. Nonetheless, in our paper we state an important caveat in representing the semantics in this way—namely, we have made assumptions about the semantics that Elmasri and Navathe may be seeking to represent that may or may not be true (Shanks et al. 2008, p. 555).
they prepare their theses. As a result, an advisory committee is formed that comprises multiple faculty members (or just one faculty member) and multiple students (i.e., the students are supervised as a research team). Moreover, even if we accept Allen and March’s “common meaning,” how does Elmasri and Navathe’s model manifest Allen and March’s stated constraint that a student can be a member of only one advisory committee?

We also wonder what Allen and March mean by the words “reasoning about committees.” For instance, consider the following question: Can a student be a member of more than one committee? We contend this question cannot be answered using Elmasri and Navathe’s conceptual model. Presumably, Allen and March’s retort would be that “common meaning” indicates a student can be a member of only one committee. In contrast, our “common meaning” includes situations where both faculty members and students can be members of multiple advisory teams (because they work on interdisciplinary teams that work to assist both faculty members and students advance their research). In any event, our question for Allen and March is the following: Based on Elmasri and Navathe’s conceptual model, exactly what reasoning can occur using the relationship labeled Committee where the response relies only on the semantics manifested in the conceptual model?

Consider, next, Teorey et al.’s example. [SE Note #3: This model from Teorey et al. (2006, page 91) was first included in Shanks et al. 2008 as their Figure 2, and then reproduced in Allen and March’s 2012 research note as their Figure 2.] Allen and March (p. 948) state:

> the purpose of that figure is not to explicate the semantics of composition but rather to illustrate rules for transforming recursive one-to-many relationships into SQL for implementation (Teorey et al. 2006). The use of the diamond was not intended to represent a composition (aggregation) relationship, but rather an association relationship as in the corresponding ER diagram from the prior page of the textbook.

We find Allen and March’s statement puzzling. If the diagram was intended to represent an association relationship and not a composition relationship, why then was a composition symbol used (particularly in a textbook intended for students who presumably are seeking to learn how to prepare high-quality conceptual models)?

Subsequent to their comments above, Allen and March proceed to provide their own interpretation of what Teorey et al.’s model must mean. We have an interpretation of Teorey et al.’s model that differs from Allen and March’s interpretation. In particular, we point out that on page 90, Teorey et al. show an ER model and on page 91 they show a UML model to represent the following text: “Engineers are divided into groups for certain projects. Each group has a leader.” They show a recursive relationship/association on an engineer entity/class. It is clear that they intend the relationship to represent a group for a project. In their UML diagram, they use an aggregation symbol on the recursive relationship. On page 36 they define an aggregation (hollow diamond) in the following way: “Aggregation indicates ‘part of’ associations, where the parts have an independent existence.” An engineer clearly cannot be part of another engineer. The aggregation symbol used in the diagram means, therefore, that Teorey et al. intend the association to stand for a composite—namely, a group of engineers.

### A Theoretical Matter

Even if most, if not all, conceptual modelers agreed that representing composites as relationships is poor practice, our study provides a theoretical explanation of why the practice is problematic. In our teaching and research on conceptual modeling, we have examined many database textbooks. We have observed that few provide any deep theoretical basis to guide practitioners in their conceptual modeling work. Instead, many are primarily “cookbooks.” They describe the syntax and semantics of conceptual modeling grammars. They also show how the grammars might be used via some domain examples. From the examples, readers often must glean themselves what constitutes good conceptual-modeling practice. Moreover, few are explicit about how composites should be modeled and why they should be modeled in a particular way.

For this reason, our study has pedagogical value. For novice conceptual modelers, it makes clear how composites should be modeled. It also explains theoretically why composites should be represented as object/entity classes (whatever the modeling grammar used). For expert conceptual modelers, it makes tacit knowledge explicit. It provides a deep rationale for the approach to modeling composites they hopefully employ.

---

4In our professional roles, we have experience not only of supervision arrangements in our own faculty but also of supervision arrangements in other faculties (via our membership of university-wide committees).

5In Elmasri and Navathe’s conceptual model, a 1:N relationship named “advisor” (again, a poor choice of a name for a relationship) shows a student can have only one “advisor,” but this does not preclude the student’s advisor nor the student being a member of multiple advisory committees.
Problematic Theoretical Foundations

Allen and March contend our use of Bunge’s (1977) ontology as the theoretical foundation for our work is problematic because

Bunge’s ontology is intended to represent concrete objects (things) that possess “substantial properties.” Concrete objects exist objectively in space and time, independent of human interpretation and ascription of meaning. Conceptual models, however, must frequently represent conceptual objects and attributes that exist subjectively, are based exclusively on human interpretation, beliefs and collective agreement, and convey ascribed meaning [our emphasis on the word “represent”].

Allen and March argue our use is a departure from Bunge’s ontology. They go on to say,

In our opinion, this deviation severely reduces any guidance a modeler would obtain from Bunge’s ontology in the identification and classification of phenomena to be represented in a conceptual model.

Here, in our view, lies the heart of Allen and March’s objections to our work. It is our use of Bunge’s ontology as a theoretical basis for conceptual modeling that they find objectionable. They contend that Bunge’s ontology applies only to “concrete objects,” whereas conceptual models must sometimes represent “subjective objects.”

Bunge (p. 58) makes a very clear distinction between conceptual objects and substantial objects (or things). The real world of substantial objects ultimately is unknowable. As a result, humans use conceptual objects to express their understanding of the real world. This dichotomy is the reason Bunge (pp. 57-58) carefully distinguishes between properties (belonging to substantial entities) and attributes (belonging to models of substantial entities):

Surely we know properties only as attributes or predicates, i.e. as components of our views of things. Still, we must distinguish the object represented from its representations if we wish to account for discovery (or invention) and ignorance, truth and error (p. 59).

In many places throughout his ontology, Bunge reiterates the notion that humans can understand the real world only through their models of the real world. He also fully understands that ultimately these models are transitory because human knowledge advances (models are social constructions).

For instance, when he builds his ontological theory of “things” (or substantial objects), he states (p. 119):

Theoretical science and ontology handle not concrete things but concepts of such [our emphasis], in particular conceptual schemata [our emphasis] sometimes called model things. Our construal of a thing as a substantial individual together with the set of all its properties…is of course such a model thing.

In short, to say that Bunge’s ontology cannot be used as a theoretical foundation for conceptual modeling because conceptual models deal with “conceptual objects and attributes” is erroneous. Contrary to Allen and March, we contend that Bunge is very clear in arguing that the only way humans can engage in discourse about or think about concrete things (and events that occur to concrete things) is via the concepts (conceptual models) that humans have devised to describe the things (and the events that occur to them).

[SE Note #4: A case can be made (a la Allen and March) that Bunge excludes concepts as the things being included or modeled in his ontology (see SE Appendix, Selections From Bunge, with quotes from his pages 5, 6, 7, 8, 12, 50, 52, 116, 117, 118, and 119). But certainly (a la Shanks and Weber) Bunge also seems to be saying that, although he is only interested in modeling concrete things, he recognizes that models of reality contain not the concrete things themselves but constructs representing the concrete “things.” The relevant question would seem to be whether Bunge would allow models (which he notes are constructs) to represent not only concrete things but also constructs that are not concrete things, such as a verbal agreement. (See SE Appendix, Selections From Bunge, with quotes from Bunge’s pages 117, 118, 119, 121).]

Whether Bunge’s ontology ultimately proves to be useful as a means of informing conceptual modeling research and practice, however, is another matter (the answer to which most likely will be informed through theoretical and empirical research that is motivated by his ontology and alternative ontologies).

Problematic Experimental Treatments

For several reasons, Allen and March argue the composites in our ontologically clear diagram violate Bunge’s definition of a composite. As a result, they contend our experimental results are flawed.

6See the SE Appendix located in the “Online Supplements” section of the MIS Quarterly’s website (http://www.misq.org).
First, Allen and March (p. 950) argue the composite called Team in Figure 5 of our paper violates Bunge’s definition of a composite

because it permits a Team to be composed of one Team Leader and a minimum of zero Team Members. That is, a Team Leader composed of no individuals other than itself (i.e., having no team members), is represented to be a Team.

Allen and March misunderstand Bunge’s ontology. [SE Note #5: But see the SE Appendix, Bunge’s page 29, Definition 1.1 of composites.] In the context of our diagram (and the domain we are modeling), a Team comprised of only a Team Leader does not mean that the Team Leader is the same thing in the world as the Team itself. In this regard, Figure 5 in our paper shows a Team thing has attributes that are different from a Team Leader thing. For instance, the attribute Date Formed, which is a meaningful attribute of a Team thing, is not a meaningful attribute in terms of a Team Leader thing. In short, their critique of our Team example is misinformed.

Second, Allen and March argue the composites called Project and Project Plan in Figure 5 of our paper violate Bunge’s definition of a composite because they can be “composed of nothing.” Again, their argument is misplaced. The problem with the Project and Project Plan classes is that they use optional mutual attributes and thus they contravene Bunge’s requirement that things in the world possess only “positive” attributes (Bunge 1977, pp. 60-61).7 Elsewhere, one of us (Weber), along with a number of other colleagues, has pointed out that (1) use of optional attributes violates Bunge’s ontology, and (2) constitutes a problematic conceptual modeling practice for a number of reasons (Bodart et al. 2001; Wand et al. 1999). The existence of optional properties implies that subclasses exist—one subclass of things always possesses the “optional” attribute, and the other subclass of things never possesses the “optional” attribute. Thus, in the conceptual model we show in Figure 5 of our paper, the optional mutual attributes imply that subclasses for Project and Project Plan exist. Indeed, given that Project and Project Plan have multiple mutual attributes, depending on their purposes conceptual modelers might choose to define multiple subclasses of these subclasses (Parsons and Wand 2008). At least one of the subclasses defined for Project and Project Plan will contain composite things.

Third, Allen and March (p. 950) argue several of our composites violate Bunge’s definition of a composite because some of their components do not have an existence independent of their composites. For instance, they contend our use of a solid diamond on the association between Requisition Line and Purchase Requisition means an instance of a Requisition Line cannot exist independently of an instance of a Purchase Requisition. They argue that this violates Bunge’s ontology because they contend his ontology requires Requisition Line to have an existence independent of a Purchase Requisition. They claim (p. 6):

In Bunge’s ontology nothing is created or destroyed; concrete objects are composed from existing concrete objects (p. 35). The components of a composite must have independent existence from the composite itself.

It is true that Bunge subscribes to the “law” in physics relating to conservation of energy. Energy might be transferred, however, from one thing to another thing. Also, things can undergo deep change (they gain or lose properties in general) such that they undergo changes of kind (Bunge 1977, pp. 219-222)—in other words, the thing has changed so fundamentally that we model it as a member of a different class.

We do not agree with Allen and March’s claim, however, that Bunge (p. 35) says or implies that “components of a composite must have an independent existence from the composite itself” (our emphasis). As best we can see, these words do not appear on the page they reference, nor do they appear in Bunge’s book. [SE Note #6: Again, see the SE Appendix, Bunge’s page 29, Definition 1.1 of composites.] Moreover, what Allen and March mean by “independent existence” is unclear. If we assume they mean the history (Bunge 1977, pp. 255-262) of the component does not depend on the history of the composite, it is easy to think of counterexamples. For instance, the psychological well being of family members both contributes to and is affected by the overall quality of their family life. In short, a family member does not have an existence that is independent of the family to which they belong. For some purposes, it might be useful to model components as having an existence that is “independent” of the composite, but Bunge would not argue that components being independent of their composite is a universal trait of the world.

In the interests of brevity, we do not consider here some deep ontological issues that pertain to the concept of a UML “composition” (versus the concept of a UML “aggregation”). In short, however, consider the Requisition Line class in Figure 5 of our paper. This class stands for or represents

7Indeed, the model in Figure 5 of our paper (and in Figure 6 of our paper) also violates Bunge’s ontology in a number of other ways. For experimental control reasons, our focus was on having ontological clarity only in terms of using entity classes rather than associations or association classes to represent composites.
things (e.g., inventory items) that have been ordered by an organization. If they have not been ordered, they lose an attribute in general, and thus they experience a change of kind (class). They are things (inventory items) of a different kind (those that have not been ordered). In the context of Allen and March’s logic (with which we disagree), the parts have existence prior to being composed with Requisition Header, but they are parts of a different kind (class) once they are composed with Purchase Requisition. Similarly, the Requisition Header stands for or represents a specific contractual relationship between an organization and a supplier. If the Purchase Requisition does not exist, however, they are an organization and supplier of a different kind—they still exist but they have lost the attribute in general pertaining to the existence of a contract between them.

Fourth, Allen and March (p. 950) argue that the ternary relationships in our ontologically unclear diagram (Figure 6 in our paper) can be interpreted as “representing mutual properties of distinct things rather than as composites.” We agree with them that ternary relationships can be used to represent mutual attributes in general between classes of things. Our argument is, however, that conceptual modelers sometimes use relationships in ways that provide ambiguous representations of real-world semantics. As a result, different readers of a conceptual model can interpret their meaning differently. In this regard, our use of nouns like Team and Project Plan for the labels of the ternary relationships in our ontologically unclear diagram creates the same kind of confusion that occurs when a noun like Committee is used as the label for a relationship between Faculty and Grad Student. As a result, our ternary relationships might be interpreted by users of the unclear diagram as representing a composite rather than a mutual property of several things.

**Problematic Construct Operationalization**

Allen and March (p. 952) “conclude that there are significant problems with the operationalization of the independent variable, ‘construct overload,’” in our research. In a nutshell, they argue that in UML a diamond symbol can only ever represent an association and not a composite object. Our use of a diamond symbol “should be interpreted” (our emphasis), therefore, as representing a relationship and not a composite. How readers of a conceptual model should interpret the model is one matter; how they do interpret the model is another matter. Ideally, the interpretation intended by the creator of the model will match the interpretation made by the reader of the model. For various reasons, this outcome might not occur (see our earlier discussion).

We have operationalized construct overload in our “unclear model” by using a diamond symbol in a way that we hypothesize (according to the theory of ontological clarity) will cause confusion among the readers of the model. Consistent with UML syntax, readers’ first reaction might be that the diamond symbols used in the figure represent associations among three object classes. Even if this were to be the case, we predict they will begin to doubt the veracity of this initial interpretation as they then try to tease out the meaning of the associations (particularly when they note either explicitly or implicitly that the labels given to the diamonds are nouns).

In this light, we reject Allen and March’s criticism (p. 952) that they “see no construct overload in Figure 4” because “all and only ternary relationships intended to communicate the existence of composites are represented using a diamond symbol, [and] all binary relationships…are represented using labeled arcs.” Whether construct overload exists is in the “eyes of the beholder” of a conceptual model, its existence depends on the explicit or implicit ontology that a reader uses to understand a domain and the reader’s knowledge of the syntax of the grammar used to generate scripts to represent the domain. In this regard, participants in our experiment had been trained to have an expectation that a ternary relationship would represent only an association between three object classes and not a composite object class. In their minds, therefore, our prediction is that they would have perceived construct overload to exist (as per our explanation in the previous paragraph).

In short, whether construct overload exists in a grammar needs to be considered from three stakeholders’ perspectives: (1) the designers of the grammar, (2) those who generate sentences (scripts or diagrams) using the grammar (model creators who use the grammar), and (3) those who interpret sentences generated using the grammar (readers of a model created using the grammar). For each stakeholder, the implicit or explicit ontology they use to understand a domain must be considered. The meaning they assign to each symbol/construct in a conceptual modeling grammar must also be considered.9

---

9 On the one hand, one might argue that the aggregation and composition diamonds in UML are redundant symbols. The cardinalities (optional versus mandatory) will show whether an aggregation or composition is being represented. On the other hand, the diamonds are perhaps a useful “signal” that the association class between two classes of entities represents a “part-of” association.

9 Historically, variations in the ways grammars are designed, used, and interpreted have motivated the study of a grammar’s pragmatics (as well as the study of its syntax and semantics).
**Problematic Experimental Implementation**

Allen and March question why we have represented only *Team*, *Project Plan*, and *Purchase Requisition* as diamonds in our ontologically unclear diagram when our ontologically clear diagram contains four aggregate/composite classes (*Team*, *Project Plan*, *Purchase Requisition*, and *Project*). It is true that we have erroneously stated in our paper (p. 561) that each aggregate/composite class has been converted to a diamond—*Project*, indeed, has not been represented by a diamond in the ontologically unclear diagram (Figure 6 in our paper).

Nonetheless, an examination of Figure 6 will show why we have not represented *Project* as a diamond. If this were to occur, we would have had diamonds linked to diamonds in the diagram: *Project Plan* with *Project*, *Phase* with *Project*, and *Team* with *Project*. Having diamonds linked to diamonds violates a well-known grammatical rule in both the entity–relationship and UML grammars. Of course, we could have chosen to violate this rule in our diagram. We did not do so because this grammatical violation was not the focus of our research.

Allen and March also ask why we represented *Phase* as an object class in our ontologically clear diagram (Figure 5 in our paper) but then represented it as a relationship (diamond) in our ontologically unclear diagram (Figure 6 in our paper). They observe that the “phase” class in our ontologically clear diagram is “neither aggregate or composite.” Because we then represented “phase” as a ternary relationship in our ontologically unclear diagram, they argue:

> if ternary relationships are intended to implicitly represent composites…this change has the effect of asserting that projects are composed of phases in the ontologically clear diagram, but that phases are composed of projects (and consumables and key deliverables) in the ontologically unclear diagram (pp. 952-953).

We designed our experimental materials in 2001–2002, and in truth we now cannot recall the specific reason for this decision. Because our participants had been trained in UML just prior to their undertaking the experiment, however, we suspect we were attempting to make the existence of construct overload more apparent in our ontologically unclear diagram by having the grammatical construct of a UML ternary relationship represent *both* a mutual attribute (as in the case of “phase”) and a composite (as in the case of “team”) in the ontologically unclear diagram. Indeed, this is the same rationale used by Allen and March to justify the existence of construct overload in their Figure 7(a) (specifically, they have an association class representing both a “simple association” and a “composite” in their figure).

Moreover, Allen and March seem to imply that our use of an association (relationship) construct in our ontologically unclear diagram must mean we are representing a composite. Clearly, we cannot have this intention whenever we use an association construct; otherwise, we could not model mutual attributes. In the types of ontologically unclear conceptual models that are our focus, association classes might sometimes represent mutual attributes and sometimes represent composites. This form of construct overload is the fundamental focus of our research.

We also find Allen and March’s comment curious because in the previous section of their paper they have argued they see no construct overload in our ontologically unclear diagram because “all and only ternary relationships intended to communicate the existence of composites are represented using the diamond symbol” (p. 952). Yet now they say (p. 952), in relation to our ontologically unclear diagram, that “the Phase relationship in Figure 4 corresponds to a class in Figure 3 that is neither aggregate or composite” (our emphasis). In short, Allen and March have contradicted their own argument.

Allen and March then argue (p. 953) “every part of the experimental task is influenced by this error in the experimental materials” (the alleged “error” is our failure to represent *Project* as a relationship and our representing *Phase* as a relationship in our ontologically unclear conceptual model). Again, these alleged errors are the essence of the problem we are investigating (see below). Moreover, our questions either directly or indirectly engage the three composites we have represented as a relationship (*Team*, *Project Plan*, *Purchase Requisition*). For these reasons, we reject Allen and March’s argument that our results are fundamentally flawed.

**Confounds in the Experimental Treatment**

Allen and March argue two factors confound the experimental treatments we used in our experiment. The first is our “imbalanced use” of ternary relationships. The second is that our experimental task favors our ontologically clear diagram “in a manner distinct from the experimental treatment” (p. 953).
Allen and March (p. 953) contend “the semantics of ternary relationships are more difficult to understand than the semantics of binary relationships, especially for novices.” Thus, they claim that better performance by participants who received the ontologically clear treatment should be expected.

Our response is twofold. First, representing a composite as a relationship *dictated* that we had to use a ternary relationship in our ontologically unclear diagram. Even if it is true that ternary relationships are more difficult to understand than binary relationships, *this is the fundamental point that underlies our theory and our experiment*—namely, for a number of reasons, it is better to represent composites as entities and not relationships! Our use of ternaries is not a confound; it is the *essence* of the treatment.

Moreover, as a way to represent the composites in our domain, we could not have replaced the ternary relationships in our ontologically unclear model with a set of binary relationships. A set of binaries does not convey the idea that the composite classes in our domain comprise two component classes and, in addition, are related to another object class. Use of binary relationships would also increase the number of constructs in the ontologically unclear diagram (relative to the ontologically clear diagram) and thus introduce another source of complexity in the diagram.

Second, as we point out in our paper (pp. 561-562), prior research pertaining to the *comprehension* of ternary relationships has produced equivocal results. For this reason, at least for the moment, we are unwilling to assume ternary relationships will inevitably lead to poorer comprehension performance. Moreover, in the context of empirically evaluating conceptual modeling grammars, Burton-Jones et al. (2009) point out the dangers of not empirically testing predictions that appear obvious.

Allen and March (p. 953) also contend the comprehension questions we gave our participants “systematically favor the ontologically clear diagram in a manner that is independent from the intended experimental treatment.” In their Appendix B, essentially they argue our comprehension questions are more difficult to answer using the ontologically unclear diagram.

Their observation that *prima facie* many of the questions appear more difficult to answer using the ontologically unclear diagram is *exactly the point of our research*. Like Allen and March, we too believe the questions are more difficult to answer. In this regard, we have provided a *theoretically based explanation* to account for why the questions will be more difficult to answer using the ontologically unclear diagram. Nonetheless, given the preliminary nature of research in this area, we are unwilling to assume our “obvious” proposition holds until we obtain at least some empirical evidence to either support or contest it (see, further, Burton-Jones et al. 2009).

In our paper, we also explained that sometimes the “correct answer” differed between the ontologically clear diagram and ontologically unclear diagram because the diagrams varied in their ability to represent the domain semantics (pp. 564-565). For each diagram, we took care to try to ensure the answers to the questions were consistent with the domain semantics represented in the diagram. In other words, the correct answers to the questions for the ontologically unclear diagram were based only on the semantics represented in the diagram (and not the semantics represented in the ontologically clear diagram).

## Deviations from UML Definitions

In our paper, we indicated (p. 560) we used “standard” UML notation in our experimental materials. Nonetheless, we noted a deviation in the way we represented attributes in the diagrams. Allen and March argue our experimental materials contain “two additional notational deviations that may have had significant, unintended influence of the study’s results” (p. 953).

Allen and March’s first concern is our placement of the names of relationships *inside* rather than outside the diamond symbol used to represent a ternary relationship. They quote Rum-baugh et al. (2005, p. 470) in support of their argument that the names of ternary associations should be placed *outside* the diamond symbol.

The convention of placing the name of a ternary relationship inside the diamond symbol is widely accepted in the conceptual modeling field. For instance, the widely used textbook by Connolly and Begg (2010) (now in its fifth edition) states (p. 327) “the UML notation uses a diamond to represent relationships with degrees higher than binary. The name of the relationship is displayed *inside* the diamond” (our emphasis). Similarly, the widely used textbook by Hoffer et al. (2009) (now in its ninth edition) shows a UML ternary relationship in Figure 15-3(c) (p. 624) with the name of the relationship (*Supplies*) *inside* the diamond symbol.

Allen and March (p. 954) go on to argue, “This [our placing names inside the diamond symbols], in conjunction with using nouns to name the ternary relationships, encourages the
misreading of the relationships as if they were classes” (our emphasis). We are flummoxed by this argument, because earlier in their research note (p. 948) they defend Elmasri and Navathe’s use of noun as a label for a relationship:

Our interpretation of the diagram [in Elmasri and Navathe] is that the “committee” relationship indicates that there is an association between individual graduate students and individual faculty members by virtue of their mutual participation in a committee, even though the committee itself is not represented in the model. If Elmasri and Navathe intended to represent the committee object they would have done so using an entity.

We concur with Allen and March that using a noun as label for a relationship potentially leads to an interpretation of a conceptual model that differs from the interpretation intended by the creator of the model. Our use of nouns for the labels of the ternary relationships in Figure 6 of our paper, however, arises because we are using relationships to represent composites (things) and not mutual properties. In the context of the ontologically unclear diagram, therefore, our use of nouns for ternary relationship labels reinforces the idea that the ternary relationships are intended to represent things (composites). Indeed, it is difficult to think of appropriate verbs or verb phrases that could be used as labels for the ternary relationships when they are intended to represent composites.

More to the point, Allen and March cannot have their cake (defending Elmasri and Navathe’s use of a noun as a label for a relationship) and eat it too (criticizing our use of a noun as a label for a relationship). [SE Note #7: As can be inferred from SE Note #2 in reference to Elmasri and Navathe’s problematic Committee example, it is debatable whether Allen and March “defend” Elmasri and Navathe’s use of this label.] Either use of a noun as a label for a relationship suggests the existence of an underlying entity type or object class, or it does not. Our view is that nouns should not be used as labels for relationships because this practice will confuse many users of conceptual models. Potentially it will mislead them to misinterpret the real-world semantics being represented by the model.

Allen and March also argue we have departed from the “standard” UML interpretation of multiplicities (cardinalities) for ternary associations (relationships). Again, they reference Rumbaugh et al. (2005, pp. 470, 472) to support their allegation.

At the time we designed our experimental materials (2001–2002), we cannot recall whether a standard interpretation of the multiplicities that pertained to a UML ternary association existed. Moreover, our examples above of contrary practices in using UML indicate that at least for the moment the notion of standard UML is problematic. Many commonalities exist in the ways UML is used currently, but clearly differences still remain.

In any case, the important issue is that our participants received a set of materials that explained how our UML syntax was to be interpreted. In addition, they could reference these materials throughout the experiment. At the outset of the experiment we also explained carefully to our participants how our UML syntax was to be interpreted. We did not allow them to commence the experiment until we were confident they understood the meaning of the syntax. Contrary to Allen and March’s suggestions, none of our participants had the type of modeling experience that would have led to dissonance about the ways we positioned our multiplicities versus the so-called “standard” positioning of multiplicities in UML.

Analysis of Performance

For our ontologically correct model, Allen and March dispute the answer we provided to question 2 of our problem-solving questions. They then impugn our results by stating (p. 955):

Without more information about how the other nine questions were scored, it is not possible to determine if this is an isolated or a pervasive problem. However, with such problems in the scoring of participant answers and the aforementioned confounds, the statistical analysis of the experimental results involving subjects’ performance cannot be interpreted as supporting the hypotheses.

Question 2 of our problem-solving questions is the following: A team leader has resigned. Does the model allow the team to continue to work on the project without him?

Allen and March argue (p. 10) “the existence of an instance of a relationship requires exactly one instance of each related entity; hence, a team cannot exist without its leader.” Thus, they conclude the answer to our question 2 is “not possible.” We assert that Allen and March’s response is incorrect because they have not answered the question asked. Instead they have answered a different question—a question akin to the following: Can a team exist without a team leader?

10 Nonetheless, in light of Allen and March’s criticism, we have gone back and checked that the “accuracy” of the answers we used to score a participant’s response as “accurate” or “not accurate.” We believe our answers to all questions are correct.
If this were the question asked, their answer of “not possible” is correct. Our question asks about a specific individual, however, and not a team leader in general. If a specific team leader resigns and is replaced by another team leader, the team can continue to work on the project. Indeed, this logic was manifested in some of our participants’ responses.

The multiple possible interpretations of our question 2 serve to illustrate one of the motivations for our research—namely, even when the semantics of a conceptual model seem “obvious,” how users of the conceptual model interpret the semantics still needs to be investigated.

Furthermore, Allen and March’s different interpretation nicely illustrates the flaw in their approach to debunking our interpretation of Elmasri and Navathe’s and Teorey et al.’s conceptual models by showing that our meaning does not match the authors’ meaning. Again, the denotational semantics associated with a conceptual model do not always match the connotational semantics associated with the model (and the differences sometimes occur in surprising ways).

Allen and March’s different interpretation of our question 2 also shows why we used a multidimensional measure of problem-solving performance. While we scored a participant’s response as “accurate” if it was congruent with the semantics we ascribed to our conceptual models, we often found interpretations of the models that differed from our interpretations but nonetheless were clear and compelling. For this reason, we also scored participants’ responses on the basis of the clarity of their answers and the clarity of their interpretation of the conceptual models. In footnote 15 of our paper (Shanks et al. 2008, p. 565), we indicated our results held across all three measures of problem-solving performance (thereby providing evidence in support of the validity and reliability of our results). Allen and March, on the other hand, rely only on a single measure in evaluating their participants’ performance.

New Experimental Evidence

Allen and March (p. 955) argue our results could be explained “equally well” by our “asymmetric use” of ternary relationships in our experiments rather than the existence of construct overload. They present the results of two experiments they conducted to bolster their argument. The first experiment was intended to show that users of conceptual modeling diagrams are better able to understand the semantics of a domain when binary relationships rather than ternary relationships are used. The second experiment was intended to show “requisite methodological procedures to test the effects of construct overload while considering the effects of differences in conceptual modeling constructs” (p. 11).

Allen and March’s First Experiment

In their first experiment, Allen and March (pp. 955-956) found their participants performed worse in their comprehension of the semantics of a domain represented by a conceptual model that had ternary relationships rather than binary relationships. In this light, they argue our use of ternary relationships in only one of the conceptual models in our experiment “introduces a strong confound” in our study.

Again, we point out that we could not avoid using ternary relationships if we were to represent composites as relationships in the conceptual model of our experimental domain. Once a composite represented as a binary relationship has to be associated with another entity class, a ternary relationship must be used. In short, the use of ternary relationships to represent a composite is not a confounding in our experiment. It is the essence of the treatment used to operationalize the proposition we are testing.

Our experiment also differs in several ways from Allen and March’s first experiment:

- Participants in our experiment were practitioners working in industry who had little, if any, experience of conceptual modeling. Participants in Allen and March’s experiments were university students (p. 955) who “had received several months training on UML including binary and ternary associations.”
- We used conceptual models that are more representative of the size of conceptual models found in practice. Allen and March used model fragments in their experiment.
- We used a ternary relationship to represent composites, whereas Allen and March used a ternary relationship to represent an association between three object classes.
- We assessed participants’ performance using a scoring system based upon three dimensions (correctness of answer, explanation clarity, and interpretation clarity).
Allen and March used a scoring system based only on a single dimension (correctness of answer).

- We used primarily a problem-solving measure to evaluate our participants’ performance. Allen and March used primarily a comprehension measure to evaluate their participants’ performance.

In light of these differences, great care needs to be exercised when reaching any conclusions based on a comparison of the results obtained from our experiment versus their first experiment.

Perhaps more importantly, the way in which Allen and March required their participants to interpret the semantics of a ternary relationship differed from the way in which we required our participants to interpret a ternary relationship. In this light, we worked through the comprehension questions associated with Allen and March’s Figure 5(a). Our experience was that answering the questions using the semantics for ternaries that they required their participants to employ was more difficult than using the semantics for ternaries that we required our participants to employ.

**Allen and March’s Second Experiment**

As with their first experiment, Allen and March’s second experiment differs in several important ways from our experiment: their use of student participants well versed in UML rather than experienced practitioners who had little, if any, knowledge of UML; their use of a simpler conceptual model and domain; and their use of a simpler measure of performance. As a consequence, once more, great care needs to be exercised in comparing the results of their second experiment with our experiment.

More importantly, Allen and March have several flaws and confoundings in the materials they used in their second experiment. These undermine the validity and reliability of their results and their claim (p. 955) that their results “show that construct overload does not have a significant effect on subjects’ performance.”

Appendix A contains an analysis of the defects that exist in the materials Allen and March used in their second experiment. In brief, however, the following problems (among others) exist:

- Figure 7(a), which Allen and March claim implements construct overload on the association symbol, in fact implements construct overload only if they rely on logic that contradicts logic they used earlier in their research note to defend Elmasri and Navathe’s problematic Committee example. [SE Note #8: As stated earlier in reference to Elmasri and Navathe’s problematic Committee example, the suggestion that Allen and March “defend” Elmasri and Navathe is debatable.]

- Figures 7(b) and 9(a), which Allen and March claim have no construct overload, in fact have instances of construct overload in their use of the attribute symbol to represent both intrinsic attributes in general and mutual attributes in general.

- Figure 9(b), which Allen and March claim implements construct overload on the labeled arc symbol, in fact fails in its implementation because the labels chosen for the arcs clearly differentiate compositions from simple associations. Allen and March also fail to recognize that construct overload exists elsewhere in their use of the attribute symbol to represent both intrinsic attributes in general and mutual attributes in general.

Allen and March have undertaken four sets of statistical analysis on the data they collected in their second experiment. In our view, for several reasons the conclusions they reach based upon their analyses are problematical.

First, they compare performance outcomes for Figure 7(a), which they describe as “Association Class Overloaded,” with Figure 7(b), which they describe as “Not Overloaded.” They indicate (p. 959) “all significant performance differences favored the treatment with no overload and no association classes (Figure 7b over Figure 7a),” which suggests that “construct overload is detrimental to human performance.”

As we point out in Appendix A, Figure 7(b) has an overloaded attribute construct, which should mitigate against finding performance differences between Figures 7(a) and 7(b). Perhaps, therefore, the association-class overload undermines performance more severely than the attribute overload. Moreover, as we point out in Appendix A, Figure 7(b) is a simpler diagram because it contains a smaller number of instances of grammatical constructs. The labels on the arcs in Figure 7(b) also clearly signal the difference between a simple association and a composite. Given these confoundings, it is difficult to know the source of the performance differences between Figures 7(a) and 7(b).

Second, Allen and March compare performance outcomes for Figure 8(a), which they describe as “Not Overloaded,” with Figure 9(b), which they describe as “Labeled Arc Overloaded.” They indicate (p. 958) “all significant performance
differences favored the treatment with overload and no association classes (Figure 9b over Figure 9a),” which suggests that “construct overload is beneficial to human performance.”

[SE Note #9: Relative to their two experiments, Allen and March say (page 958):

\textit{If the two experimental forms are evaluated independently, then effect of construct overload is confounded by the degree to which subjects understood association classes…That is, if the first experimental form were used, and the effects of association classes were ignored, then the analysis would suggest that construct overload is detrimental to human performance. However, if the second experimental form were used, and the effects of association classes were ignored, then the analysis would suggest that construct overload is beneficial to human performance. Only by combining the experimental forms can the analysis separate the effects of construct overload from the effects of association classes.}]

As we point out in Appendix A, the labeled arcs in Figure 9(b) do \textbf{not} introduce construct overload. Instead, overload exists in Figure 9(b) because the attribute construct has been used to represent both intrinsic attributes in general and mutual attributes in general (but this form of overload also exists in Figure 9(a)). Moreover, as we point out in Appendix A, contrary to Allen and March, we argue that construct overload does in fact exist in Figure 9(a) because their experimental participants would have \textit{interpreted} the use of the association class construct to be a form of construct overload. In a nutshell, our view is that Figure 9(b) is less overloaded than Figure 9(a). Thus, contrary to Allen and March’s conclusions, their results \textit{support} the proposition that the existence of construct overload undermines human performance.

Third, Allen and March undertake a “between-subjects” analysis by comparing performance outcomes with Figure 7(a), which they describe as “Association Class Overloaded,” with performance outcomes with Figure 9(a), which they describe as “Not Overloaded.” They also undertake a “between-subject” analysis by comparing performance outcomes with Figure 7(b), which they describe as “Not Overloaded,” and performance outcomes with Figure 9(b), which they describe as “Labeled Arc Overloaded.” Because they detect no statistically significant performance differences in their comparisons, they conclude that “construct overload is \textit{not} a salient predictor of human performance” (p. 961, our emphasis).

We conclude otherwise. As we have indicated above and in Appendix A, \textit{all of Allen and March’s experimental materials manifest construct overload in some form}. Based on their results, the best they can conclude is that the different forms of construct overload present in their four treatments on average did not lead to differential performance among their experimental participants. This conclusion must also be tempered by the fact that Allen and March used conceptual models that lacked the semantic complexity of conceptual models that most likely would be used in practice.

**Conclusions**

Referencing Popper (1963), Allen and March argue (p. 962) they have “falsified” the theory of ontological clarity and that therefore it “must either be discarded or modified to account for new experimental findings.” The problematic nature of Popper’s notion of “falsification” is well known, however, among philosophers of science (see, e.g., Godfrey-Smith 2003, pp. 63-67). For instance, the validity and reliability of any experiment are difficult to assess, and thus it is difficult to determine unequivocally what the results of any experiment mean for a theory. Popper’s notion of falsification also does not deal well with theories that make \textit{probabilistic} predictions—in other words, predictions that an outcome is likely or unlikely to occur (which is often the case for many social science theories).

Moreover, given the flaws we see in Allen and March’s experimental evidence, we reject their claim about the theory of ontological clarity being falsified. Nonetheless, we agree with them about “the difficulty of formulating research questions and appropriately developing and executing experimental procedures” (p. 962) in the conceptual modeling domain (see, further, Burton-Jones et al. 2009). We also agree with them that most likely the original theory of ontological clarity will have to be “modified to account for new experimental findings” (p. 962). For instance, new evidence might lead to the boundaries of the theory being identified and articulated. We welcome such refinements to the theory.

Allen and March also contend (p. 962) “we see no compelling evidence to indicate that Bunge’s ontology holds significant promise as an underlying foundation for conceptual modeling.” Citation evidence indicates, however, that many other scholars within the conceptual modeling field do see value to Bunge’s ontology.\textsuperscript{12} Moreover, Allen and March argue (p. 962),

\textsuperscript{12}Google Scholar™ provides an easy source of citation evidence.
With no construct to represent conceptual objects or events we contend that Bunge’s ontology suffers significant construct deficit with respect to the representation of business domains.

This comment is misplaced. As we pointed out earlier in this response to Allen and March, Bunge (pp. 116-119) deals specifically with the notion of conceptual objects in his ontology. Furthermore, he deals at length with the notion of events (and processes) in his ontology (pp. 221-226).

[SE Note #10: But see the SE Appendix, Selections From Bunge, quotes from pages 218-219 where Bunge states that in his ontology, events, and processes refer only to concrete things, not to constructs.]

Allen and March’s critique of our paper is one of several they have written criticizing the work that one of us (Weber) has done with Wand in relation to using Bunge’s ontological theory to provide a formal foundation for some types of conceptual modeling phenomena (e.g., Allen and March 2006a, 2006b; March and Allen 2010). In spite of attempts to resolve various matters of contention, little progress has been made. In a Kuhnian sense, we believe the problem is that a number of us in the conceptual modeling field are operating within different “paradigms” (Kuhn 1970).

Science can be conceived a marketplace for ideas. As producers of ideas, scientists offer their products (their ideas) in this marketplace. Other scientists are the consumers in this marketplace. They evaluate the products and choose whether to “consume” them. Moreover, in due course, even products (ideas) that are consumed (used) may be discarded as better products (ideas) emerge. In this regard, science is littered with examples of theories that for a time have provided valuable insights but that eventually have been replaced by better theories. This outcome does not mean these former theories are worthless; indeed, they might have provided valuable insights that underpinned the development of the new theories that replaced them. In this light, at least for the moment, we would prefer to leave it to other colleagues in the conceptual modeling field to make judgements about the merits of Bunge’s ontology and the theory of ontological clarity.

Acknowledgments

An Australian Research Council Discovery Grant funded the research that is the subject matter of this response to the critique of our work by Allen and March. Also, three of our coauthors on our original MIS Quarterly paper have now left academe. They have read our response to Allen and March and support our arguments, but we take responsibility for the content of our response.

References


About the Authors

Graeme Shanks is Professor of Information Systems at the University of Melbourne. He received his Ph.D. in Information Systems from Monash University. Prior to becoming an academic, Graeme worked for a number of years as programmer, programmer-analyst, and project leader in several large organizations. His research interests focus on conceptual modeling, data quality, decision support systems, identity management and the implementation and impact of enterprise and interorganizational systems. He is a member of the editorial boards of six journals and was recently a member of the Australian Research Council College of Experts.

Ron Weber is Dean, Faculty of Information Technology, Monash University. He graduated with a first-class honours degree and University Medal in accounting from the University of Queensland and an MBA and Ph.D. in management information systems from the University of Minnesota. His research interests lie in the areas of the ontological foundations of information systems, management of the information systems function, and computer control and audit. He is a past president of the Association for Information Systems and a past Editor-in-Chief of the MIS Quarterly.

Appendix A

Analysis of Flaws/Problematic Aspects of Conceptual Models Used in Allen and March’s Second Experiment

Figure 7(a). Association Class Overloaded

In Figure 7(a), Allen and March argue (p. 956) that overload exists because “association class notation is used to represent a simple association (i.e., Works For and Contracts With) as well as a composite (i.e., Team and Assignment).” It seems they expect participants in their experiment to interpret the Team and Assignment association classes as entity/object classes—in other words, classes of things rather than mutual properties in general (see Rules 1 and 5, Evermann and Wand 2006, p. 24). Earlier in their paper, however, Allen and March criticized our interpretation of one of Elmasri and Navathe’s relationships as a composite. Recall, Elmasri and Navathe give the label “Committee” to a relationship between two entities labeled “Faculty” and “Grad_Student.” As a consequence, we argued that Elmasri and Navathe’s conceptual model has construct overload because they have used a relationship construct to represent both a mutual attribute in general and a class of things (composites). Moreover, we argued that their use of noun label for the relationship (Committee) is likely to further predispose a user of Elmasri and Navathe’s diagram to interpret the relationship as representing a composite. In defence of Elmasri and Navathe, however, Allen and March (p. 948) argue: “The label ‘committee’ on the relationship simply stands in place of another, potentially more descriptive label such as ‘advises/is advised by,’ representing the role played by each entity in the relationship. This interpretation is more consistent with the way in which graduate committees are typically conceptualized.”

Ironically, Allen and March have used our logic to justify their claim for construct overload in Figure 7(a). If instead we use their logic, our argument would be that users of Figure 7(a) would make a “more consistent” interpretation of Team and Assignment as association classes and not entity/object classes. For instance, a plausible interpretation is (1) Client Employee “works in a team with” Auditor, and (2) Team “is assigned” a Task. Either they subscribe to our arguments, in which case Figure 7(a) indeed has overloaded association classes, or they subscribe to their own arguments, in which case the association classes in Figure 7(a) are not overloaded.
**Figure 7(b). Not Overloaded**

Allen and March claim (p. 957) the conceptual model shown in Figure 7(b) of their paper contains no construct overload. They are incorrect. In Figure 7(a), they show an association class called *Contracts With* that has four attributes: *Contract Begin Date*, *Contract End Date*, *Contract Price*, and *Contract Requirements*. Because these attributes pertain to an association class, based on Bunge’s (1977) ontology they constitute *mutual attributes in general* between the classes joined by the association class—namely, Account Rep and Client (see Rule 3, Appendix 1 of Evermann and Wand 2006, p. 24). In Figure 7(b), however, these attributes are embedded within the Client Class, along with the *five intrinsic attributes in general* of the Client Class—namely, Company Name; Address; City; State; and Phone. As a result, contrary to their claim, Allen and March’s conceptual model in Figure 7(b) has construct overload because they have used a single UML construct (an attribute) to represent two ontological constructs: (1) an intrinsic attribute in general, and (2) a mutual attribute in general.

Moreover, representing *Contract Begin Date*, *Contract End Date*, *Contract Price*, and *Contract Requirements* as *mutual* attributes in Figure 7(a) implies that a *history* of contracts between clients and account representatives is to be kept (via multiple instances of the *Contracts With* association class). Representing these attributes as *intrinsic* attributes of the *Client* object class in Figure 7(b), however, implies that only the *current* values of these attributes are to be kept for a particular client. In other words, the histories of contracts between account representatives and clients are semantics that are not of interest to domain stakeholders.

[SE Note #11: The arguments on both sides of this particular issue are fairly technical, perhaps beyond the reach of the typical experimental subject. Allen and March counter Shanks and Weber’s argument above by pointing out that the cardinality on the contracts-with association in Figure 7a indicates that a client can only engage in the association once. Therefore at any time a client has one and only one Account Rep. Accordingly, they suggest that the possibility of a “contract history” is not admitted by the diagram.]

In short, the meaning of *Contract Begin Date*, *Contract End Date*, *Contract Price*, and *Contract Requirements* is different, depending on whether they are represented as mutual attributes or intrinsic attributes. Given that Allen and March used a within-subjects experimental design, potentially their participants were confused about the meaning to be ascribed to these attributes because of the conflicting representation in Figures 7(a) and 7(b) and thus the semantics of the domain being represented via the diagrams.

Figure 7(b) has fewer instances of grammatical constructs (seven classes and six associations) than Figure 7(a) (nine classes and eight associations). If Figures 7(a) and 7(b) had had the same number of classes and the same number of associations, the superior performance that Allen and March ascribe to participants who used Figure 7(b) might not have been evident in their experimental results.

Compared to Figure 7(a), Figure 7(b) not only has no instances of the association class construct, it also has labels on its arcs. These labels clearly signal the difference between a simple association and a composite. If labeled arcs had been used in Figure 7(a), the superior performance that Allen and March ascribe to participants who used Figure 7(b) might not have been evident in their experimental results. In short, it is not clear whether any of the performance differences that arise between users of Figure 7(a) and users of Figure 7(b) can be attributed to non-use of association classes in Figure 7(b) and/or non-use of labeled arcs in Figure 7(a).

**Figure 9(a). Not Overloaded**

Allen and March claim (p. 959) the conceptual model shown in Figure 9(a) of their paper contains no construct overload. They are incorrect. As in Figure 7(b), they have construct overload because they have used a single UML construct (an attribute) to represent two ontological constructs: (1) an intrinsic attribute in general, and (2) a mutual attribute in general (see the analysis above for Figure 7(b) for the attributes in the Client and Client Employee classes).

Allen and March imply that Figure 9(a) is not overloaded because the association class construct has been used only to represent composites in the diagram. Given their participants had been trained extensively in UML, however, they still might have been confused by the diagram. Presumably, participants would have expected that the association class construct would have been used to represent associations that had their own attributes and aggregation (or composition) constructs would have been used to represent composites. In short, construct overload still would have existed if participants perceived that the association class construct was being used to represent real-world phenomena that usually were represented by another UML grammatical construct.
Figure 9(b). Labeled Arc Overloaded

Allen and March claim (p. 959) the conceptual model shown in Figure 9(b) of their paper has construct overload because “labeled arc notation is used to represent simple association (i.e., Works For and Contracts With) as well as to represent composition (i.e., Team and Assignment).” In our view, however, labels such as “Contracts With” and “Works for” suggest semantics that are clearly different from the semantics suggested by a label such as “Is Part Of.” As a result, readers of Figure 9(b) will see that the arcs used in the figure represent two different types of semantics (association and composition).

Nonetheless, Figure 9(b) has construct overload, but not in the form that Allen and March have stated. Rather, as in Figures 7(b) and 9(a), they have construct overload because they have used a single UML construct (an attribute) to represent two ontological constructs: (1) an intrinsic attribute in general, and (2) a mutual attribute in general (see the analysis above for Figure 7(b) for the attributes in the Client and Client Employee classes).