1-1-2008

Associations and Mutual Properties - An Experimental Assessment

Joerg Evermann
Memorial University of Newfoundland, jevermann@mun.ca

Haidir Hallimi
Victoria University of Wellington

Associations and Mutual Properties – An Experimental Assessment

Joerg Evermann
Memorial University of Newfoundland
jevermann@mun.ca

Haidir Halimi
Victoria University of Wellington

ABSTRACT

Associations are a widely used construct of object-oriented languages. However, the meaning of associations for conceptual modelling of application domains remains unclear. Ontological considerations in past research suggest that associations are related to the concept of mutual properties. Specifically, previous research has suggested that mutual properties, not associations, should be modelled, and guidelines for doing this in UML have been offered. This paper presents the results of an experimental study, which suggest that this guidance does in fact lead to improved models.

INTRODUCTION

Object-oriented modelling languages are increasingly being used for describing business and organizational application domains (conceptual modelling). In order to have well-defined meaning, their constructs must be defined in terms of the elements of the application domain they are intended to describe (Harel and Rumpe, 2004). The use of constructs without clearly defined meaning can lead to ambiguous or confusing models.

The meaning of the association construct, central to object-oriented modelling, remains unclear, as the following attempts at a definition show:

- An association is "the simplest form of a relationship" (IBM, 1997, pg. 195)
- "An association represents the relationships between objects and classes" (Bahrami, 1999, p.26)
- "Relationships associate one object with another" (Embley, 1992, pg. 18)
- "If two classes have an association between them, then instances of these classes are, or might be, linked." (IBM, 1997, pg. 195).
- "An association sets up a connection. The connection is some type of fact that we want to note in our model" (Siegfried, 1995, pg. 105)

The original definition by Rumbaugh sheds little light on the issue:

- "A relation associates objects from *n* classes. ... A relation is an abstraction stating that objects from certain classes are associated in some way." (Rumbaugh, 1987, pg. 466)

This lack of clarity remains even in the latest UML standard:

- "An association defines a semantic relationship between classifiers." (OMG, 2005, pg. 36)

To clarify the meaning and semantics of the association construct, previous research has used upper-level ontologies, primarily that of Mario Bunge (Bunge, 1977, 1979) as descriptions of the relevant semantic domain (Evermann and Wand, 2001b, 2001a, 2005b; Green and Rosemann, 2000; Opdahl and Henderson-Sellers, 2002; Wand *et al.*, 1999; Wand and Weber, 1993). These ontologies are argued to describe what exists (or is perceived to exist) in the application domain and as such provide the set of relevant concepts that should be used to define the semantics of language constructs.

Identifying the meaning of associations for conceptual modelling should be based on an examination of their actual or intended purpose or use. Associations in conceptual modelling have two distinct purposes (Evermann, 2005a). Their first purpose is to indicate and enable message passing. UML for example requires that classes be connected by associations so that their instance objects can exchange messages. Booch (1991) terms this a *usage relationship*, Coad and Yourdon (1990) call them *message connections*.

A second purpose of associations is to represent properties that are relevant or meaningful to a modeller, as indicated by the above quotes from prominent researchers, and the following: "It is important that relations be considered a semantic construct" (Rumbaugh, 1987, pg. 467), "associations define the way objects of various types can be linked or connected - enabling the construction of conceptual networks" (Martin and Odell, 1992, pg. 259), a "class relationship might indicate some kind of semantic connection" (Booch, 1991, pg. 96).

Based on the first identified use of associations as message connections, Evermann (2005a) has argued that there is no corresponding ontological construct, as the prevalent ontology in conceptual modelling (Bunge, 1977, 1979) offers communication mechanisms that differ considerably from message passing. Based on the second identified use of associations, as semantic connections, Evermann (2005a) has argued such properties are relevant to a modeller and therefore subjective, i.e. not part of the domain proper. In conclusion, Evermann (2005a) proposes that associations have no counterpart in the domain and hence should not be used. Instead, modellers should directly represent mutual properties in their conceptual models of application domains.

Implicit in the mutual property proposal (Evermann, 2005a) is the promise of a better model. Existing literature on ontological evaluation of conceptual modelling languages suggests that languages that conform to Bunge's ontology (Bunge, 1977, 1979) will yield improvements in models created with them. These improvements have been found primarily with respect to increased domain understanding in the model interpreter (Gemino and Wand, 2003, 2004), but also with respect to time and ease of interpretation (Aranda *et al.*, 2007; Topi and Ramesh, 2002). The cognitive reason for the observed improvement is the idea of ontological clarity (Gemino and Wand, 2003, 2004, 2005), which suggests that an ontologically-informed model is a better fit with existing cognitive structures (Evermann, 2005b; Aranda *et al.*, 2007; Scaife and Rogers, 1996) and therefore more effectively and efficiently integrated with existing knowledge to enable expanded reasoning capabilities.

While Evermann (2005a) makes the implicit promise of improved models based on sound theoretical reasoning, only anecdotal support by means of a case study is offered. Hence, this research aims to offer a more rigorous examination of the following research question:

> *Does the use of mutual properties instead of associations in conceptual modelling, as proposed in (Evermann, 2005a), lead to increased domain understanding by the model interpreter?*

The remainder of the paper proceeds as follows. The next section briefly reviews the guidelines for modelling with mutual properties given in (Evermann, 2005a). The following section then introduces the experimental design for the present study, followed by a description of the models that serve as experimental stimuli and the instrument for the dependent variable. Data analysis is then presented, and the findings are interpreted and discussed. The paper concludes with a brief summary and opportunities for future research.

MUTUAL PROPERTIES

In Bunge's ontology (Bunge, 1977, 1979) the world consists of things, which have properties. Mutual properties, also called shared properties, are properties of two or more things. Hence, any change of a mutual property in one thing is a change in all the things that share the property. When a series of changes in a thing A involves changes to a mutual property P_m, this may start a series of changes in the thing or things that share this property, e.g. thing B. Thus, thing A has *acted on* thing B through property P_m. Hence, interaction in this ontology is described in terms of mutual properties. Conversely, mutual properties serve to "carry" interaction.

Using mutual properties in conceptual models requires a language construct (textual or graphical symbol) to represent them. The proposal in (Evermann, 2005a) suggests the use of the association class attribute symbol for this, stressing that the association class symbol itself is uninterpreted and carries no meaning. Only the notation is borrowed, with the guideline that association class attribute symbols are only to be used to represent mutual properties as described here. Hence, representation of interaction by means of mutual properties and what is traditionally called static structure, models involving objects and attributes, go hand-in-hand in this proposal.

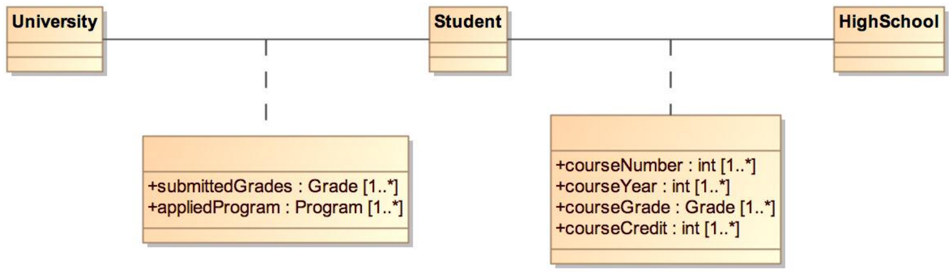

Figure 1: Example model using mutual properties, from (Evermann, 2005a)

An example from the case study in (Evermann, 2005a) is shown in Figure 1. Notice that the associations themselves are not labelled, as they have no meaning. The attributes of the association classes enable and represent possible interactions between the connected things. For example, a student can attend a high school. The interaction of attending is represented as changes in any or all of the attributes `CourseNumber`, `CourseYear`, `CourseGrade` or `CourseCredits`. Note that these are shared; it is meaningless to speak of a course or a student without also considering the school or institution this course is taken at. Similarly, a student may apply to a university by changing the attributes `submittedGrades` or `appliedProgram`.

EXPERIMENTAL DESIGN

To assess the implicit claim of better models that is made in (Evermann, 2005a), an experiment is conducted. We examine model interpretation of object-oriented diagrams by novice modellers. We use novice modellers as the effect sizes, relative to their (expected) low overall performance, are expected to be larger, whereas any effect on experienced modellers might be relatively small compared to their high overall performance and therefore more difficult to detect. While differences between novices and experts have been shown in object-oriented programming (Davies *et al.*, 1995; McKeithen *et al.*, 1981; Pennington *et al.*, 1995) where novices use more syntactical than semantic classification, and experts use more functional than object classification, no such empirical results are available for object-oriented modelling. However, as object-oriented programming and modelling are based on the same abstraction principles, we argue that with the increased focus on syntax by novices, any syntactic effects, as examined in this study, might be more pronounced for novices than for experts (any aid will help novices more than experts). Thus, we chose internal validity over external validity, an issue we take up again in our discussion section.

We compare two UML class models, one that is constructed using the mutual properties proposal (Evermann, 2005a) and another that uses associations. The focus of observation (the dependent variable) is the domain understanding, operationalized as problem solving capability, following Gemino and Wand (2003, 2005) and Gemino (1999). We measure and control for both modelling and domain expertise of subjects. Tables 1 and 2 show how this study is positioned in the frameworks of Aranda *et al.* (2007) and Gemino and Wand (2004) for the experimental evaluation of modelling languages.

Type of Task	Problem solving ("functional task")
Language expertise	Novice modellers
Domain expertise	Measured as self-reported
Problem size	Simple

Table 1: Affecting variables from (Aranda et al., 2007)

Content	Restaurant domain
Grammar/construct	UML, associations and association classes
Nature of comparison	Intra-grammar (different ways of using UML)

Use of grammar	Use of mutual properties as proposed in (Evermann, 2005a)
Medium of content delivery	Graphics
User characteristics	Novice UML modellers
Task	Interpretation of model
Focus of observation	Product of task: Domain understanding
Criterion for comparison	Effectiveness: Problem solving questions

Table 2: Variables and dimensions from (Gemino and Wand, 2004)

In terms of the framework by Topi and Ramesh (2002), our research primarily addresses the effect of data modelling formalism on user performance, rather than user attitudes. We control for modelling skill and domain knowledge, which are identified as moderators of this effect (Topi and Ramesh, 2002). However, we specifically exclude other human factors, such as training (although any effect of training is mediated by modelling skills according to Topi and Ramesh (2002)), cognitive style, intellectual ability or work experience. We also exclude task characteristics, such as complexity and size, from the study. While these have been shown to be relevant (Bajaj, 2004), we have no reason to believe that the mutual properties proposal in (Evermann, 2005a) is limited to specific task characteristics.

This study also relies on the theoretical guidance offered by Gemino and Wand (2004). They propose informational and computational equivalence of the chosen material (stimuli) to ensure that the results reflect only the effect of the presentation options. Informational equivalence is defined as "all information in one is also *inferable* from the other and vice versa". The emphasis on "inferable" makes it clear that this goes further than a depiction of the information in the material, but also encompasses the use of additional, existing knowledge by the model interpreter to *infer* information not explicitly provided. We follow the critique by Burton-Jones *et al.* (2007) who suggest this is impossible to achieve in practice. Moreover, following the argument by Evermann and Wand (2006) and counter to the proposal by Parsons and Cole (2005), we claim that this definition of informational equivalence is unsuitable for our purpose. If not from additional cognitive processing ("inference") that is enabled by one representation and not the other, where should an observable difference between the two representation options arise from? Gemino and Wand (2004) suggest that computational equivalence builds on informational equivalence and not only seeks to eliminate differences in effectiveness, as done by informational equivalence, but also differences in the efficiency of the representations. Again, it is precisely because we are interested in the differences in effectiveness and efficiency of the two representation options that we do not subscribe to the principles of informational and computational equivalency as proposed by Gemino and Wand (2004) and Parsons and Cole (2005) in this study. We do however ensure that the information that is *explicitly depicted* in the stimuli is equivalent, i.e. we do not *show* more information in one condition than in the other.

Any differences in effectiveness and efficiency are, in line with Gemino and Wand (2004), expected to arise out of ontological clarity of the modelling grammar. The use of mutual properties, which are directly based on ontology, rather than associations, which have no ontological interpretation, is expected to yield cognitive difference in the efficiency and effectiveness with which the diagram can be integrated with existing knowledge. These cognitive differences in effectiveness and efficiency are expected to cause differences in the ability to reason with the newly interpreted model information. These differences are in turn observable as differences in problem solving ability in the domain (Gemino and Wand, 2003, 2005; Gemino, 1999).

MODELS – EXPERIMENTAL STIMULI

The domain description that is chosen for this method had been used as a teaching case, although not within the sample frame that we drew our subjects from. The case describes a restaurant kitchen that prepares delivery orders as well as in-restaurant orders. As a teaching case, a substantial textual description of the domain was available, from which a UML model was constructed. The mutual-properties model was developed by the second author, working from the description and reasoning provided in (Evermann, 2005a) as well as related guidelines in (Evermann and Wand, 2001b, 2001a, 2005b, 2005a). The model was developed iteratively bye the second author and validated with the first author until both agreed that it complied with the mutual properties proposal in (Evermann, 2005a). The model underwent six iterations until it was deemed compliant. An excerpt of the final model is shown in Fig. 2.

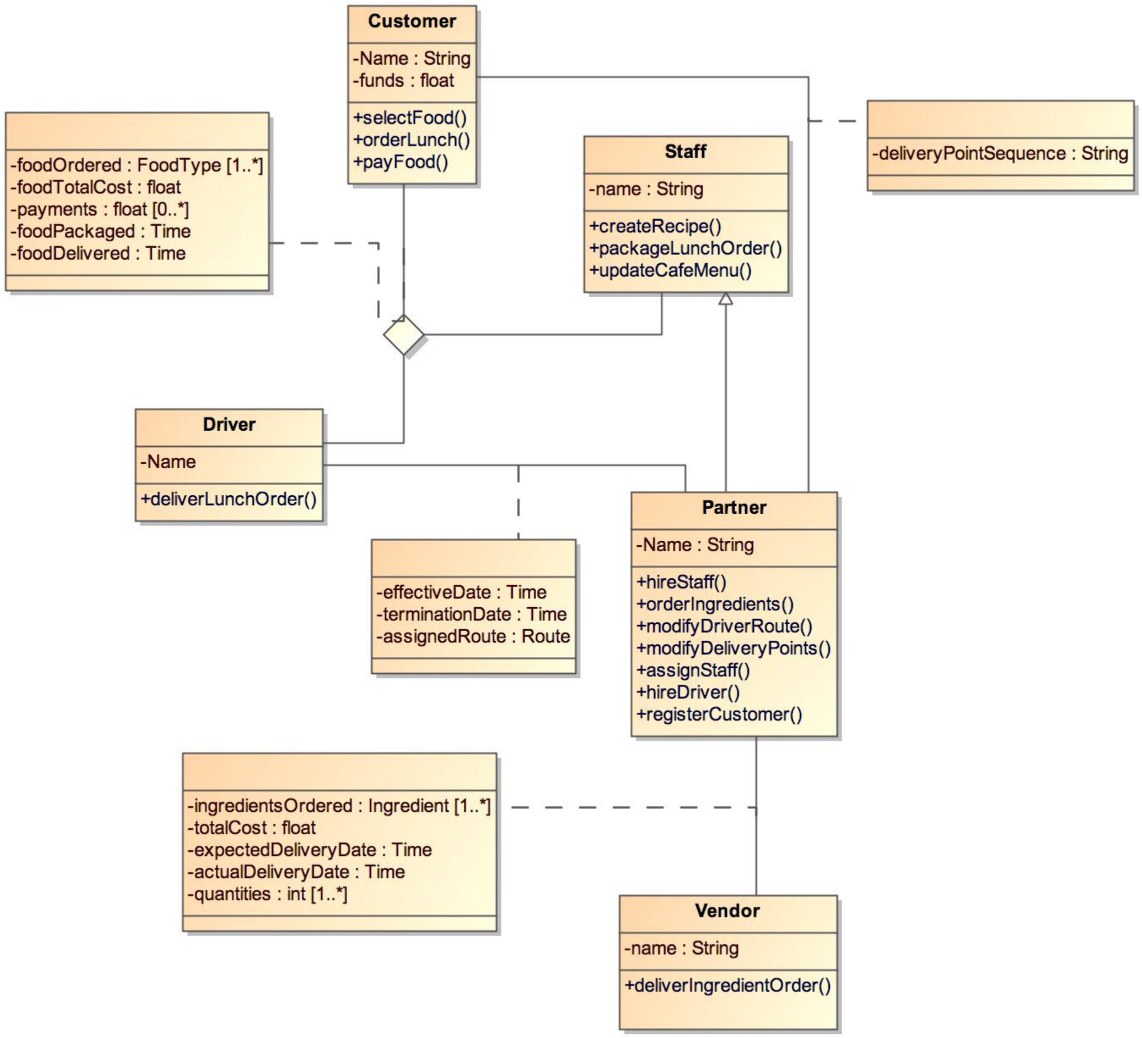

Figure 2: Excerpt from model using mutual properties

To maintain the amount of information *shown* in the model, the association model was derived from the mutual properties model. Specifically, mutual properties were converted to either associations or attributes of classes. For each bundle of mutual properties, represented by a set of association class attributes, four post-graduate students in software engineering with modelling experience in UML were asked to read the case description and then assign these properties to either an association or a class. No serious discrepancies arose and Fig. 3 shows an excerpt of the final model.

DEPENDENT AND CONTROL VARIABLES

Dependent and control variables were measured by means of a questionnaire. The dependent variable, domain understanding, was operationalized as problem solving capability, following established examples (Gemino, 1999; Gemino and Wand, 2005; Bodart *et al.*, 2001; Evermann and Wand, 2006; Burton-Jones and Meso, 2002). The set of questions for subjects was adapted primarily from (Evermann and Wand, 2006) and covers four sections. Model comprehension questions can be answered using information provided and require no inferences. Comprehension questions force subjects to engage with all aspects of the diagram. Problem solving questions are patterned after examples in (Gemino, 1999; Bodart *et al.*, 2001; Evermann and Wand, 2006) and cannot be answered without making inferences. Thus, performance on these questions is a result of domain understanding *beyond mere recall*. This is an accepted operationalization of domain understanding based on educational theory (Mayer, 1987, 1989). A third section of the questionnaire measures UML knowledge and is taken from (Evermann and Wand, 2006). The final section of the questionnaire contains items to measure subjects' self-reported domain knowledge. These were patterned after questions used in (Evermann and Wand, 2006). The questionnaire was pilot-tested with advanced undergraduate students in software engineering, who had completed three modelling courses, to ensure that questions are easy to understand and unambiguous. No problems were reported. The experiment also collected timing data for subjects on both the comprehension questions and the problem solving questions.

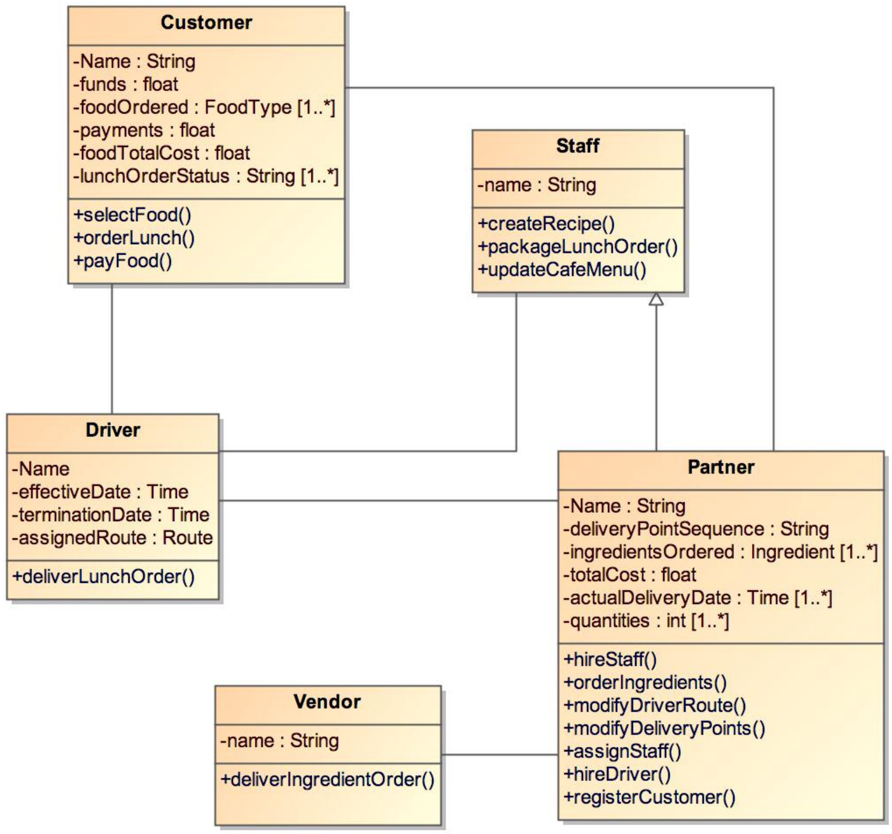

Figure 3: Excerpt from model using associations

DATA ANALYSIS

Data was collected from 29 students that were randomly sampled and randomly assigned to either the mutual-properties model or the association model. Students were run through the experiment individually and outside of class hours. The sample frame was the class list of three courses on UML development and software engineering, so that sufficient UML knowledge was guaranteed. All subjects were enrolled in either a Bachelor of Information Technology or a Computer Science degree. The degrees have similar course requirements leading to modelling and software engineering courses. Since these courses did not cover the association class notation of our mutual properties condition, a one-page introduction was provided to subjects in the mutual-properties condition (recall that the associations model did not use association classes).

Problem solving capability was rated as the total number of correct answers for all six problem solving questions. To ensure an unbiased assessment of correctness, we adapted the methods proposed by Gemino and Wand (2005) and Burton-Jones and Weber (1999) and used an answer guide and multiple coders. An initial set of correct answers was prepared by the second author. This set was then given to two domain experts who were familiar with the case study. Both experts had extensive knowledge in the restaurant and hospitality industry, as well as some knowledge of information modelling. The final answer guide was the union of the answers identified by the two experts.

The second author subsequently coded subjects' responses using the answer guide. A second coder, a graduate student in information systems, independently coded the responses. The interrater agreement (Cohen's κ) was .922 for the mutual properties model and .920 for the association model, which is considered excellent agreement (Nyerges *et al.*, 1998). Table 3 shows the problem solving performance and other descriptive data for the two models.

Model	N	Variable	Unit of measurement	Min	Max	Mean	SD
Mutual properties	15	Comprehension	# of correct answers	19	26	23.60	1.844
		Problem-solving		5	16	11.67	3.155
		UML Knowledge		13	19	15.67	1.633
		Domain Knowledge	7-pt Likert scale	1	4	2.33	.900
		Problem Solving Time	Minutes	6	17	10.93	3.453
		Comprehension Time	Minutes	15	35	24.33	6.61
Associations	14	Comprehension	# of correct answers	21	24	23.00	.961
		Problem-solving		0	15	8.71	3.931
		UML Knowledge		13	18	15.07	1.269
		Domain Knowledge	7-pt Likert scale	1	5	2.43	1.222
		Problem Solving Time	Minutes	5	20	11.5	3.995
		Comprehension Time	Minutes	12	43	29.79	9.593

Table 3: Descriptive Statistics

Analysis of data was done by means of ANCOVA. Normality of response data is required for the F-test and was tested for using the Shapiro-Wilks test. Separate tests for problem solving normality in both experimental conditions showed no significant departures from normality for the data (S-W is insignificant at .366 and .663). Graphical QQ-Normal plots also showed no significant deviation from normality. Levene's test of homogeneity of variance across the two experimental groups is not significant at .550, indicating no reason to reject homogeneity of variance.

To analyze the data, we ran two ANCOVA analyses. We found that due to the relatively small sample size, the full model including interaction effects shows a marginally significant effect (Table 4). However, the marginal significance is in this case the result of a relatively small sample size and the inclusion of a relatively large number of irrelevant independent variables in the model. The number of degrees of freedom of the residuals drops faster than the sum of squares of the residuals. Thus, contrary to the usual case of adding explanatory value by reducing the unexplained error, in this case the inclusion of variables actually lowers the power of the ANOVA F-test. Hence, when excluding the non-significant factors from the model for reasons of parsimony, the effect of the different models is clearly significant (Table 5).

```
                             Df  Sum Sq Mean Sq F value  Pr(>F)
Model                         1  63.120  63.120  4.3204 0.05223 *
UMLKnowledge                  1   0.045   0.045  0.0031 0.95636
DomainKnowledge               1   5.145   5.145  0.3522 0.56026
Time                          1  14.440  14.440  0.9884 0.33331
Model:UMLKnowledge            1   4.651   4.651  0.3183 0.57958
Model:DomainKnowledge         1   0.952   0.952  0.0652 0.80140
Model:Time                    1   2.977   2.977  0.2038 0.65710
UMLKnowledge:DomainKnowledge  1  24.930  24.930  1.7064 0.20790
UMLKnowledge:Time             1  17.862  17.862  1.2227 0.28340
DomainKnowledge:Time          1   6.216   6.216  0.4255 0.52245
Residuals                    18 262.972  14.610
```

Table 4: Full model ANOVA

```
                             Df Sum Sq Mean Sq F value  Pr(>F)
Model                         1  63.12   63.12  5.0097 0.03365 **
Residuals                    27 340.19   12.60
```

Table 5: Parsimonious model ANOVA

The ANCOVA results in Tables 4 and 5 show a significant effect of only the experimental condition ("model") on the problem-solving scores. Other controlled factors, UML knowledge, domain knowledge, and time taken for problem solving did not have a significant effect, nor did their two-factor interaction effects. The main effects plot of the data is shown in Fig. 4 and shows clearly the difference in problem solving performance.

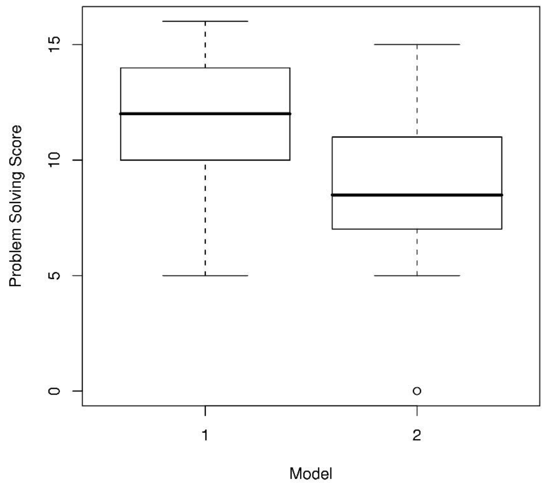

Figure 4: Effects of model on problem solving performance ("1" = mutual properties, "2" = associations)

DISCUSSION

While the study was clearly able to establish significant differences between the performances of subjects in the two experimental conditions (means of 11.57 compared to 8.71 total correct answers for six questions), the study was at the limit of its power, as evidenced by the ANCOVA issues described in the previous section. While the effect was not large (a 33% improvement in problem solving performance), this order of magnitude is in line with the effects observed in other studies that compared ontologically motivated modelling languages based on a problem solving operationalization of domain understanding. For example, Evermann and Wand (2006) report a 26% increase in problem solving capabilities for a set of ontological modelling rules, Burton-Jones and Weber (1999) report a 5% increase for ontologically clear languages, while Gemino and Wand (2005) report a 23% and 11% increase in performance for ontologically motivated mandatory properties. As these prior studies are also based on Bunge’s ontology (Bunge, 1977, 1979), the present study adds to the empirical support of this ontology.

Our proposition was the opposite of that advanced by Burton-Jones and Weber (1999). There, the authors suggested by different reasoning that association classes are ontologically unclear and would lead to a disadvantage in diagram interpretation performance. Both the association class as well as its attributes were interpreted as ontological properties leading to overload. Instead, the proposal in (Evermann, 2005a) and tested here removes this overload by suggesting that association classes have no ontological interpretation, but should only be used as a notational aid.

Results by Burton-Jones and Weber (1999) suggest that association classes only lead to diminished interpretation performance when domain knowledge is lacking. This effect is not present when domain knowledge is high, although a non-significant effect in the expected direction was found. Burton-Jones and Weber (1999) conclude that their study was possibly lacking in power, without however presenting a power estimate.

The results in the present study appear to be in line with this prior study. Here we have found a marginally significant result. Given that the power level in the present study was reasonably high (approx. 65%), one explanation, based on evidence in (Burton-Jones and Weber, 1999), is a medium to high domain familiarity by our subjects, which should reduce the effect size and therefore requires higher power in the study. While we have no comparison group, our ratings of self-reported ratings of domain knowledge are about 2.4 of 5, indicating moderate domain knowledge. On the other hand, the effect in the present study was in the opposite direction of the one found by Burton-Jones and Weber (1999).

Another study (Burton-Jones and Weber, 2003) also examined the interpretation of association classes. However, in their study, diagram interpretation was tested using comprehension, rather than problem solving ability. In contrast to Burton-Jones and Weber (2003), a post-hoc test for our study did not find a significant difference in comprehension between the two groups (p=.2868), nor did we make a prediction about any such effect. This non-significant result is also in line with a summary of related research that suggests we should not expect any differences in comprehension scores (Gemino and Wand, 2004). However, a different argument is presented in (Parsons and Cole, 2005) where the authors suggest that comprehension is a precursor to problem solving capability and we must therefore strive to understand and explain comprehension first. We see both types of operationalization of understanding, comprehension and problem solving, as being useful in their own right.

Our result is in line with the generally positive findings of the case study that is reported by Evermann (2005a). That case study was conducted using a development project with a team of experienced modellers and software engineers, while the present study uses relative novices. This may indicate that positive results from mutual properties models can be achieved both by novices as well as experts, although more detailed studies are needed to quantify the effects on experts. As we have argued above, we would expect the effects to be smaller, relative to the already high performance of model experts.

While the results of our study are statistically and practically significant, our study was limited in a number of ways and generalizations of our results to other situations and contexts must be made with caution. Specifically, our study examined model interpretation in a single domain by novice modellers. While this study was intended to exhibit high internal validity to determine whether any effect might be found, the limited external validity will need to be addressed by future research. More contextual factors need to be added (Gemino and Wand, 2005; Topi and Ramesh, 2002) to determine the extent and boundaries of the situations in which beneficial effects can be realized.

CONCLUSION

In summary, our results indicate that ontological clarity in a modelling language does improve the interpretation of models. Our research started from the premise that our direct representation of mutual properties would better convey the concept of mutual properties to model interpreters. Our results add to prior case study research (Evermann and Wand, 2006) in a multi-method research on ontological modelling. This enhances the validity and reliability of both studies.

As graphical representation plays a crucial part in understanding (Larkin and Simon, 1987), the graphical representation of mutual properties may have been a factor in this research. Alternative graphical representation may contribute to even better interpretive performance by model audience.

Modelling practices and modelling languages based on ontological foundations have attracted significant attention in the recent research literature. Our research adds to and confirms this already substantial body of knowledge. However, in contrast to many studies that use ontology as a tool to evaluate existing languages (Green *et al.*, 2005; Green and Rosemann, 2001, 2000; Opdahl *et al.*, 1999; Opdahl and Henderson-Sellers, 2002; Rosemann *et al.*, 2006) the present research goes one step further. Rather than presenting modelling rules as a guide to adapt the semantics of a given language (Evermann and Wand, 2005b), we propose to directly use ontological concepts for modelling. This would by definition be an ontologically clear way of modelling (Wand and Weber, 1993). The present proposal is but one small step towards this goal. We have focused on showing that the direct use of ontological concepts is feasible and beneficial. However, to extend this proposal to the full ontology, a new notation with the appropriate tool support will need to be provided. Future work in this direction is currently being planned.

REFERENCES

1. Aranda, J., Erns, N., Horkoff, J., and Easterbrook, S. (2007). A framework for empirical evaluation of model comprehensibility. In *Proceedings of the International Workshop on Modeling in Software Engineering, Washington, DC*.
2. Bahrami, A. (1999). *Object oriented systems development*. Irwin/McGraw-Hill, Boston, MA.
3. Bajaj, A. (2004). The effect of the number of concepts on the readability of schemas: An empirical study with data models. *Requirements Engineering*, **9**, 261–270.
4. Bodart, F., Patel, A., Sim, M., and Weber, R. (2001). Should optional properties be used in conceptual modelling? A theory and three empirical tests. *Information Systems Research*, **12**(4), 384–405.
5. Booch, G. (1991). *Object oriented design with applications*. Benjamin/Cummings, Redwood City, CA.
6. Bunge, M. A. (1977). *Ontology I: The Furniture of the World*, volume 3 of *Treatise On Basic Philosophy*. D. Reidel Publishing Company, Dordrecht, Holland.

7. Bunge, M. A. (1979). *Ontology II: A World of Systems*, volume 4 of *Treatise On Basic Philosophy*. D. Reidel Publishing Company, Dordrecht, Holland.
8. Burton-Jones, A. and Meso, P. (2002). How good are these UML diagrams? an empirical test of the Wand and Weber good decomposition model. In L. Applegate, R. Galliers, and J. I. DeGross, editors, *Proceedings of the Internataionl Conference on Information systems, Barcelona, Dec. 2002*.
9. Burton-Jones, A. and Weber, R. (1999). Understanding relationships with attributes in entity-relationship diagrams. In *Proceedings of the 20th International Conference on Information Systems (ICIS), Charlotte, NC*, pages 214–228.
10. Burton-Jones, A. and Weber, R. (2003). Properties do not have properties: Investigating a questionable conceptual modeling practice. In D. Batra, J. Parsons, and V. Ramesh, editors, *Proceedings of the Second Annual Symposium on Research in Systems Analysis and Design, Miami, FL.*
11. Burton-Jones, A., Wand, Y., and Weber, R. (2007). Informational equivalence, computational equivalence, and the evaluation of conceptual modelling. In *Proceedings of the 6th AIS SIGSAND Symposium*, pages 35–42.
12. Coad, P. and Yourdon, E. (1990). *Object-Oriented Analysis*. Yourdon Press, Englewood Cliffs, NJ.
13. Davies, S. P., Gilmore, D. J., and Green, T. R. (1995). Are objects that important? Effects of expertise and familiarity on classification of object-oriented code. *Human Computer Interaction*, **10**, 227–248.
14. Embley, D. W. (1992). *Object-oriented systems analysis: a model-driven approach*. Prentice Hall, Inc., Englewood Cliffs, NJ.
15. Evermann, J. (2005a). The association construct in conceptual modelling - an analysis using the Bunge ontological model. In O. Pastor and J. F. e Cunha, editors, *Proceedings of the 17th International Conference on Advanced Information Systems Engineering, CAiSE 2005, Porto, Portugal*, pages 33–47. Springer Verlag.
16. Evermann, J. (2005b). Towards a cognitive foundation for knowledge representation. *Information Systems Journal*, **15**, 147–178.
17. Evermann, J. and Wand, Y. (2001a). An ontological examination of object interaction in conceptual modeling. In *Proceedings of the Workshop on Information Technologies and Systems WITS'01, New Orleans, December 15-16, 2001*, pages 91–96.
18. Evermann, J. and Wand, Y. (2001b). Towards ontologically based semantics for UML constructs. In H. Kunii, S. Jajodia, and A. Solvberg, editors, *Proceedings of ER 2001*, pages 354–367.
19. Evermann, J. and Wand, Y. (2005a). Ontology based object-oriented domain modelling: Fundamental concepts. *Requirements Engineering*, pages 146–160.
20. Evermann, J. and Wand, Y. (2005b). Towards formalizing domain modeling semantics in language syntax. *IEEE Transactions on Software Engineering*, **31**(1), 21–37.
21. Evermann, J. and Wand, Y. (2006). Ontological modelling rules for UML: An empirical assessment. *The Journal of Computer Information Systems*, **46**(5).
22. Gemino, A. (1999). *Empirical Comparisons of Systems Analysis Modeling Techniques*. Ph.D. thesis, University of British Columbia, Canada.
23. Gemino, A. and Wand, Y. (2003). Evaluating modeling techniques based on models of learning. *Communications of the ACM*, **46**(10), 79–84.
24. Gemino, A. and Wand, Y. (2004). A framework for empirical evaluation of conceptual modeling techniques. *Requirements Engineering*, **9**, 248–260.
25. Gemino, A. and Wand, Y. (2005). Complexity and clarity in conceptual modelling: Comparison of mandatory and optional properties. *Data and Knowledge Engineering*, **55**(3), 301–326.
26. Green, P. and Rosemann, M. (2000). Integrated process modelling: An ontological analysis. *Information Systems*, **25**(2), 73–87.
27. Green, P. and Rosemann, M. (2001). Ontological analysis of integrated process models: Testing hypotheses. *The Australian Journal of Information Systems*, **9**, 30–38.
28. Green, P., Rosemann, M., and Indulska, M. (2005). Ontological evaluation of enterprise systems interoperability using ebXML. *IEEE Transactions on Data and Knowledge Engineering*, **17**, 713–725.

29. Harel, D. and Rumpe, B. (2004). Meaningful modeling: What's the semantics of "semantics"? *IEEE Computer*, pages 64–72.
30. IBM (1997). *Developing object-oriented software: An experience-based approach*. Prentice Hall, Inc., Upper Saddle River, NJ.
31. Larkin, J. H. and Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, **11**, 65–99.
32. Martin, J. and Odell, J. J. (1992). *Object-oriented analysis and design*. Prentice Hall, Englewood Cliffs, NJ.
33. Mayer, R. E. (1987). *Educational Psychology: A Cognitive Approach*. Little Brown, Boston, MA.
34. Mayer, R. E. (1989). Models for understanding. *Review of Educational Research*, **59**(1), 43–64.
35. McKeithen, K., Reitman, J., Rueter, H., and Hirtle, S. (1981). Knowledge organization and skill differences in computer programmers. *Cognitive Psychology*, **13**, 307–325.
36. Nyerges, T., Moore, T., Montejano, R., and Compton, M. (1998). Developing and using interaction coding systems for studying groupware use. *Human-Computer Interaction*, **13**, 127–165.
37. OMG (2005). *Unified Modeling Language: Superstructure*. Object Management Group. Document formal/05-07-04.
38. Opdahl, A. and Henderson-Sellers, B. (2002). Ontological evaluation of the UML using the Bunge-Wand-Weber model. *Software and Systems Modeling*, **1**(1), 43–67.
39. Opdahl, A., Henderson-Sellers, B., and Barbier, F. (1999). An ontological evaluation of the OML metamodel. In E. Falkenberg and K. Lyytinen, editors, *Information System Concepts: An Integrated Discipline Emerging*. IFIP/Kluwer, Netherlands.
40. Parsons, J. and Cole, L. (2005). What do the pictures mean? Guidelines for experimental evaluations of representation fidelity in diagrammatical conceptual modeling techniques. *Data and Knowledge Engineering*, **55**, 327–342.
41. Pennington, N., Lee, A. Y., and Rehder, B. (1995). Cognitive activities and levels of abstraction in procedural and object-oriented design. *Human Computer Interaction*, **10**, 171–226.
42. Rosemann, M., Recker, J., Indulska, M., and Green, P. (2006). A study of the evolution of the representational capabilities of process modeling grammars. In *Proceedings of CAiSE 2006, Luxembourg, June 5-9*, pages 447–461.
43. Rumbaugh, J. (1987). Relations as semantic constructs in an object-oriented language. In *Proceedings of the 1987 Conference on Object Oriented Programming Systems and Languages and Applications, Orlando, FL.*, pages 466–481. ACM Press.
44. Scaife, M. and Rogers, Y. (1996). External cognition: How do graphical representations work? *International Jounral of Human-Computer Studies*, **45**, 185–213.
45. Siegfried, S. (1995). *Understanding object-oriented software engineering*. IEEE Press, New York, NY.
46. Topi, H. and Ramesh, V. (2002). Human factors research on data modeling: A review of prior research, an extended framework and future research directions. *Journal of Database Management*, **13**(2), 3–19.
47. Wand, Y. and Weber, R. (1993). On the ontological expressiveness of information systems analysis and design grammars. *Journal of Information Systems*, (3), 217–237.
48. Wand, Y., Storey, V. C., and Weber, R. (1999). An ontological analysis of the relationship construct in conceptual modeling. *ACM Transactions on Database Systems*, **24**(4), 494–528.