Conceptual Commonalities in Modeling of Business and IT Artifacts

Haim Kilov
Stevens Institute of Technology, USA

Ira Sack
Stevens Institute of Technology, USA

INTRODUCTION

The proverbial communication gap between business and IT experts is mostly due to the fact that what is considered obvious to some business experts might not be obvious or even known to IT experts, and might substantially differ from what is considered obvious to other business experts. Thus, different stakeholders may understand the business domain and problems in tacit and quite different ways, and at the same time might be unaware of these differences. This leads to business-IT misalignment, and therefore to many serious information system failures, from life-threatening to financial or simply very annoying (loss of customers' trust and patience).

The article provides a concise overview of the topic, includes definitions of some essential concepts and constructs together with industrial examples of their use in modeling and in fostering business-IT alignment, and shows how a small subset of UML has been successfully used to represent the essential structure of a model.

BACKGROUND

Creating business and IT system models readable and understandable to all stakeholders is possible only if the system of concepts underlying such models is well-defined, understandable to the model *readers*, and does not require extensive and painstaking explanations. Because the experiences of different stakeholders are usually quite different, the fundamental underlying concepts should be invariant with respect to the specific business (or IT) domain of interest.

Fortunately, a *system* of simple and elegant *abstract* modeling concepts and constructs exists and has been stable for centuries. Precise definitions of its semantics come from exact philosophy, mathematics, programming, and systems thinking. It has been successfully used in theory, in industrial practice (including international standards such as the Reference Model of Open Distributed Processing (ISO/IEC, 1995)), and in teaching of business and IT modeling. It includes such generic concepts as system, model, abstraction, structure, relationship, invariant, state, action, behavior, conformance, type, composition, template, name in context, and so forth, thus providing a solid foundation for systems of appropriate domain-specific concepts, such as contract, trade, confirmation, derivatives, options trade, and so forth, for the financial domain.

Why are we reusing concepts from exact philosophy? As Mario Bunge observed, "all factual sciences, whether natural, social, or mixed, share a number of philosophical concepts…and a number of philosophical principles" (Bunge, 2001). Moreover, philosophers "won't remain satisfied with examples [...]; they will seek general patterns" (Bunge, 2004). The work of such outstanding systems thinkers as Mario Bunge and F.A. Hayek includes clear and concise definitions and descriptions of such concepts and of some fundamental patterns which are essentially the same as those formulated—independently—in the best IT-based sources. Specifically, the concept of a *relation* is indispensable for understanding a system:" so long as the elements... are capable of acting upon each other in the manner determining the structure of the machine, their other properties are irrelevant for our understanding of the machine" (Hayek, 1952), and "the structure of a system is the set of all the relations among its components, particularly those that hold the system together" (Bunge, 2003). The same kinds of generic relations hold together very different systems, thus providing a solid foundation for bridging the communication gap between business and IT experts.

E.W. Dijkstra observed decades ago (Dijkstra, 2007) that "it is the sole purpose of the specifications to act as the interface between the system's users and the system's builders. The task of "making a thing satisfying our needs" as a single responsibility is split into two parts– "stating the properties of a thing, by virtue of which it would satisfy our needs" and "making a thing guaranteed to have the stated properties." These considerations apply both to the "business side" and to the "IT side" of any application development (or purchase) project.

Some examples of simple and elegant generic specifications understandable to their readers include the relational data model (Codd, 1970), certain high-level programming languages (Wirth, 1995), and the Macintosh user interface. The underlying implementations may have been complex,

but it was of no importance to the users of these systems: the "internals" were not exposed. However, in too many instances businesses have been unnecessarily restricted by "requirements" imposed by inadequate IT systems. As a well-known example, consider restrictions imposed on data by IT systems that require a set of predefined manual codes (without any business meaning) and prohibit the use of business-specific aliases. These restrictions—a typical example of business-IT misalignment—still result in serious losses for businesses, although excellent IT-based solutions to these problems were described more than 30 years ago, for example, in Gilb and Weinberg (1977).

A SYSTEM OF REUSABLE ABSTRACT CONCEPTS

A business model ought to be abstract enough to be understood, and therefore, while being precise, it should not be excessively detailed. But it also *cannot* be too detailed: as observed by Hayek, in any complex system "of life, mind and society" it is possible to "determine only the general character of the resulting order and not its detail." The purpose of a (high-level) model of a complex domain is to "bring about an abstract order – a system of abstract relations – concrete manifestations of which will depend on a great variety of particular circumstances which no one can know in their entirety" (Hayek, 1985). Such a model is essential for strategic decision making; and because tactical and operational decision making are determined by strategic decisions, such a model becomes essential for any kind of business decision.

As early as 1605, Bacon noted that "amongst so many great foundations of colleges in Europe, I find strange that they are all dedicated to professions, and none left free to Arts and Sciences at large." This is what "systems thinking" is about, and it has been around, under different names, for millennia. It is based on mathematics and exact philosophy–areas of human endeavor that have also been around for millennia (see, for example, the eloquent presentation in (Russo, 2004)). Furthermore, if the business stakeholders did not state the requestes properties of the IT "thing" (also sometimes known as "business rules") for any reason–for example, they were never asked, "everyone knew" what these properties were supposed to be, or it was not known "how to ask"– then the developers make these properties up and often specified them only in their code (very often unreadable to anyone except–prehaps–the developers themselves), so that as a result the IT thing has an important but not too useful property "it does what it does".

William Kent presents (Kent, 1978) an approach in which asking apparently trivial questions (like "What is an information systems thing?" "What is a Real World thing?" "What does 'the same thing' mean?" "What is a name?" etc.) leads to substantial clarification of a business domain and of business problems. This is an essential component of the framework for business-IT alignment because understanding and articulation of questions and problems is much more important than answering or finding solutions: "deeper understanding of the real features of a problem ... is an essential prelude to its correct solution" (Johnstone, 1977). Understanding a problem is possible only when the business domain for that problem is demonstrably understood in the same manner by all stakeholders, in order for the (inevitably) different viewpoints of different stakeholders to be conceptually compatible. More specific questions asked about a business domain, such as "what is a bank?" and "what is a margin call?" proposed by Dines Bjørner (Bjørner, 2006), lead to understanding of a business domain and articulating that understanding. As one of many examples of failures due to inadequate understanding, consider *The Spectator*'s assessment (Vincent, 2007) of hedge fund modeling: incomplete models for which invariants were not made explicit and too much was left outside as tacit assumptions led to very expensive failures of quantitative hedge funds due to violations of these assumptions: "shares were moving in ways that hadn't been programmed into the computer models." No wonder that, as a result, *Financial Times* (Davies & Tett, 2007) described the "Flight to simplicity" and understandability requested by users or potential users of financial instruments who wanted to return to "old fashioned banking" sketched in the article in exactly the same manner as in Dunbar (1901); for a simple example, a credit card is conceptually the same as a letter of credit a century ago.

Clearly, using abstraction is a prerequisite for understanding and for separating the concerns of various business and IT stakeholders. But this is not sufficient: As observed by F.A. Hayek: "Until we have definite questions to ask we cannot employ our intellect; and questions presuppose that we have formed some provisional hypothesis or theory about the events" (Hayek, 1985). A domain model, or its fragment, is just such a provisional theory. If it does not exist yet for the specific domain of interest, we may try to use appropriate generic models that almost always do exist; and if no generic models exist, then we can try to compose such a model out of appropriate business patterns.

This concept of business patterns is not new at all. Adam Smith eloquently presented it about 250 years ago in *The Theory of Moral Sentiments*:

When a number of drawings are made after one pattern, though they may all miss it in some respects, yet they will all resemble it more than they resemble one another; the general character of the pattern will run through them all; the most singular and odd will be those which are most wide of it; and though very few will copy it exactly, yet the most accurate delineations will bear a greater resemblance to the most careless, than the careless ones will bear to one another.