Conceptual Modeling Method for Separation of Concerns and Integration of Structure and Behavior

Remigijus Gustas, Karlstad University, Sweden
Prima Gustienė, Karlstad University, Sweden

ABSTRACT

Identification of discontinuities, separation of concerns, and dealing with the evolutionary changes of requirements is difficult in conceptual modeling. The limited human mind allows focusing on one particular requirement at a time in isolation. One fundamental problem is that all conventional conceptual modeling techniques deal with collections of loosely linked meta-models, which are defined by different types of diagrams. Typically, system development methods project interactive, behavioral, and structural aspects of information systems' conceptual representations into disparate views. Therefore, the semantic integrity of various architecture dimensions is difficult to achieve. The difficulties stem from the paradigmatic mismatch between static and dynamic constructs. The advantage of the conceptual modeling approach presented in this paper is flexibility. It is demonstrated by case study examples that sequential, underlying, enclosing, overriding, and overlaying interaction loops between actors provide the foundation for the composition of complex scenarios, which span across organizational and technical system boundaries. The presented semantic integration and system decomposition principles target business process modeling experts and information system designers, because they are essential for introducing evolutionary changes and managing complexity of information system conceptualizations.

Keywords: Conceptual Modeling, Interaction Dependency, Overlays, Semantic Integration, System Decomposition Principles, Underlying and Enclosing Loops

INTRODUCTION

Conventional information system (IS) analysis and design methods are restricted in their ability to distinguish among crosscutting concerns, which span across various types of diagrams.

DOI: 10.4018/jismd.2012010103

It does not matter whether designers apply structured analysis and design (SAD) methods (Gane & Sarson, 1979; Yourdon & Constantine, 1979), object-oriented (Blaha & Rumbaugh, 2005) or component based methods: their expressive power is limited in separating various concerns. Poor concern separation technique is one of the reasons why the way systems are
Currently built is rather primitive. Consequently, managing the complexity of specifications in software engineering is a problem that can be certainly attributed to various limitations of traditional IS modeling and design methods. The following software engineering issues have remained problematic over the last three decades (Jones, 2009):

- There are more defects in requirements and design than in source code,
- Initial requirements are seldom more than 50% complete,
- There are more defects in test cases than in software itself,
- About 7% of all defect repairs will accidentally inject new defects,
- About 5% of software outsource contracts end up in litigation.

In traditional areas of engineering, developers are able to present their design decisions by using a finalized computation-neutral representation. This is not the case in the area of system engineering. The limitations of conventional system modeling methods result in two side effects, which in aspect-oriented software development (Jacobson & Ng, 2005) are known as tangling and scattering. Tangling occurs when the software component or class, instead of fulfilling a particular concern, encapsulates a diverse set of concerns. If a particular concern is spread across multiple components, then this situation is called scattering. When the requirements implied by that concern are modified, the designer must identify all related components and find out how these components are affected by introduced changes. Especially, modifying requirements, which are related to a big number of diagrams, becomes quite problematic. Poor understanding of concerns makes it difficult to make even simple evolutionary extensions of IS specifications. Separation of crosscutting concerns is the first fundamental problem, which cannot be solved without modifying the modeling foundation in system analysis and design. In this paper, we will introduce system decomposition principles, which suggest a new and more natural way of managing complexity in system engineering.

The declarative nature of value flow exchanges help technical system designers to analyze underlying business events, which are quite comprehensible for stakeholders such as business process modeling experts, enterprise architects, and users. Diagnosing value flows among different organizational components in IS engineering is important for solving the alignment problem (Wieringa, 2008; Wieringa & Gordijn, 2005) of value models (Gordijn & Akkermans, 2000) with the behavioral effects and structural changes in various classes of objects. Value exchanges and related coordinating events can be used as guidance for designers to move smoothly from system analysis to design, without a requirement to represent a complete solution. By sending and receiving value flows, the actors enter into commitments regarding their privileges, rights, responsibilities and obligations. One of the reasons why conventional system analysis and design methods are not suitable for modeling the deontic aspects of organizations (Wagner, 2003), such as commitments and claims (Chopra et al., 2010), is that they are unable to capture value and coordinating flows among organizational subsystems. Interaction-based thinking has proved to be fruitful in the area of enterprise engineering (Dietz, 2006). However, there is a paradigmatic mismatch between the traditional object-based IS modeling methods and system analysis approaches, which are based on the modeling of interactions between actors. The paradigmatic differences are an obstacle in finding an elegant solution for the alignment problem of business process design with IT operations (Wieringa, 2008). So, the second fundamental problem is unclear principles of blending between the traditional system and enterprise engineering methods. Our intention is to introduce a semantically integrated modeling method, which allows combining interaction dependencies between actors with the conventional semantic relations in the area of system analysis and design.

A limited human mind is capable of focusing on a particular concern at a time in isolation without paying too much attention to
Stabilization of Large-Scale Fuzzy Interconnected System
(2017). Large-Scale Fuzzy Interconnected Control Systems Design and Analysis (pp. 34-83).
www.igi-global.com/chapter/stabilization-of-large-scale-fuzzy-interconnected-system/181988?camid=4v1a

Fast Data Processing for Large-Scale SOA and Event-Based Systems
Marcel Tilly and Stephan Reiff-Marganiec (2015). International Journal of Systems and Service-Oriented Engineering (pp. 54-77).
www.igi-global.com/article/fast-data-processing-for-large-scale-soa-and-event-based-systems/137070?camid=4v1a