Shay Colson
S. Dischiave
IST 659
September 15, 2009

Data Modeling - Conceive, Collaborate, Create

Introduction: The early conceptual beginnings of data modeling trace back to the origins of the database itself. While E.F. Codd was finalizing his thoughts on the relational database, others were working on techniques that would eventually empower the creation of the modern database and take databases to new heights. Not only is performing even the most basic database design impossible without the use of data modeling, a well-implemented data modeling solution can elegantly facilitate everything from the simplest Microsoft Access database to a complete business process re-engineering. This paper will examine the history of data modeling, describe the most popular and accepted methods, and discuss in detail data modeling best practices as they apply to database design.

Best Practice: Simply put, a data model is a way of "describing what sort of data will be held and how it will be organized." (Simsion and Witt 2004) Data models provide "a set of concepts that can be used to describe the structure of and operations on a database." (Navathe 1992). Instead of relying on the habits of each individual project manager or database administrator, data models provide a formalized methodology for designing databases that incorporates several levels of functionality and data relations.

In a seminal 1974 article, the American National Standards Institute (ANSI) laid out three distinct instances of data modeling: conceptual schema, logical schema, and physical schema (Bachman 1974). These three schema form the bedrock upon which nearly all modern databases are designed. Rather than thinking of these three schema as mutually exclusive, worrying that you must choose one and only one, perhaps it is more effective to envision them as overlapping processes with distinct characteristics and capabilities. A successful database design process will almost certainly incorporate more than one data modeling schema. Scott Ambler includes an excellent description of each type in his 2004 book, <u>The Object Primer</u>:

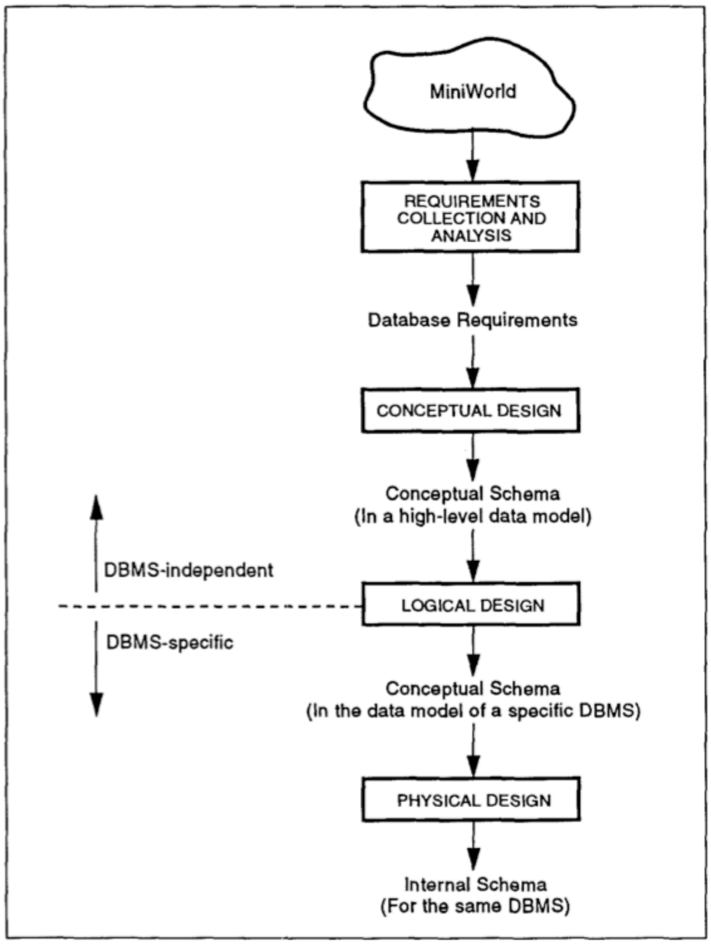

Figure 1: Database Schema Design Process (Navathe 1992)

•**Conceptual:** "Used to explore domain concepts with project stakeholders." These "models are often created as part of your initial requirements envisioning efforts as they are used to explore the high-level static business structures and concepts."

•**Logical: "**Used to explore the domain concepts, and their relationships, of your problem domain." As opposed to the "high-level" view of the conceptual schema, the logical schema depict "entity types, the data attributes describing those entities, and the relationships between the entities."

•**Physical:** "Used to design the internal schema of a database, depicting the data tables, the data columns of those tables, and the relationships between the tables." (Ambler 2004) Physical models also concentrate, as one might expect, on the physical computing hardware needed to enable a particular database.

In addition to the modeling schema mentioned above, data modeling typically involves one of two methodologies: Strategic (top-down) or System Analysis (bottom-up). These methodologies outline information gathering and orientation strategies that are employed as the project moves through the above-mentioned schema.

Whitten defines two methods of data modeling in his <u>Systems Analysis and Design Methods</u> book, which other industry experts agree are the most common and useful (Silverston, Inmon et al. 1997):

- *Strategic Data Modeling*: Building on the strategic vision and plans of the organization, Strategic Data Modeling includes the "vision and architecture for information systems," allowing organizations to identify, prioritize, and develop projects using their enterprise data model as a starting point for any projects. (Whitten and Bentley 2005) Because this methodology begins with a strategic vision or plan and works down the enterprise structure to achieve the end, methodology similar to this is also referred to as "top-down" data modeling.
- *Data modeling during systems analysis*: Typically, data modeling during system analysis focuses on a single information system where the model includes only entities and relationships. (Whitten and Bentley 2005) Because this methodology begins with entities and relationship (or the bottom of the organizational chart), methodology similar to this is also referred to as "bottom-up" data modeling.

Rather than deciding arbitrarily which method of data modeling is best, individuals and organizations should make their choice based on the needs of the particular project. See *Conclusion and Recommendations* for more on this selection process.

Computer Aided Software Engineering (CASE) has given rise to a multitude of data modeling tools that can be used to assist in working through the modeling schema process. Many of these tools can output directly into database management systems (DBMS), allowing much of the groundwork performed in the data modeling phase to remain directly applicable in the implementation phase. These tools also continue to blur the lines between the three design schema (conceptual, logical, and physical) and the two most prevalent data modeling methods (top-down and bottom-up).

Assessment: Data modeling techniques have become the cornerstone of database design. Without employing at least some type of data modeling technique, building any sort of successful database becomes impossible. There should be no question that using data modeling techniques is a database design best practice, and should be the first step of any database project, regardless of size, scope, purpose, or budget. Deciding which techniques to employ is admittedly a much more difficult question. Would your project benefit from a top-down approach, or would a bottom-up approach yield better results? Should you begin with a strategic vision and work down from there, or incorporate the data modeling during systems analysis from the bottom up? Each database project will demand a different decision, but for some general recommendations, see the next section.

Conclusion and Recommendations: When deciding which methodologies to employ, organizations should follow a few general rules (but note that these are not always the case - what follow are generalizations):

1. Nearly all database design projects should begin with the conceptual design schema. One instance where it may be useful to begin with the physical design schema would be if you were migrating an existing database to different hardware or a new DBMS and needed only to change the physical components, not the data or the organization or relationships of that data.
2. Employ a CASE tool. CASE tools allow advancements made in the design process to be easily incorporated into the implementation phase of any database project, and also make revisions, additions, or changes easier to include as well.
3. When choosing between top-down and bottom-up, consider the role of the project in the whole of the organization. If the database stands to revolutionize the entire way the organization does business, perhaps a top-down approach would yield better results. If the project is focused on one particular department, line of business, or some other relatively small need, then perhaps a bottom-up approach will work fine. In general, if data sharing is of high-priority for the project, a top-down approach will better facilitate this process.

Data modeling forces an organization to confront the reality of their data needs. Not only does data modeling ensure that organizations to very literally step through their data flow, data value, structure, and use, but well-done data modeling can provide insights that would otherwise not be readily apparent. Designing organizational databases on top of well-modeled plans that are appropriate for the organization is the key to successful database design and implementation.

Glossary

Data modeling: "describing what sort of data will be held and how it will be organized." *(Simsion and Witt 2004)*

Conceptual Design Schema: "Used to explore domain concepts with project stakeholders. These models are often created as part of your initial requirements envisioning efforts as they are used to explore the high-level static business structures and concepts."

Logical Design Schema: "Used to explore the domain concepts, and their relationships, of your problem domain. As opposed to the "high-level" view of the conceptual schema, the logical schema depict entity types, the data attributes describing those entities, and the relationships between the entities."

Physical Design Schema: "Used to design the internal schema of a database, depicting the data tables, the data columns of those tables, and the relationships between the tables. (Ambler 2004) Physical models also concentrate, as one might expect, on the physical computing hardware needed to enable a particular database."

Top-Down Design: See Strategic Data Modeling.

Strategic Data Modeling: "Building on the strategic vision and plans of the organization, Strategic Data Modeling includes the "vision and architecture for information systems," allowing organizations to identify, prioritize, and develop projects using their enterprise data model as a starting point for any projects."

Bottom-Up Design: See Data Analysis During System Design.

Data Analysis During System Design: "Typically, data modeling during system analysis focuses on a single information system where the model includes only entities and relationships." (Whitten and Bentley 2005) Because this methodology begins with entities and relationship (or the bottom of the organizational chart), methodology similar to this is also referred to as "bottom-up" data modeling.

Computer Aided Software Engineering (CASE): "Tools which address the problems associated with various phases of software development." *(Bergin 1993)*

Database Management Systems (DBMS): "A set of programs used to define, administer, and process databases and their associated applications." *(Taylor 2006)*

Sources Cited

Ambler, S. W. (2004). The Object Primer: Agile Model-Driven Development with UML 2.0, Cambridge University Press.

Bachman, C. W. (1974). "Summary of current work ANSI/X3/SPARC/study group: database systems." SIGMOD Rec. 6(3): 16-39.

Bergin, T. J. (1993). Computer-aided software engineering, Idea Group Inc (IGI).

Navathe, S. B. (1992). "Evolution of data modeling for databases." Commun. ACM 35(9): 112-123.

Silverston, L., W. H. Inmon, et al. (1997). The Data Model Resource Book: A Library of Logical Data and Data Warehouse Designs, John Wiley & Sons.

Simsion, G. and G. Witt (2004). Data Modeling Essentials, Third Edition, Morgan Kaufmann.

Taylor, A. G. (2006). SQL for dummies, For Dummies.

Whitten, J. and L. Bentley (2005). Systems Analysis and Design Methods, McGraw-Hill/Irwin.