Chapter XI

Criteria for Comparing Information Modeling Methods: Informational and Computational Equivalence

Keng Siau, University of Nebraska-Lincoln, USA

ABSTRACT

Information modeling methods are key to the success of IS development projects. The problem in the software engineering field is not the lack of modeling methods but the proliferation of modeling methods with little theoretical foundation and empirical evidence to support their usefulness. Evaluation techniques are therefore necessary to compare and contrast these modeling methods. In this chapter, we suggest an evaluation approach based on the human information processing paradigm and the theory of equivalence of representations. This evaluation approach proposes that modeling methods can be evaluated and compared using two criteria: informational and computational equivalence.

INFORMATION MODELING

Information modeling is the process of formally documenting the problem domain for the purpose of understanding and communicating amongst stakehold-
Information models are central to information systems analysis and design. On one hand, they provide the conceptual basis for communicating and thinking about information systems (Willumsen, 1993). On the other hand, they provide a formal basis for tools and techniques used in the design and development of information systems (Kung & Solvberg, 1986). The phrase, “Let a thousand flowers bloom”, offers an apt description of the current state of the information modeling field, which is inundated by hundreds of different modeling methods (Bubenko, 1986; Olle, Sol & Verrijn-Stuart, 1982) and over 1000 brand name methodologies worldwide (Jayaratna 1994). The quest by researchers to develop the next modeling technique has been wittily termed the YAMA (Yet Another Modeling Approach) syndrome and the NAMA (Not Another Modeling Approach) hysteria.

Structured modeling was one of the most important modeling paradigms. It is a data-oriented approach with data flow as the central feature. Even though the object-oriented method became popular in the early 1980s, it was not until the 1990s that a shift to object-oriented modeling began to emerge, and the object-oriented paradigm gradually became the standard approach throughout the whole software development process (Engels & Groenewegen, 2000). The focus of object-oriented modeling is objects, which interact with each other in the system through their inherent behaviors (Rumbaugh et al., 1991).

Ironically, it was a so-called method war that hindered the success of object-oriented modeling approaches in the beginning of the 1990s (Engels & Groenewegen, 2000). There were more than 50 object-oriented methods in 1994 (Booch, Rumbaugh & Jacobson, 1999). As a result, users of these methods encountered tremendous trouble finding a modeling language that could completely meet their needs in object-oriented modeling and design. The object-oriented method war came to an end by an industry standard modeling language: Unified Modeling Language (UML) (Siau & Cao, 2001).

Despite the vibrant research arena in modeling, most modeling methods are introduced based on the common sense and intuition of researchers. Theoretical foundation is either nonexistent or considered nonessential (Siau, Wand & Benbasat, 1997). For example, Coad and Yourdon (1991) wrote:

> It would be intellectually satisfying to the authors if we could report that we studied the philosophical ideas behind methods of organization, from Socrates and Aristotle to Descartes and Kant. Then, based on the underlying methods human beings use, we could propose the basic constructs essential to an analysis method. But in truth we cannot say that, nor did we do it. (p. 16)

However, one’s common sense may be wrong and radically misleading at times. Churchland (1988) pointed out that our common-sense psychological