A Rigorous Approach for Mapping Workflows to Object-Oriented IS Models

Stef Joosten, Open University, The Netherlands & Ordina-Anazagoras, The Netherlands
Sandeep Purao, Georgia State University, USA

ABSTRACT

Conceptual business models and conceptual models of information systems represent widely different perspectives, which are difficult to integrate due to differences in structure and meaning that exist across these two viewpoints. Such problems in integration can be difficult for an information architect to predict. Yet, the interoperability between the two is essential to ensure efficacy of the information system that will be built. Our research provides an approach to integrate a specific business modeling vehicle, based on workflows, against a specific IS implementation vehicle, based on object-oriented programming. The approach takes into account semantic constraints to ensure that the two perspectives can be integrated, thus moving the integration beyond the level of structures. Using a deductive argument form, we apply semantic constraints to the core concepts of in workflow management, mapped onto those in object-oriented programming languages. The mapping builds on ontological models of both domains—treating the concepts in each as things in the world—and integrates both worlds into a single ontology enriched with semantic constraints. The key innovation of this work, therefore, lies in reasoning about the semantics of conceptual mappings by standard logical means. The results demonstrate a practical approach to map semantic constraints across domains with example derivations. The research, thus, extends existing work closer towards practical application in several contexts including projects involving embedded workflow. The approach has been successfully used in several large real-world projects to reduce design efforts by increasing the consistency, simplifying the overall structure, and making the enterprise architecture more concrete.

Keywords: object-oriented programming, IS implementation, workflow management

INTRODUCTION

Systems development involves construction of the anticipated or planned conceptual business model, and projecting from it the explicit conceptual model of the information system to be built. The two perspectives—business models and information systems models—are, thus, closely related, and the efficacy of the information system can often depend upon ensuring inter-operability between the two. The two models, however, represent widely different perspectives. The incompatibility between the two perspectives can be attributed to differences in meaning and structure. Differences in meaning refer to the existence and relevance of different concepts in different domains. For
example, the concepts ‘actor’ and ‘activity’ may be important in the business domain, whereas ‘object’ and ‘message’ are some concepts relevant in the IS domain. Differences in structure refer to different constraints, which ensure creation of valid and internally consistent populations based on these concepts. For example, a constraint in the business domain may state that an ‘activity’ may be performed only if it has an ‘actor’ associated with it. A constraint in the IS domain may state that an ‘object’ may have access to the attributes and methods of another, if its ‘class’ inherits from the class of the other. Mapping these two perspectives, thus, is a difficult problem.

The objective of this paper is to develop an approach for mapping a conceptual business modeling vehicle, based on workflows, against an IS modeling vehicle, based on object-oriented programming. Workflows refer to coordination of activities, at the organizational and technical level, for forming and managing business processes. The fundamental concepts in this area are beginning to acquire clarity of definition with empirical investigations and theoretical work (van der Aalst, 2001; Casati et al., 1996; Ellis and Nutt, 1993; Joosten et al., 1994; Joosten, 1996; Kiepuszewski et al., 2000). Coordination, which is the basis for workflows (Malone and Crowston, 1991), has received considerable attention as the basis for organizational activity (Lipnack and Stamps, 1997, Nelson et al., 2000).

On the other hand, object-oriented programming requires the programmer to think about the world in terms of objects and messages (Cox 1984, Entsmonger 1990), and allows creation or derivation of new types (Entsminger 1990). Though a standard, accepted underlying model for object-oriented programming remains elusive, the essential elements of object-oriented programming are well-settled (Henderson-Sellers and Edwards, 1994, Arnow and Weiss, 1998; Fontoura et al., 2000; Nelson et al., 2000a).

We use a deductive approach, based on ontological representation of the two domains for this purpose. The approach involves multiple intermediate steps. First, we discover and represent core concepts of object-oriented programming (Nierstrasz, 1986; Takagaki and Wand 1991) and workflows (Casati et al., 1996, Joosten, 1996). Next, we add semantic constraints to these core concepts to complete the ontology. We then map the two ontologies onto each other by making appropriate choices. Finally, we derive consequences of the mapping with a view to establishing feasibility of our approach, and demonstrating some interesting results.

A related research stream that addresses problems similar to ours is methodology engineering (Kumar and Welke, 1992; Kelly, Lyytinen and Rossi, 1996; Dahanayake, Sol and Dietz, 1997). Our problem, however, differs from that stream, which involves mapping of different methods for building IS models, requiring discovery and exploitation of shared viewpoints and purpose. Our problem, on the other hand, seeks to map information from a conceptual business modeling vehicle to an information system modeling vehicle. These two often share neither a common perspective, nor a common purpose, providing little direction for reasonable solutions.

The remainder of this paper is structured in five sections. The next section outlines and explains an approach for representing the ontologies, and presents the notations used for this purpose. Then, we develop the specific formal ontologies, for object-oriented programming and
Fault-Tolerant Quorum Consensus Scheme for Replication Control in Mobile Distributed Database Systems: FTQC
www.igi-global.com/article/fault-tolerant-quorum-consensus-scheme/51200?camid=4v1a