

Problems with recursive relationships and relationships with attributes in ER models

Andrew Burton-Jones
 Sauder School of Business
 University of British Columbia
 Vancouver, Canada
andrew.burton-jones@sauder.ubc.ca

Kate Lazarenko
 NEHTA - National E-Health Transition Authority
 Brisbane, Australia
Kate.Lazarenko@nehta.gov.au

Ron Weber
 Faculty of Information Technology
 Monash University
 Caulfield East, Australia
ron.weber@monash.edu

ABSTRACT

Systems analysts and information modelers create conceptual models to describe and document the semantics associated with the domain to be supported by an information system. These models typically are generated using a semi-formal grammar that provides constructs for representing semantics and rules for employing and combining the constructs. In this research in progress, we study two widely used conceptual modeling constructs—namely, relationships with attributes and recursive relationships. Although both constructs might be useful for some modeling purposes, they also suffer from a number of limitations that currently are not well understood. We describe these limitations and recommend that modelers avoid using both constructs when an accurate and complete understanding of a domain's semantics is critical. We also provide alternative ways to model a domain's semantics that have the purpose of overcoming the limitations associated with these constructs. These alternatives are motivated by prior research in the conceptual modeling area on classification and optional properties. They involve clarifying variations in semantics associated with different classes of things. These variations in semantics arise from natural or social laws that operate in a domain.

Keywords

Conceptual model, ontology, semantics, ER model.

INTRODUCTION

Underlying the design of any information system is an implicit or explicit model of the domain the system supports. For this reason, models play a critical role in systems development (Selic 2003). Many different types

are used, ranging from conceptual models, which represent the semantics of the domain, to design and implementation models, which show how these semantics will be implemented in a given technology.

The focus of the research we are currently undertaking is conceptual models. These models are produced using a conceptual modeling grammar. Arguably, the most widely used grammars are the entity-relationship (ER) grammar (Davies et al. 2006; Fettke 2009) and UML's class diagrams grammar (Dobing and Parsons 2006). Like any grammar, they contain constructs and rules for using these constructs. In our research, we are examining two constructs that are often employed in ER diagrams and UML class diagrams—namely, relationships with attributes and recursive relationships (Figures 1A, 1B).

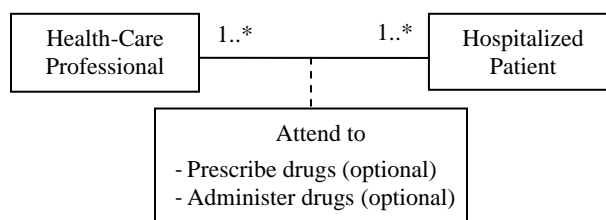


Figure 1A. A Relationship with Attributes

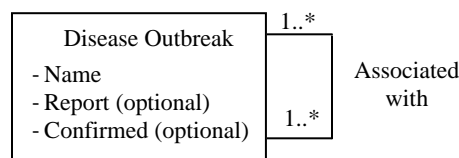


Figure 1B. A Recursive Relationship

Relationships with attributes and recursive relationships were both described in the original specifications of the ER grammar and UML class diagrams grammar (Chen 1976; Rumbaugh et al. 1999). They continue to be described and advocated in widely used texts (e.g., Hoffer et al. 2007). Clearly, their extensive use over many years suggests they have useful characteristics. Nonetheless, the precise nature of these characteristics has not received in-depth research. In this regard, we are aware of only four studies of the strengths and weaknesses of relationships with attributes (Burton-Jones and Weber 1999; Burton-Jones and Weber 2003; Parsons and Cole 2004; Evermann and Halimi 2008). Moreover, we are not aware of any in-depth studies of the strengths and weaknesses of recursive relationships.

The four studies that examined relationships with attributes offer different opinions on the merits of this construct. In terms of benefits, Evermann and Halimi (2008) suggest they offer a way to model bundles of mutual properties that arise when things interact in a domain. For instance, in Figure 1, prescription and administration of drugs happens only as a result of an interaction between a health-care professional and a patient. A relationship with attributes provides a concise way of representing this interaction (see also Evermann and Wand 2005, pp. 151-153).

A second purported benefit of relationships with attributes is that they provide a way to show property precedence (Parsons and Cole 2004). Property precedence is an ontological notion used to understand how one or more properties precedes, is more general than, or is sufficient for one or more other properties (Bunge 1977, pp. 80-81). For example, to say the property of attending to a patient precedes the property of prescribing a drug is to say health professionals will only prescribe drugs on a subset of occasions in which they attend to a patient and will never prescribe drugs without attending to a patient.

Two studies have highlighted potential limitations of using relationships with attributes. Burton-Jones and Weber (1999) suggest they could lead to lack of ontological clarity in conceptual models when the attributes of a relationship are used to represent properties of a property. Because some ontological theories propose that properties do not have properties (e.g., Bunge 1977, pp. 98-99), their use in a conceptual model could reflect so-called 'construct excess' (a case in which a grammatical construct does not have an ontic correlate) (Wand and Weber 1993). Based on this reasoning, Burton-Jones and Weber (1999) predict users will sometimes find such constructs to be unclear and unnatural. A second criticism has been that relationships with attributes could result in a loss of semantics about laws constraining things in the domain that is being modeled (Burton-Jones and Weber 2003). In the research we are currently undertaking, we extend these views.

We are not aware of any investigations of the merits of recursive relationships. Intuitively, one benefit is that they provide a means of abstraction, thereby helping analysts to simplify semantics that would otherwise appear complex. Simplification is often an important benefit of diagrams (Moody 2009). Nevertheless, we are not aware of any systematic studies that examine how well recursive relationships serve this goal. Moreover, Dullea and Song (1999) show recursive relationships can be quite complex. Also, at least one study suggests that semantics in such models can sometimes be lost—for instance, by losing information about wholes and parts in a domain (Wand et al. 1999, p. 525). We provide a more general analysis of this view below.

In summary, the aim of the research we are undertaking is to determine (a) why certain problems can arise with relationships with attributes and recursive relationships, and (b) how these problems might be overcome without losing the benefits that can be obtained from using them (e.g., understanding interactions and property precedence and benefiting from abstraction and simplification).

THE UNDERLYING PROBLEM

When constructing a conceptual model, a fundamental task that systems analysts and information modelers must perform is to decide which classes of things to show in the model. A key principle in choosing classes is that each instance in a class should have a set of common properties:

“Identifying phenomena as instances of the same class indicates that they are similar in some way. In traditional ...approaches, a class is viewed as a set of instances possessing a common set of properties.... An instance can be a material object, action, event, or any other phenomenon. [A] property refers to any statement about the characteristics of an instance, including static aspects (e.g., attributes), dynamic aspects (possible changes), and rules (constraints on attributes and changes)” (Parsons and Wand 2008, p. 842, emphasis added).

A major reason why commonality (or variation) arises among the properties of things in a domain is that the properties often reflect the existence of a natural or social law. The law restricts values the properties can take or restricts relationships among the properties (Bunge 1977, p. 129). For example, natural laws restrict the values of a person's age, and social laws restrict a firm's compensation policies relating to employees' salaries and the relationship between employees' salaries and their seniority. The laws themselves are properties of the things they cover. Unfortunately, it is easy to lose sight of these laws and create classes that do not take their existence into account.

For instance, consider the conceptual models in Figure 2. To some extent, each model masks variations in the properties (or combinations of properties) that things have as a result of some law. Figure 2A shows this problem is not limited to models that employ relationships with attributes or recursive relationships. Rather, it is a more-general problem that can occur with conceptual models.

A frequent reason why property variations are sometimes masked in a conceptual model is the use of optional properties. For instance, in Figure 2A, employees may or may not receive a bonus depending on the value of another of their intrinsic properties—namely, their level of seniority within their organization. The employee class therefore fails Parsons and Wand’s (2008) requirement that classes possess a set of properties in common, because only some members of the class possess the bonus property.

In Figure 2B, the cardinality (multiplicity) constraints pertaining to the Assignment association class apply to only one of the properties of this class—namely, the charge-rate property. Different cardinality (multiplicity) constraints are needed if the conceptual model is to show that each client in the set of clients to which an employee has been assigned must be given a different priority level. As a mutual property of Employee and Client, therefore, Assignment misrepresents some aspects of their interaction/relationship.

In Figure 2C, Mentor is a mutual property (represented via a recursive association) between two different employees. One has the role of a mentor; the other has the role of a mentee. Not all employees play the role of mentors, however. Similarly, not all employees play the role of mentee. Moreover, a mentor can have one of more

mentees, whereas a mentee can have only one mentor. Mentors are paid a bonus depending on the number of employees they mentor. Thus, the Employee class fails Parsons and Wand’s (2008) requirement that classes possess a set of properties in common, because only some members of the class possess the bonus intrinsic property and the mentor mutual property.

The examples in Figure 2 are just a subset of the many ways this underlying problem can occur. We propose, however, that in all these cases the solution is to reclassify classes in the model so commonality (homogeneity) exists among properties in the class. As the earlier quote from Parsons and Wand (2008) indicates, such classes may include classes of things (as in the employee class in Figure 2), actions (as in Figure 2B), or any other type of class. The next sections illustrate this line of argument for relationships with attributes and recursive relationships.

RELATIONSHIPS WITH ATTRIBUTES: PROBLEMS AND SOLUTIONS

Figure 3 shows a relationship with attributes. Following principles in past research, the association class is used to represent a bundle of properties (charge rate, priority level, leader, and overtime rate) that arise from an interaction (an assignment) between an employee and a client (Evermann and Wand 2005; Evermann and Halimi 2008). Similarly, following principles of property precedence, each attribute of assignment is preceded by the assignment itself (Parsons and Cole 2004).

Nonetheless, even when these principles are followed, the model could still be problematic if variations arise as a result of laws in the domain. For instance, consider the following three cases:

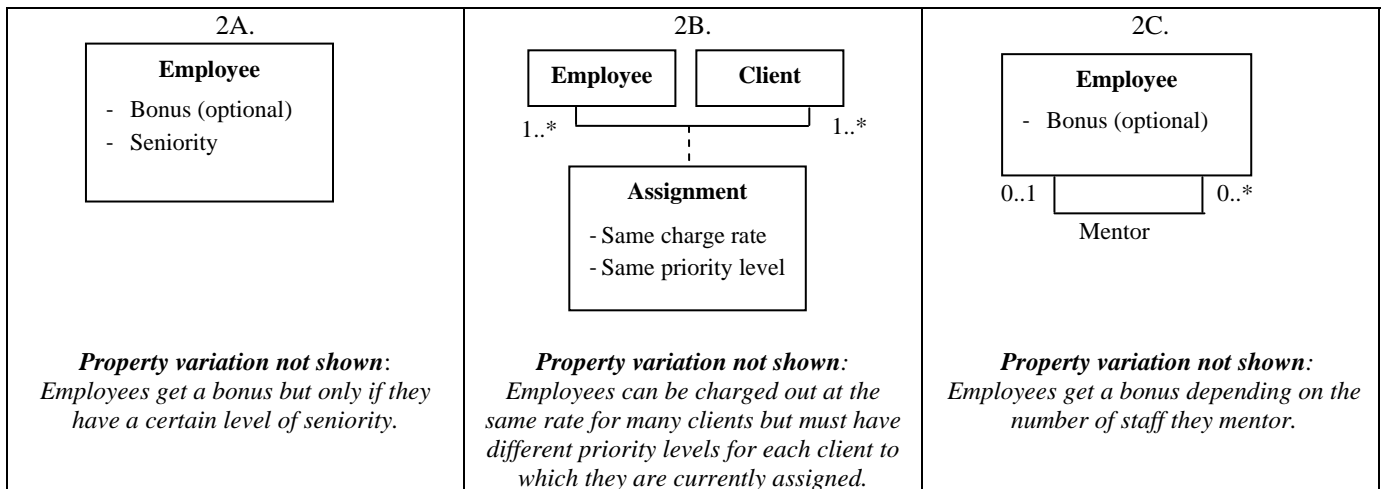


Figure 2. How Inappropriate Classification Can Mask Property Variations in a Domain

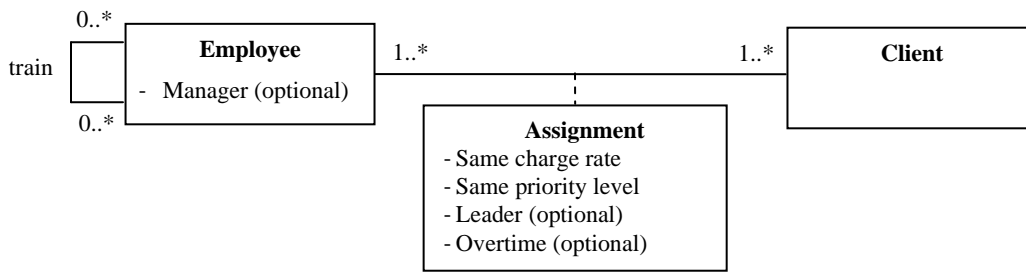


Figure 3. A Model with Relationships with Attributes

1. **Participation:** Some attributes of the relationship have a more-restricted level of participation in the relationship than the level expressed for the relationship as a whole. Following Figure 2B, an example in Figure 3 is where an employee can have assignments with many clients at one time and have the same charge-out rate for different clients. They are constrained, however, to have different priority levels for each of the clients they are servicing at a particular time.
2. **Attributes:** The presence or value of some attribute of a relationship depends on the presence or value of some other attribute of the relationship (or of the things involved in the relationship). For example, in Figure 3, perhaps assignment leaders must be managers, and as a result they are not eligible for overtime payments. Hence, leader, overtime, and manager are shown as optional properties.
3. **Relationships:** The presence or value of some attributes of the relationship depends on the presence or value of another relationship between things in that domain. For example, in Figure 3 perhaps all leaders of assignments are required to undertake training of subordinates. Hence, train is shown as an optional property, because only some employees (managers, who are also leaders) undertake training.

In all these cases, the relationships-with-attributes construct in Figure 3 fails to show a complete set of domain semantics. This is because it “bundles” properties into the relationship. As a result, different laws are also bundled together, which results in a loss of information. The solution, once again, is to specify a new model that ensures commonality of properties within classes (both classes of things and classes of interaction). We follow this approach in Figure 4, which provides a solution for each of the problematic cases we have described above.

<p>Excerpt of solution for Figure 3, Case 1 (Participation). For clarity, other parts of the model not relevant to Case 1 are not shown in this excerpt.</p>	
<p>Excerpt of solution for Figure 3, Case 2 (Attributes). For clarity, other parts of the model not relevant to Case 2 are not shown in this excerpt.</p>	
<p>Excerpt of solution for Figure 3, Case 3 (Relationships). For clarity, other parts of the model not relevant to Case 3 are not shown in this excerpt.</p>	

Figure 4. Models that Clarify Variations in Properties within Classes

RECURSIVE RELATIONSHIPS: PROBLEMS AND SOLUTIONS

Figure 5A shows a class with two recursive relationships (adapted from Dullea and Song 1999, p. 3). Variations in properties can be masked in such a model when a dependency exists between relationships or between a relationship and an attribute. For example, Figure 5A shows some employees manage other employees, while other employees manage themselves. This representation masks a variation in properties, because it implies three sets of employees exist—namely, those who manage, those who are managed, and those who both manage and are managed by others.

Similarly, assume an attribute of employees is bonus. Variations will be masked in Figure 5A if the presence or level of a bonus depends on another relationship—for example, whether an employee is a manager. The solutions in such cases involve constructing classes that clarify these variations and ensure homogeneity within remaining classes. For example, Figure 5B clarifies the three subclasses and indicates which subclass obtains a bonus.

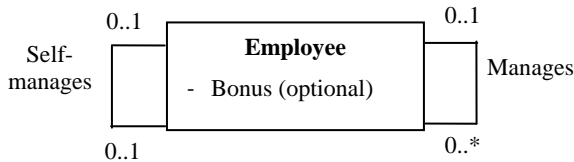


Figure 5a. A Model with Recursive Relationships (Masking Variations)

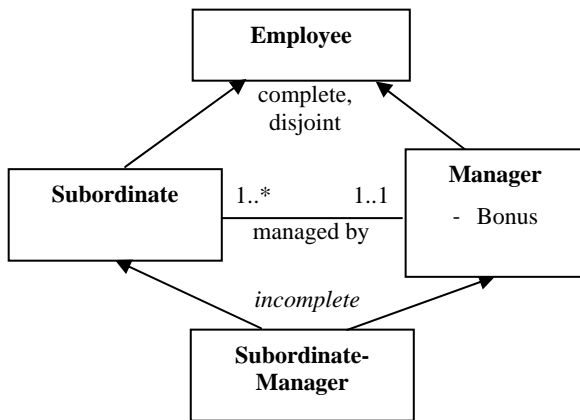


Figure 5b. An Alternative Model (Clarifying Variations)

CONCLUSIONS AND FUTURE WORK

Because both relationships with attributes and recursive relationships bundle properties together in a concise manner, they sometimes mask variations among properties that arise from laws that operate in the domain. When this outcome occurs, these constructs fail to

represent important domain semantics. Such problems can be avoided by replacing relationships with attributes and recursive relationships with alternative constructs that more precisely (a) specify the laws in the domain, and (b) ensure commonality of properties exists among instances of any given class.

One cost of our alternative modeling approach is that our alternative (more-complete) diagrams prima facie often appear more complex (in the sense they contain more instances of grammatical constructs). In this light, depending on modelers’ purposes in preparing a conceptual model, they might wish to have different models of the same phenomena. Some models might be used for high-level communication with stakeholders, in which case having models with fewer grammatical constructs might be beneficial. Other models might be used for database design or automatic generation of systems, in which case having precise specification of the semantics of a domain is essential. If electronic versions of the models are prepared, users could simply choose which version (complex or simple) they wish to employ at any given time. In some electronic versions, users might also have facilities to “explode” or “collapse” conceptual models depending on their needs.

Our ongoing work is focusing on clarifying the underlying the problems we identified with relationships with attributes and recursive relationships, formalizing the underlying theory and the solutions we propose, and undertaking a program of empirical studies to evaluate the feasibility and usefulness of our recommendations. In addition, we are investigating the relationship between property precedence and functional dependence (as used in the theory of database normalization) to determine how they can be used to help modelers create class structures that preserve commonality among the properties of a class.

ACKNOWLEDGMENTS

This research program is supported by funds from the Australian Research Council.

REFERENCES

Bodart, F., Sim, M., Patel, A., and Weber, R. (2001) Should Optional Properties be Used in Conceptual Modelling? A Theory and Three Empirical Tests, *Information Systems Research*, 12, 4, 385-405.
 Bunge, M. (1977) *Treatise on Basic Philosophy: Volume 3: Ontology I: The Furniture of the World* Reidel, Boston.
 Burton-Jones, A., and Weber, R. (1999) Understanding Relationships with Attributes in Entity-Relationship Diagrams, *Proceedings of the 20th International Conference on Information Systems*, Charlotte, NC, 214-228.

- Burton-Jones, A., and Weber, R. (2003) Properties Do Not Have Properties: Investigating a Questionable Conceptual Modeling Practice, *Proceedings of the 2nd Annual Symposium on Research in Systems Analysis and Design*, D. Batra, J. Parsons and V. Ramesh (eds.), Miami, FL, p. 14 pp.
- Chen, P.P.S. (1976) The Entity-Relationship Model: Toward a Unified View of Data, *ACM Transactions on Database Systems*, 1, 1, 1976, 9-36.
- Davies, I., Green, P., Rosemann, M., Indulska, M., and Gallo, S. (2006) How do Practitioners use Conceptual Modeling in Practice? *Data & Knowledge Engineering*, 58, 3, 358-380.
- Dobing, B., and Parsons, J. (2006) How UML is Used, *Communications of the ACM*, 49 5, 2006, 109-113.
- Dullea, J., and Song, I.Y. (1999) A Taxonomy of Recursive Relationships and Their Structural Validity in ER Modeling, *Proceedings of the 18th International Conference on Conceptual Modeling (ER '99)*, Lecture Notes in Computer Science 1728, Paris, France, 384-398.
- Evermann, J., and Halimi, H. (2008) Associations and Mutual Properties--An Experimental Assessment, *Proceedings of the 14th Americas Conference on Information Systems*, J. Parsons and Y. Yuan (eds.), Toronto, ON, 1-11.
- Evermann, J., and Wand, Y. (2005) Ontology Based Object-Oriented Domain Modelling: Fundamental Constructs, *Requirements Engineering*, 10, 2005, 146-160.
- Fettke, P. (2009) How Conceptual Modeling Is Used, *Communications of the AIS*, 25, 1, 571-592.
- Hoffer, J.A., Prescott, M.B., and McFadden, F.R. (2007) *Modern Database Management*, (8th ed.) Pearson Prentice Hall, Upper Saddle River, N.J.
- Moody, D.L. (2009) The "Physics" of Notation: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering, *IEEE Transactions on Software Engineering*, 35, 6, 756-779.
- Parsons, J., and Cole, L. (2004) An Experimental Examination of Property Precedence in Conceptual Modelling, *Proceedings of the 1st Asia-Pacific Conference on Conceptual Modelling (APCCM 2004)*, Conferences in Research and Practice in Information Technology, Dunedin, NZ, 10pp.
- Parsons, J., and Wand, Y. (2008) Using Cognitive Principles to Guide Classification in Information Systems Modeling, *MIS Quarterly*, 32, 4, 2008, 839-868.
- Rumbaugh, J., Jacobson, I., and Booch, G. (1999) *The Unified Modeling Language Reference Manual* Addison Wesley, Reading, MA.
- Selic, B. (2003) The Pragmatics of Model-Driven Development, *IEEE Software*, 20, 5, 19-25.
- 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.
- Wand, Y., and Weber, R. (1993) On the Ontological Expressiveness of Information Systems Analysis and Design Grammars, *Journal of Information Systems*, 3, 217-237.