

A RESEARCH NOTE ON REPRESENTING PART-WHOLE RELATIONS IN CONCEPTUAL MODELING

Gove N. Allen

Marriot School of Management, Brigham Young University, 783 Tanner Building,
Provo, UT 84602 U.S.A. {gove@byu.edu}

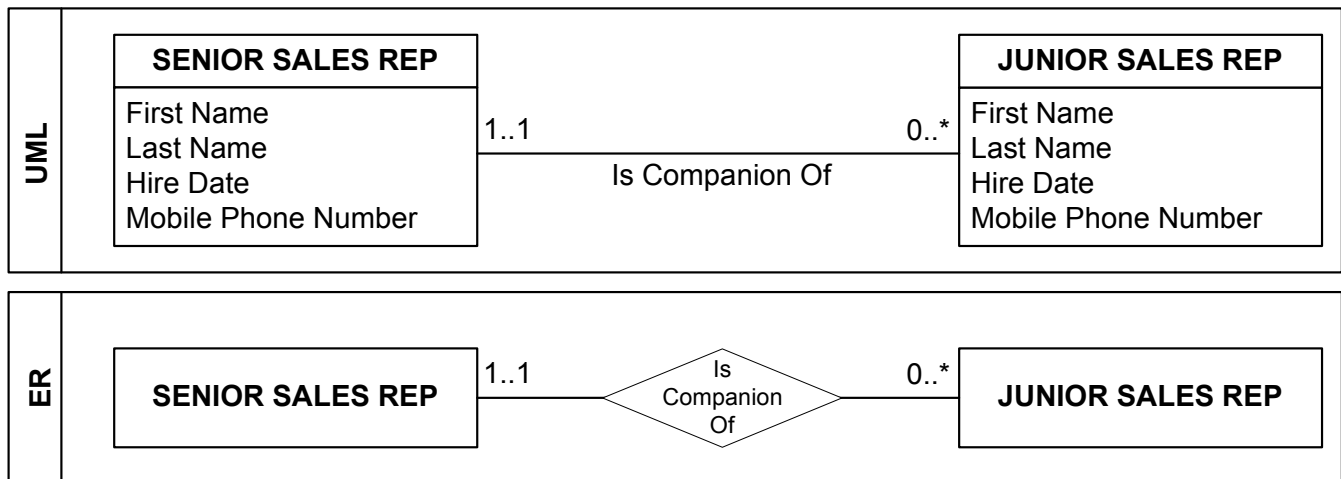
Salvatore T. March

Owen Graduate School of Management, Vanderbilt University,
Nashville, TN 37203 U.S.A. {sal.march@owen.vanderbilt.edu}

Appendix A

A Review of Relevant UML and ER Notation

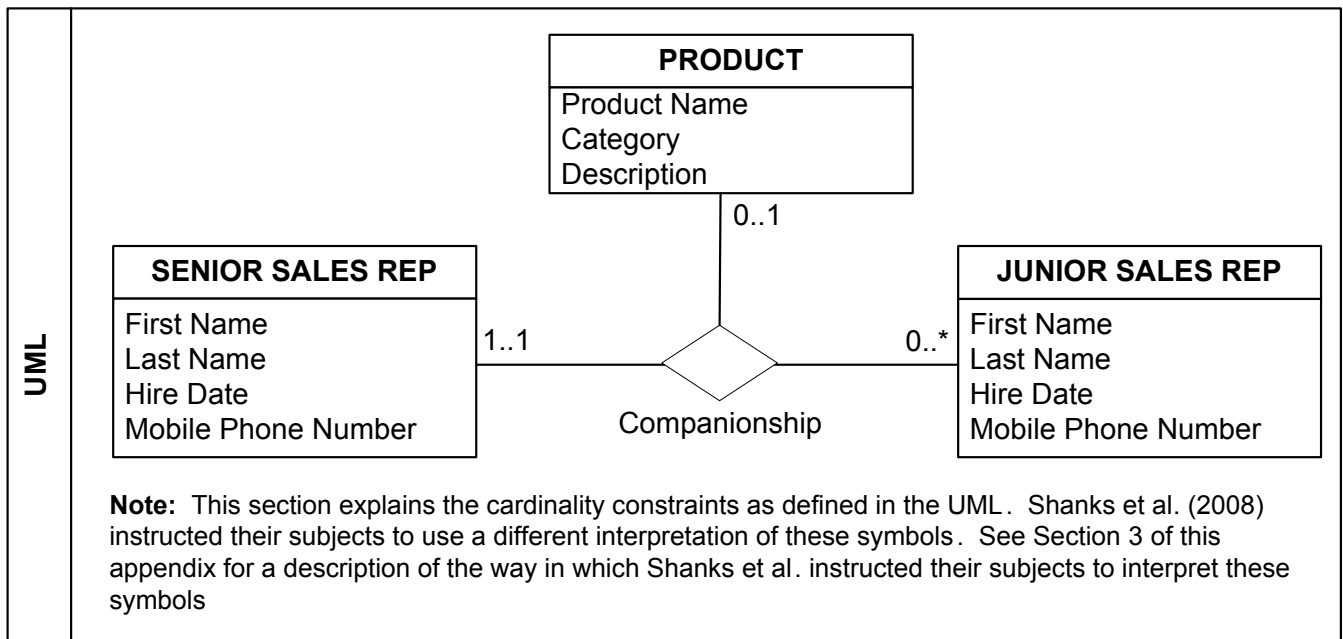
Section 1: Semantic Details of ER and UML Binary Association Relationships



A specific SENIOR SALES REP may or may not be the companion of any JUNIOR SALES REP. Each SENIOR SALES REP can be the companion of many JUNIOR SALES REPs. Each JUNIOR SALES REP must be associated with exactly one SENIOR SALES REP. An instance of the “Is Companion Of” relationship is a specific combination of a single SENIOR SALES REP and a single JUNIOR SALES REP. There can only be one instance of the relationship for a given combination of SENIOR SALES REP and JUNIOR SALES REP.

Sample Question	Answer
Is it possible for a SENIOR SALES REP to not be the companion of a JUNIOR SALES REP?	Yes. The minimum cardinality constraint near JUNIOR SALES REP is zero (0). This means that a SENIOR SALES REP need not participate in the association.
Can a JUNIOR SALES REP be the companion of more than one SENIOR SALES REP simultaneously?	No. The maximum cardinality constraint near the SENIOR SALES REP specifies that a JUNIOR SALES REP can be associated with at most one SENIOR SALES REP.
What is the name of the relationship between SENIOR SALES REP and JUNIOR SALES REP?	"Is Companion Of"

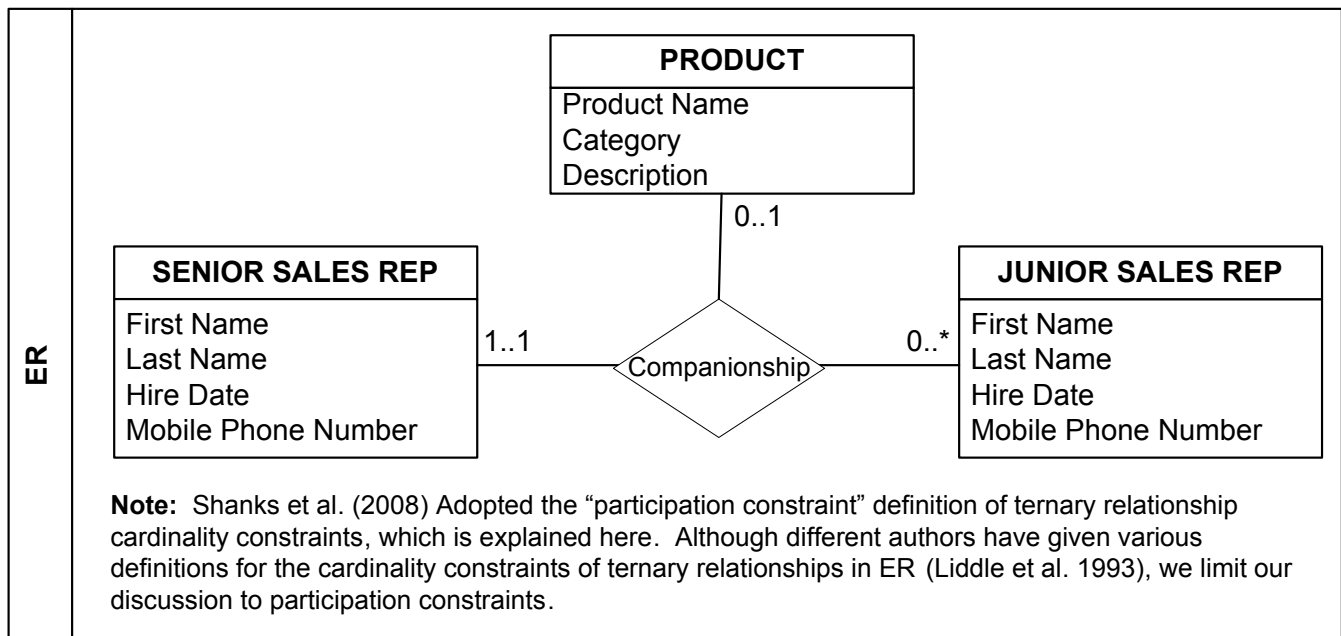
Section 2: Semantic Details of UML Ternary Association Relationships



A Companionship is a single instance of the ternary relationship; it associates exactly one SENIOR SALES REP, exactly one PRODUCT and exactly one JUNIOR SALES REP. The cardinality constraint near SENIOR SALES REP (minimum = 1; maximum = 1) indicates that every possible combination of JUNIOR SALES REP and PRODUCT must be associated with exactly one SENIOR SALES REP. The cardinality constraint near PRODUCT (minimum = 0; maximum = 1) indicates that although every possible combination of JUNIOR SALES REP and SENIOR SALES REP need not be associated with a product (i.e., there is no instance of the relationship that associates that combination of JUNIOR SALES REP and SENIOR SALES REP) each combination can be associated with at most one PRODUCT. The cardinality constraint near JUNIOR SALES REP (minimum = 0; maximum = *) indicates that although every possible combination of PRODUCT and SENIOR SALES REP need not be associated with a JUNIOR SALES REP, it is possible for a specific combination to be associated with more than one JUNIOR SALES REP (Rumbaugh et al. 2005, p. 471).

Question	Answer
What is a Companionship?	An instance of the Companionship association; a unique triplet of one SENIOR SALES REP, one JUNIOR SALES REP, and PRODUCT.
Can a Companionship have more than one PRODUCT?	No. A Companionship is the combination of exactly one each of SENIOR SALES REP, JUNIOR SALES REP, and PRODUCT.
Can a specific combination of SENIOR SALES REP and JUNIOR SALES REP be associated with more than one PRODUCT?	No. The cardinality constraint near PRODUCT indicates that a given combination of SENIOR SALES REP and JUNIOR SALES REP can have at most one PRODUCT.
Is it possible for a SENIOR SALES REP to participate in no Companionship?	Yes. Because neither all combinations of PRODUCTS and SENIOR SALES REP require a JUNIOR SALES REP nor all combinations of JUNIOR SALES REP and SENIOR SALES REP require a PRODUCT, it is possible to have a SENIOR SALES REP that does not participate in a Companionship.
Is it possible for a JUNIOR SALES REP to participate in no Companionship?	No. The cardinality constraint near SENIOR SALES REP requires every possible combination of PRODUCT and JUNIOR SALES REP to be associated with a SENIOR SALES REP.

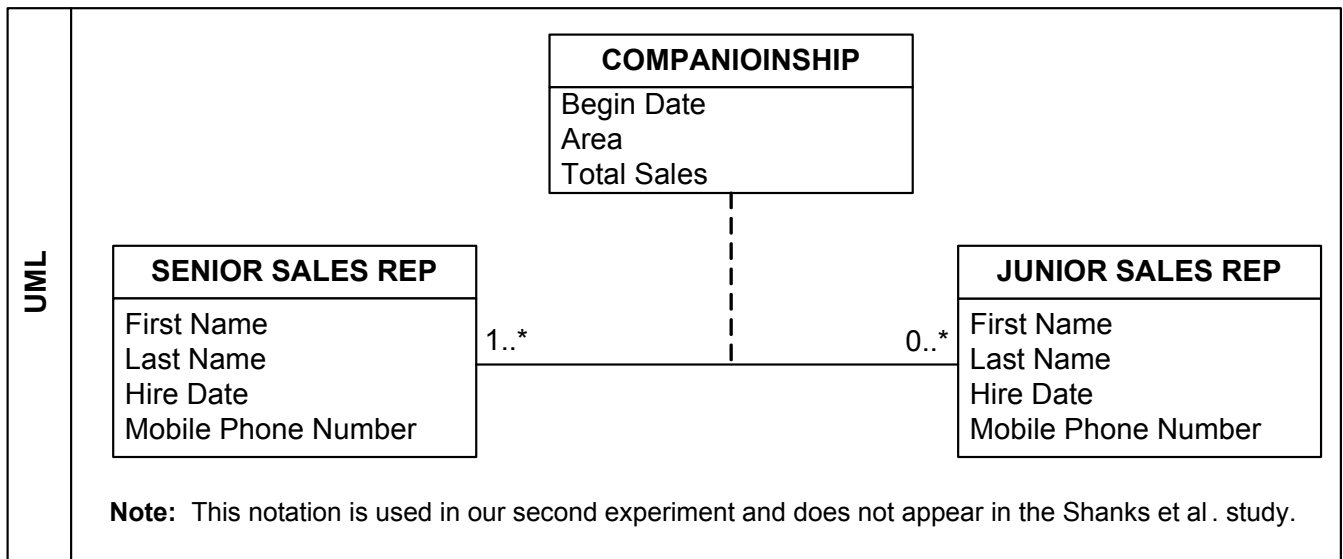
Section 3: Semantic Details of ER Ternary Association Relationships



A Companionship is a single instance of the ternary relationship; it associates exactly one SENIOR SALES REP, exactly one PRODUCT and exactly one JUNIOR SALES REP. The cardinality constraint near SENIOR SALES REP (min = 1; max = 1) indicates that each SENIOR SALES REP must appear in the Companionship relationship exactly one time. The cardinality constraint near PRODUCT (minimum = 0; maximum = 1) indicates that each PRODUCT need not participate in a Companionship but cannot participate in more than one. The cardinality constraint near JUNIOR SALES REP (minimum = 0; maximum = *) indicates that a JUNIOR SALES REP need not participate in a Companionship but could appear in many.

Question	Answer
What is Companionship?	An instance of the Companionship association. It is a specific combination of a SENIOR SALES REP and a JUNIOR SALES REP. This is the nature of an association class.
Can a Companionship have more than one PRODUCT?	No. A Companionship is the combination of exactly one each of SENIOR SALES REP, JUNIOR SALES REP, and PRODUCT.
Can a specific combination of SENIOR SALES REP and JUNIOR SALES REP be associated with more than one PRODUCT?	No. Each SENIOR SALES REP can only participate in a single Companionship.
Is it possible for a SENIOR SALES REP to participate in no Companionship?	No. Each SENIOR SALES REP must participate in exactly one Companionship.
Is it possible for a JUNIOR SALES REP to participate in no Companionship?	Yes. The minimum cardinality of "0" near JUNIOR SALES REP indicates that each need not appear in any Companionship.

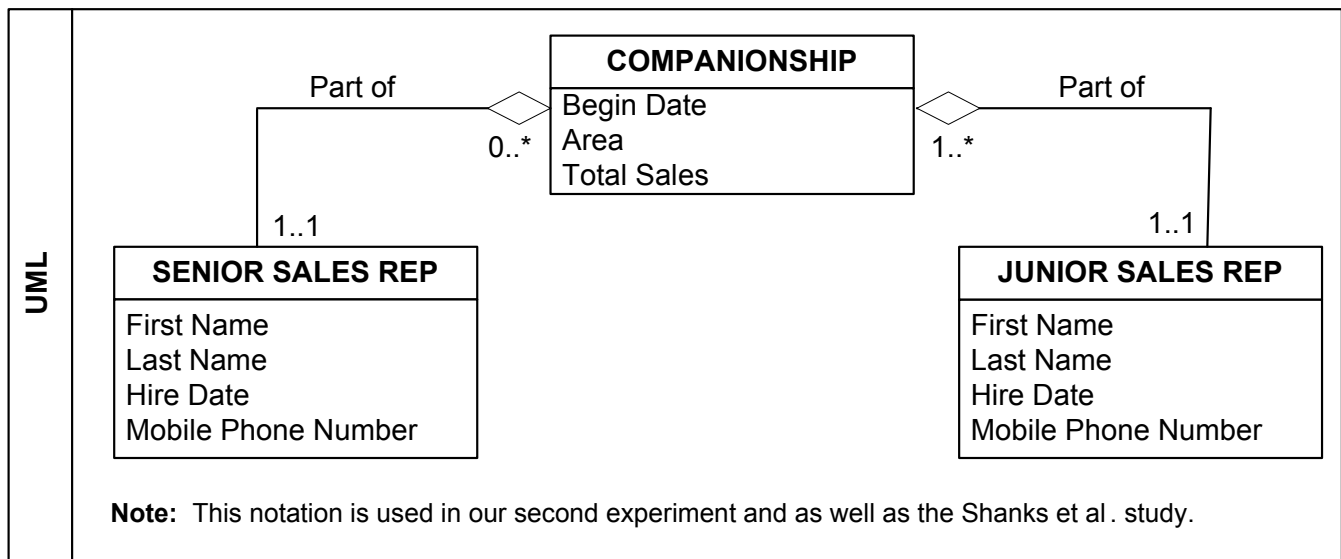
Section 4: Semantic Details of UML Association Classes



A specific SENIOR SALES REP may or may not be associated with any JUNIOR SALES REP. Each SENIOR SALES REP can be associated with many JUNIOR SALES REPs. Each JUNIOR SALES REP must be associated with at least one SENIOR SALES REP but could be associated with more. A specific combination of a single SENIOR SALES REP and a single JUNIOR SALES REP is called a COMPANIONSHIP. There can only be one COMPANIONSHIP for a given combination of SENIOR SALES REP and JUNIOR SALES REP.

Question	Answer
What is a COMPANIONSHIP?	An instance of the COMPANIONSHIP association. It is a specific combination of SENIOR SALES REP and JUNIOR SALES REP. It is also an instance of the COMPANIONSHIP class. This is the nature of an association class.
Is it possible for a SENIOR SALES REP to not participate in a COMPANIONSHIP?	Yes. The minimum multiplicity near JUNIOR SALES REP is zero (0). This means that a SENIOR SALES REP need not participate in the association.
Can a JUNIOR SALES REP participate in two COMPANIONSHIPS simultaneously?	Yes. The maximum multiplicity near the SENIOR SALES REP specifies that a JUNIOR SALES REP can be associated with more than one SENIOR SALES REP. There is no constraint that prevents the two associations from existing at the same time.
Can a specific combination of a SENIOR SALES REP and a JUNIOR SALES REP have two Areas?	No. A specific combination of SENIOR SALES REP and JUNIOR SALES REP is a COMPANIONSHIP. Each COMPANIONSHIP has a single Area. Because COMPANIONSHIP is an association class, it is identified by its components—there can only be one COMPANIONSHIP for a given combination of SENIOR SALES REP and JUNIOR SALES REP.
What is the name of the association between SENIOR SALES REP and JUNIOR SALES REP?	COMPANIONSHIP. The name of the association class is the name of the association.
What is "Area"?	An attribute of the COMPANIONSHIP association or an attribute of the COMPANIONSHIP class. Both statements are true.

Section 5: Semantic Details of UML Aggregation



A COMPANIONSHIP is composed of exactly one JUNIOR SALES REP and one SENIOR SALES REP. A specific SENIOR SALES REP need not be a component of any COMPANIONSHIP. Each SENIOR SALES REP can be a component of many COMPANIONSHIPS. Each JUNIOR SALES REP must be a component of at least one COMPANIONSHIP but could be component of more.

Question	Answer
What is a COMPANIONSHIP?	An instance of the COMPANIONSHIP class. It is a thing that is composed of one SENIOR SALES REP and one JUNIOR SALES REP. The small diamonds that touch the COMPANIONSHIP class indicate that it is composed of members of the associated classes. The cardinality constraints near the other two classes (1..1) indicate that a COMPANIONSHIP must have exactly one of each.
Is it possible for a SENIOR SALES REP to not be a part of a COMPANIONSHIP?	Yes. The minimum cardinality constraint for SENIOR SALES REP near COMPANIONSHIP is zero (0). This means that a SENIOR SALES REP need not be a part of any COMPANIONSHIP.
Can a JUNIOR SALES REP participate in two COMPANIONSHIPS simultaneously?	Yes. The maximum multiplicity near COMPANIONSHIP for JUNIOR SALES REP specifies that a JUNIOR SALES REP can be a part of more than one COMPANIONSHIP.
Can a specific combination of a SENIOR SALES REP and a JUNIOR SALES REP have two Areas?	Yes. A COMPANIONSHIP is composed of exactly one SENIOR SALES REP and one JUNIOR SALES REP. Although each COMPANIONSHIP has only a single Area, no constraint prevents two different COMPANIONSHIPS from being composed of the same combination of SENIOR SALES REP and JUNIOR SALES REP.
What is the name of the association between SENIOR SALES REP and JUNIOR SALES REP?	No such association exists. SENIOR SALES REP and JUNIOR SALES REP are each individually associated with COMPANIONSHIP. In each case, the name of the association is "Part of."
What is "Area"?	An attribute of the COMPANIONSHIP class.

Appendix B

Confounds Related to the Problem-Solving Questions Used in the Shanks et al. Study

Figure B1 lists the questions in the experimental task used by Shanks et al. (2008). Several are considerably more difficult to answer in the ontologically unclear diagram than in the ontologically clear diagram. Others make statements that are incompatible with the semantics expressed in ontologically unclear diagram, potentially causing confusion among subject in that treatment.

Consider, for example, question 10 (see Figure B1). The question asks if it is possible to find the most current project plan for a project. According to Shanks et al., the answer to this question for both treatments is "possible." For the ontologically clear treatment (Figure 3), a subject need only look at the Project Plan class and see that it has an attribute named "Project Plan Version #" to understand that project plans have versions and that this attribute holds the information necessary to provide the answer.¹ However, for the ontologically unclear treatment (Figure 4), the process is not nearly so simple.

To arrive at an answer of "possible" for this question subjects in the ontologically unclear treatment would need to understand that an instance of the Project Plan relationship is the association of one project instance, one budget instance, and one scope instance. Budget has an attribute named "Budget Version #" and Scope has an attribute named "Scope Version #." A subject must reason that if a given Project Plan instance associates a specific Project instance, with a Budget instance that has the highest value of "Budget Version #" and a Scope instance that has the highest value of "Scope Version #" for that Project instance then the project plan is the current one. Further, the subject must also reason that for any given project, the budget with the highest "Budget Version #" will appear in the same project plan as the scope with the highest "Scope Version #." This requires a careful examination of the participation constraints of the Project Plan relationship. The "1" near Budget

¹It is unclear why the Shanks et al. argue that subjects using the ontologically clear diagram would "check the 'budget #' and 'scope #' attributes to determine whether they are those associated with the latest version of the project plan" (footnote 13, p. 563) rather than check the "project plan version #" attribute.

For each of the following questions, participants were asked to choose from the following three responses: possible; not possible; not sure. They were then asked to briefly explain their answer.

1. Project X is made up of ten phases. Does the model allow more than one team leader to work on some of the phases?
2. A team leader has resigned. Does the model allow the team to continue to work on the project without him?
3. A client has requested that a project will start in five months time. A department wants to create a team with a leader now, and give the leader the next five months to select appropriate team members as she pleases. Is this possible?
4. Project Z has overlapping phases, which occur simultaneously. Is it possible for a single team member to work on two concurrent phases?
5. Project Y consists of two phases. Phase 1 has been completed without consumables. Phase 2 now requires the purchase of consumables such as floppy disks, zip drives, and architects' pencils. Does the model allow the team to purchase these necessary consumables?
6. The client of project D decides on a weekly basis what work will be required for the project. The project has no key deliverables but will require consumables. Can the project keep track of the consumables it purchases?
7. The client for project X wishes to make purchase requisitions for its next three phases, as there are rumors that the consumables they require will be experiencing a shortage. The supplier has not yet been decided. Does the model allow the purchase requisitions to be made?
8. An employee wishes to check their skill appropriateness for an upcoming project by checking the key deliverables and scope of the project. At this point, the project has been divided up into phases, but contains no project plan. Using the diagram, can the employee check their suitability?
9. Client Z has used the same supplier for the past 5 years. The supplier charges at a discount rate and bills at the end of each phase. Does the model allow the purchase of consumables if there are no key deliverables?
10. An employee becomes a member on a project that has been running for 3 years. He has been given a project plan, but its budget and scope do not seem to match the size and completeness of the project. Can he check that he has the current plan?
11. A client has had to significantly reduce the budget of a project with five phases remaining. The reduced budget is not enough to cover all these phases. Can the team leader prioritize the remaining phases?

Figure B1. Experimental Questions Used by Shanks et al. (Appendix A in Shanks et al. 2008, p. 573)

indicates that each budget is a part of one and only one project plan. The “1” near Scope indicates that each scope is a part of one and only one project plan. Because each project plan must have a scope and a budget, for a new Project Plan instance to exist, it must be associated with a new Budget instance and a new Scope instance. Accordingly, the newest instance of the Budget class will always be associated with the newest instance of the Scope class in the most recently created project plan. Given this reasoning, a subject can determine that it is possible to tell if a given project plan is the most current.

Clearly, arriving at the answer using the ontologically unclear diagram is more difficult than arriving at the answer using the ontologically clear diagram. It is important to note that the added difficulty arises not because of any differences in “ontological clarity” but because the information needed to answer the question resides not in a single attribute of a single class but in two different attributes of two different classes and requires reasoning about participation constraints and instance creation.

Similarly, question 11 asks if phases can be prioritized. This question is easy to answer affirmatively (possible) in the ontologically clear diagram because “Priority Level” is an attribute in the Phase class. No such attribute exists in the ontologically unclear diagram. However, before a subject using this diagram can determine that the answer is “not possible,” she must conduct a search of other entities and attributes and assess if there might be some other way to establish the priority of phases, a more involved and complex process than finding the single attribute “Priority Level.”

Finally, several questions include statements that are incompatible with the ontologically unclear diagram, while none include statements that are incompatible with the ontologically clear diagram. For example, questions 1 and 5 assert that projects are composed of phases. This assertion is explicitly represented in the ontologically clear diagram, but is inconsistent with the ontologically unclear diagram. If the ternary

relationships in the ontologically unclear diagram are intended to implicitly represent composites then the interpretation is not that a project is composed of phases, but rather that a phase is composed of a project, a key deliverable and a consumable.

Question 5 further states that a phase “has been completed without consumables.” This is possible in the ontologically clear diagram but not in the ontologically unclear diagram (a phase instance in the ontologically unclear diagram is the association of exactly one project, one key deliverable, and one consumable; all are required for a phase to exist). Similarly, question 3 indicates that teams can have more than one member. Again, this is possible in the ontologically clear diagram but not in the ontologically unclear diagram (a team instance in the ontologically unclear diagram is the association of exactly one project, one leader and one member). Subjects faced with such contradictory statements in the experimental materials would undoubtedly be forced to reevaluate their understandings of the ontologically unclear diagram as they tried to resolve these conflicts, potentially making the task more difficult and potentially reducing their confidence in their knowledge about the meaning of the diagram.

Because these questions systematically favor the ontologically clear treatment in ways that are independent of “ontological clarity,” they confound the analysis of subject performance. It is not possible to conclude that the source of any observed performance differences is ontological clarity. Performance cannot be interpreted as supporting the study’s propositions.

References

- Liddle, S., Embley, D., and Woodfield, S. 1993. “Cardinality Constraints in Semantic Data Models,” *Data and Knowledge Engineering*, (11), pp. 235-270.
- Rumbaugh, J., Jacobson, I., and Booch, G. 2005. *The Unified Modeling Language Reference Manual* (2nd ed.), Boston: Addison-Wesley.
- Shanks, G., Tansley, E., Nuredini, J., Tobin, D., and Weber, R. 2008. “Representing Part–Whole Relations in Conceptual Modeling: An Empirical Evaluation,” *MIS Quarterly* (32:3), pp. 553-573.