

How to Draw an Architectural Data Model in UML

This article is an excerpt from David Hay's latest book, *UML and Data Modeling: A Reconciliation*, recently published by Technics Publications. It is Copyright David C. Hay, and reprinted with permission.

Since Leibniz there has perhaps been no man who has had a full command of all the intellectual activity of his day. Since that time, science has been increasingly the task of specialists, in fields which show a tendency to grow progressively narrower. A century ago there may have been no Leibniz, but there was a Gauss, a Faraday, and a Darwin. Today there are few scholars who can call themselves mathematicians or physicists or biologists without restriction.

A man may be a topologist or an acoustician or a coleopterist. He will be filled with the jargon of his field, and will know all its literature and all its ramifications, but, more frequently than not, he will regard the next subject as something belonging to his colleague three doors down the corridor, and will consider any interest in it on his own part as an unwarrantable breach of privacy.

- Norbert Wiener, *Cybernetics*; 1948.¹

The book is about two “camps” in the information management world that each represent large bodies of specialized knowledge. Those in each camp suffer from the specialization phenomenon described above by Dr. Wiener. Each seems to be seriously unenlightened about the other.

Data modeling or object modeling? Whose side are you on? Why are there sides? What's going on here?

After a decade of various people's trying to represent data structures graphically, the entity/relationship version of the data model was formalized in 1976,² and variations on it have followed. The Unified Modeling Language (UML) was officially released a little over twenty years later, in 1997.³ Its adherents claim that UML's “Class Model” is the rightful successor to the data model. Others are not convinced.

¹ Norbert Wiener. 1948, 1961. *Cybernetics: of Control and Communication in the Animal and the Machine, second edition*. (Cambridge, MA, The MIT Press). 2.

² Peter Chen. 1977. “The Entity-Relationship Approach to Logical Data Base Design”. The Q.E.D. Monograph *Series: Data Management*. Wellesley, MA: Q.E.D. Information Sciences, Inc. This is based on his article, “The Entity-Relationship Model: Towards a Unified View of Data”, *ACM Transactions on Database Systems*, Vol. 1, No. 1, (March 1976), pages 9-36.

³ Object Management Group (OMG). 1997. “UML Specification version 1.1”. (OMG document ad/97-08-11). Published at <http://www.omg.org/cgi-bin/doc?ad/97-08-11>.

2 Bibliography

The fact of the matter is that the intellectual underpinnings and the orientation of UML's object-oriented model are very different from those of the data modelers' entity/relationship model. There appears to be a kind of intellectual "impedance mismatch" between the two approaches.* This is partially *technological*, as object-oriented programmers attempt to save persistent object data in relational databases—which have significantly different structures from them.⁴ It's also a *cultural* mismatch, however, coming from significant differences in world views about systems development.⁵ UML, after all, was originally intended to support object-oriented *design*, while data (entity/relationship) modeling was intended to support the *analysis* of business structures. These are very different things.

[The book describes the underpinnings of the two points of view and the points where they disagree. It then goes on to provide an approach to the use of the UML notation for producing a truly conceptual, architectural entity/relationship diagram.]

Summary of the Approach

To create a conceptual (semantic or architectural) entity/relationship model using a UML diagramming tool, follow these guidelines:

1. *Show domain-specific entity cases only.* Consider only classes that are collections of things of significance to the enterprise or the domain being addressed. These are referred to here as *entity classes*.
2. *Use Symbols Selectively.*
 - Use only *appropriate* symbols
 - Class (entity class)
 - Attribute
 - Association (relationship)
 - Cardinality for attributes and relationships
 - Exclusive or (xor) Constraint
 - Use **some UML-specific** symbols with care

* The analogy is derived from electrical engineering, where the term "impedance matching" refers to the use of a transformer to make the load (impedance) required on a target device (such as a loudspeaker) match the load produced on a source device (such as an amplifier). This is described in (among other places): American Radio Relay League, 1958. *The Radio Amateur's Handbook: The Standard Manual of Amateur Radio Communication*. (Concord, New Hampshire: The Rumford Press).

⁴ Ted Neward. 2006. "The Vietnam of Computer Science", *The Blog Ride: Ted Neward's Technical Blog*. June 26, 2006. Retrieved from <http://blogs.tedneward.com/2006/06/26/The+Vietnam+Of+Computer+Science.aspx>, 7/10/2011.

⁵ Scot Ambler. 2009 "The Cultural Impedance Mismatch", *The Data Administration Newsletter*. August 1, 2009. Available at: <http://www.tdan.com/view-articles/11066>.

- Enumeration
- Derived Attribute
- Package

Do not use any other UML symbols

- Abstract entities
- Association class
- Behavior
- Composition
- Navigation
- Ordered
- Visibility

➤ *Add* one symbol

- <<ID>> stereotype
- (Or use new property {isID})

3. Define Data Model Relationship ends as Predicates, not UML Roles.
4. Define domains, using data types.
5. Understand “Packages and “Namespaces”.
6. Follow Display Conventions:
 - *Spaces in Names* – Include spaces inside multi-word entity class and attribute names.
 - *Role Positions* – Position the predicate next to the object entity.
 - *XOR* – Do not include the label in an “XOR” relationship.
 - *Cardinality Display* – Display mandatory one cardinality as “1..1”, not “1”.
7. For *aesthetic* reasons, do the following.
 - Stretch and position entity class boxes so that no relationship has an “elbow”. (There are no bent lines).
 - Turn off the ability to display operations, so the entity class box has only one horizontal line.
 - Arrange the entity classes so that the “many” end of each relationship is at the left or top (the “starry skies” approach).
 - Limit a subject area to no more than 15 entity classes to show on one page.
 - Present the model in a succession of diagrams. On the first diagram, show no more than 2-5 entity classes, all highlighted. On each successive page, add no more than 2-5 entity classes and highlight them.

4 Bibliography

- In general, display attributes only in the diagram where their entity class first appears. Suppress them on all subsequent diagrams, unless they are needed to explain a particular concept. (Suppress them by coloring them white.)

[The book includes a chapter from *Enterprise Model Patterns: Describing the World*, to demonstrate the approach described. It also includes, as an appendix, a history of the information processing industry, in terms of two streams: first data processing and object-oriented design; second data management and architecture. It is from history that the “impedance mismatch” arose.]

Biography:

In the Information Industry since it was called “data processing”, Dave Hay has been producing data models to support strategic and requirements planning for more than twenty-five years. As President of Essential Strategies, Inc. for 18 years, Dave has worked in a variety of industries, including, among others, finance, clinical pharmaceutical research, and all aspects of oil production and processing. Projects entailed various aspects of defining corporate information architecture, identifying requirements, and planning strategies for the implementation of new systems.

Dave’s recently-published book, “*Enterprise Model Patterns: Describing the World*”, is an “upper ontology” consisting of a comprehensive model of any enterprise—from several levels of abstraction. It is the successor to his ground-breaking 1995 book, “*Data Model Patterns: Conventions of Thought*”. In between, he has written comprehensive books on requirements analysis and metadata design.

He has spoken at numerous international and local DAMA, semantics, user group, and other conferences.