Journal of Information Systems Education, Vol. 17(1)

The Essential Skills of Data Modeling

Richard T. Watson
Department of MIS
University of Georgia
Athens, GA 30602-6273 USA
rwatson@terry.uga.edu

ABSTRACT

The critical data modeling issue is learning to think like a data modeler. The representation method is a less important concern, because all dialects of these methods capture the same core data. For data modeling teachers, there are two issues. First, what representation method enables quick sketching of models on a board? Second, what method should students use to capture the fine detail for their assignments? Other issues related to teaching data modeling are also discussed, including the argument for intertwining the teaching of data modeling and SQL

Keywords: Data Model, Representation, E-R diagram, UML, Teaching Data Modeling, Data Modeling and SQL

1. THE CRITICAL ISSUE

Approximately 20 years of teaching data modeling have provided ample evidence to me that the critical issue is not how to represent entities and relationships. The essential skill is learning to identify entities and the correct relationships among them. Students demonstrate over and over again the difficulty that they have in learning to think like a data modeler. The various representation methods, such as Chen's E-R diagram (Chen 1976) and UML (Rumbaugh, Jacobson, and Booch 1999), are easily learned. However, proficiency with a representation method does not make a data modeler. Data modeling is a higher-level skill than drawing a diagram.

I reiterate continually to my students, and in my textbook (Watson 2006), that the purpose of a data model is to capture reality so that the database based on the data model can be used to answer questions about reality. A model that fails to represent reality is likely to fail at some point because a client's question cannot be converted to SQL. When real-world entities or relationships are not represented in a data model, then real-world queries about these missing entities and relationships cannot be answered. I learned this need to focus on reality representation early in my career when I was given data collected by pharmacy researchers who needed some statistical analysis. The very first analysis they asked me to run, which was central to their research, could not be answered because they had not collected data on the relationship between two central entities. The data model did not represent the reality they wanted to study, and I could not answer a key question about that reality.

Some of the typical errors that students make include:

- Not recognizing that an attribute is an entity
- Failing to generalize several entities as a single entity
- Not reading a relationship both ways and thus making a cardinality mistake
- Ignoring exceptions that result in a failure to represent reality.

These are problems of domain understanding and not representation. It does not matter how you represent errors of domain interpretation. They are still errors.

As far as I am concerned, reality is far more important than representation. Even if there were a single best representation, and I don't believe there is, as I will argue shortly, it is worthless if the resulting data model does not capture the domain of interest's reality. It is all about reality and not so much about representation.

2. REPRESENTATION CHOICES

From a teaching perspective, there are two issues with regard to representation. First, I need a method that enables me to quickly sketch a model on a board. Second, I need a method that my students can use to capture the fine detail of a model and generate SQL create and constraint statements.

Thus, I don't believe that one can argue that one data modeling dialect is superior to another without considering the context in which it is used or the tools available. If a student comes to my office to ask for help with a model, I find it far easier to work with them using a sheet of paper or whiteboard than sitting at a computer using a CASE tool.

2.1 Minimalist, Quick Modeling

For quick modeling on a board or sheet of paper, when dealing with a class or small group of students respectively, a data modeling dialect is required that quickly records entities, identifiers, attributes, and relationships. It should

also be a system that is quickly amended in response to class questions or recognition of exceptions. Thus, I don't worry about modality (optional or mandatory), but I am concerned about recognition of weak entities (represented by a '+') because of their impact on primary keys and maybe foreign keys. My models (see Figure 1) sparsely capture the essentials.

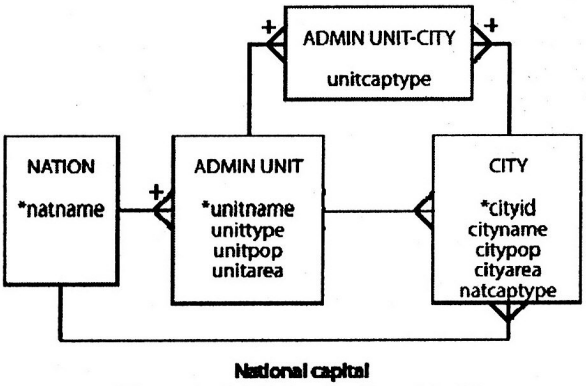

Figure 1: A Minimalist Data Model

2.2 CASE Tools

In my current class, students use DB Visual Architect (DB VA) under the auspices of Visual Paradigm's academic partnership. DB VA is typical of the E-R modeling tools in common use in industry. Models can be drawn quickly and fine detail recorded, such as modality and data type. When the model is complete, students can generate SQL statements for creating the database.

As Table 1 shows, translation between the two representations' dialects is a simple task, and my students have had little difficulty in making the transition between my board drawings and DB VA. Indeed, some students directly translate my drawings into DB VA format in class so they have a record of all the models I have discussed. I had to cover two aspects of DB VA with the class that were not immediately obvious: representation of recursive relationships and weak entities.

Relationship	Minimalist	DB Visual Architect
1:1	——	+---+
1:1 with a weak entity	+——	+—+
1:m	—<	+--O<
1:m with a weak entity	+—<	+—O<

Table 1: Comparison of data modeling dialects

2.3 UML

As an Australian who grew up using a currency based on pounds, shillings, and pence (12 pennies to a shilling and 20 shillings to a pound), and the Imperial measurement system, I am a convert to simpler systems and standards. Australia switched to a decimal currency in 1966 and commenced the transition to the SI system in 1970.

Several years ago I proposed to adopters of my database text that I would switch to UML as the representation dialect. I was surprised at the level of opposition. I also asked some database designers about their preferences, and almost universally they were opposed to the idea. Data modelers are happy with the present representation dialects used by data modeling design packages such as DB Visual Architect. In line with my customers' desires, I did not move to UML. Rather, I added some material showing comparisons of UML class models and the data modeling dialect used in the text.

3. TEACHING MODELING

Given that the central issue is learning to data model and not learning a data model, we need to focus on how to teach data modeling. As the Chinese proverb proclaims, "There are many paths to the top of the mountain, but the view is always the same." I would like now to explain my approach to climbing the data modeling proficiency mountain, which is a result of seeking to find the easiest path to the top.

For more than a decade, I've used an integrated spiral approach. I integrate the teaching of data modeling and SQL so that students can relate the abstract (the data model) to the concrete (querying a database). The spiral element of this approach means that I teach a little bit of data modeling and then enough matching SQL to enable students to make the link between what they can model and how they can create and interrogate such models. For example, the first data modeling class covers a single entity, creating a single table database, and SQL for a single table.

I found that if I taught data modeling first and SQL later, students could not grasp some of key features of the relational model (e.g., foreign keys). The resulting lack of understanding lowered motivation and teaching was more challenging. After teaching in this manner for some terms, I rethought my approach with a view to finding a better method. Fortunately, I recalled from my days of teaching COBOL that I had followed each lesson on a small chunk of the language with a programming exercise to reinforce the concepts covered. This method of teaching COBOL had worked well for me, and thus I transferred the general approach to teaching data modeling and SQL by intertwining them. My classroom experiences soon convinced me that the integrated method was more successful. Students had less problems learning data modeling and quickly understood the purpose of database design and how it related to SQL. They could more readily see the connections between the two elements and gradually developed a cohesive understanding of data modeling and SQL.

4. CONCLUSION

Jonathan Swift, the 18th century Anglo-Irish satirist, wrote about a squabble over which end one should open a boiled egg, the pointy or the flat end (Swift 1955). Of course, it does not matter. What really matters is the taste of the egg.

Similarly, it is not an issue of how you represent the design of database, but whether the design captures reality.

5. REFERENCES

Chen, Peter (1976), "The entity-relationship model - toward a unified view of data." ACM Transactions on Database Systems 1 (1):9-36.

Rumbaugh, James, Jacobson, Ivar and Booch, Grady. (1999), The unified modeling language reference manual, The Addison-Wesley object technology series. Reading, Mass.: Addison-Wesley,

Swift, Jonathan. (1955), Gulliver's travels. London: Oxford University Press,

Watson, Richard T. (2006), Data management: databases and organizations. 5th ed. New York, NY: John Wiley,

AUTHOR BIOGRAPHY

Richard T. Watson is the J. Rex Fuqua Distinguished Chair for Internet Strategy and Director of the Center for Information Systems Leadership in the Terry College at the University of Georgia. Professor Watson has published more than 100 journal articles and written books on electronic commerce and data management. His work has been accepted by leading academic and practitioner journals, and has been translated into several languages. He has given invited seminars in more than 20 countries for companies and universities. Dr. Watson is the Past President of AIS, a visiting professor at Agder University College, Norway, Fudan University, China, and a consulting editor to John Wiley & Sons. He has been a co-chair of ICIS and a senior editor for *MIS Quarterly*.

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096