Modeling Approach for Integration and Evolution of Information System Conceptualizations

Remigijus Gustas, Karlstad University, Sweden

ABSTRACT

Most information systems development methodologies are based on conceptual modeling of static and dynamic views, which are represented by totally different types of diagrams. Understanding of the interplay among interactive, behavioral and structural aspects of specifications is necessary for identification of semantic integrity problems between business process and business data. Typically, semantic inconsistencies and discontinuities between collections of conceptual representations are not easy to detect and to comprehend for information system designers due to static and dynamic aspects of models being visualized in isolation. The goal of this paper is to present a modeling approach for semantic integration and evolution of static and dynamic aspects of conceptual models. Visualization of interplay among structural, interactive and behavioral aspects of computation-neutral representations helps to understand crosscutting concerns and integrity problems of information system conceptualizations. The main advantage of the presented conceptual modeling approach is stability and flexibility of diagrams in dealing with the evolutionary changes of requirements. Therefore, the developed modeling foundation is targeted to both business managers and information system designers for the purpose of computation-neutral integration and evolution of information systems specifications.

Keywords: Behavior, Evolution, Interaction, Semantic Integration of Static and Dynamic Aspects, Structural Changes of Objects

INTRODUCTION

Information system architectures can be defined on various granularity levels and expressed by modeling views, which are normally represented by using different types of diagrams. It is common to the industrial versions of system analysis and design methods to distinguish disparate views and dimensions of enterprise architecture (Zachman, 1987). The Zachman Framework can be viewed as taxonomy for understanding of different types of diagrams. It defines various dimensions of business application and data architectures such as What, How, Who, Where, When and Why. Integrity of different architecture views and dimensions is a fundamental problem in most Information System (IS) methodologies. Semiformal IS
modeling methods such as structured analysis and design (Gane & Sarson, 1979; DeMarco, 1979; Yourdon & Constantine, 1979) are plagued by the paradigm mismatch between static and dynamic types of diagrams, which represent business processes and business data in isolation.

Object-oriented design (Booch et al., 1999) languages were introduced to overcome the integrity problems of static and dynamic aspects. Unified Modeling Language (UML) (OMG, 2010) provides various types of diagrams, which are applied for representation of interactive, behavioral and structural aspects of information system specifications. Every modeling approach, which covers a collection of different diagrams, must contain a systematic method for detection of inter-model inconsistency and incompleteness. The static and dynamic aspects of IS specifications are complimentary and they cannot be analyzed in isolation. The structural aspects describe characteristics of objects, which are invariant in time. The interactive and behavioral aspects describe dynamic characteristics of objects over time. All object-oriented design languages are based on collections of meta-models (Glinz, 2000). There is often semantic discontinuity and overlapping in IS specifications, because static and dynamic constructs of meta-models and notations do not fit perfectly. Semantic integrity of diagrams is difficult to achieve in this situation. This problem was addressed in the development of the ArchiMate language, which can be used for bridging from business to Information Technology (IT) layers of enterprise system architecture (Jonkers et al., 2004). A number of ontologically grounded modeling rules proposed by Evermann and Wand (2009) are necessary to achieve integrity among UML diagrams. However, enforcing these rules using such de facto standard as UML is still quite problematic. Principles of interplay among UML constructs, which define semantics of business processes and business data, are not completely clear.

Working with the collections of meta-models is more cumbersome for the achievement of semantic modeling quality (Lindland et al., 1994). Semantic integrity problems can be measured in terms of semantic inconsistency and incompleteness between specifications on various levels of abstraction. Inter-model consistency is hard to achieve for non-integrated model collections. Modeling techniques that are realized as collection of models are more difficult to comprehend for system analysis experts who are responsible for business alignment with implementation-oriented architectural descriptions of enterprise systems. UML was developed with the ultimate goal of unifying the best features of the graphical modeling languages and creating a de facto industry standard for object-oriented software design. Recently, UML started to evolve into a language for business and enterprise modeling. However, the decomposition principles and principles of separation of crosscutting concerns (Jacobson & Ng, 2005) are ambiguous in UML. This situation contributes to a more complicated process for introducing and managing evolutionary changes of information system conceptualizations. Semantic integration principles of interactive, structural and behavioral aspects of IS specifications are not completely clear in UML. The treatment of these deficiencies would require modification of the UML foundation. Introducing fundamental changes in UML, with the purpose of integration of collection of models, is a complex research activity. However, such attempts would allow using UML as a computation-neural conceptual modeling language. It is necessary for understanding and reasoning about enterprise system architectures (Finkelstein, 2004) across organizational and IT system boundaries. It is recognized that UML support for such task is vague, because semantic integration principles of different diagram types are lacking (Harel & Rumpe, 2004).

Understanding the needs of an enterprise organizational system in the early analysis stages of a system development projects requires knowledge about stakeholders, their goals, interactions, and alternative actions (Horkoff & Yu, 2010). i* models are intended
Related Content

An Experimental Comparison of System Diagrams and Textual Use Cases for the Identification of Safety Hazards

Measurement-Based Timing Analysis for Reconfigurable Embedded Systems
[www.igi-global.com/chapter/measurement-based-timing-analysis-reconfigurable/50427?camid=4v1a](www.igi-global.com/chapter/measurement-based-timing-analysis-reconfigurable/50427?camid=4v1a)
Threat Modeling in Agile Software Development

Katana: Towards Patching as a Runtime Part of the Compiler-Linker-Loader Toolchain