ABSTRACT

The problem of identifying relevant classes (entities) and associations (relationships) is a fundamental problem for conceptual modeling. In previous work the authors introduced a conceptual modeling methodology named OMP (Ontology-based Modeling Patterns), which is based on the analysis of class categories representing entity types that are organized in the form of ontology. Since then the authors have explored a way to improve the methodology. As a result, in this paper the authors introduce a new conceptual modeling framework, entitled CAM (Class/Association-analysis-based Modeling), which is based on the analysis and classification of association types as well as entity types. The main objective of CAM is to serve as a tool to facilitate teaching the fundamentals of conceptual modeling to students in a systematic way, by providing extensible and adaptable entity/association classificatory systems that can be directly used in the problem-solving process. In this paper the authors present the CAM framework and illustrate its application.

Keywords: Class Modeling, Conceptual Modeling, Entity Classification, Ontology, Relationship Analysis

INTRODUCTION

Conceptual modeling is defined as constructing a conceptual representation of the application domain of an information system (Wand & Weber, 2002; Weber, 2003). Since the activity constitutes one of the initial steps in information systems analysis and design, identifying classes and associations that accurately capture relevant phenomena in the given domain is fundamental for conceptual modeling (Parsons & Wand, 1997; Song et al., 2004).

In this paper we introduce a new conceptual modeling framework, entitled Class/Association-analysis-based Modeling (CAM). The two main components of the framework consist of analysis/classification of entity types and analysis/classification of association types. The third component consists of classification of modeling patterns. The CAM framework, as such, is supported by the top-down, overarching analysis/classification of entity classes and association types, and it supports the bottom-up, modular construction of class diagrams using modeling patterns. Except the fact that it also includes the analysis/classification of entity
types, the CAM framework is distinct from a framework we previously introduced, i.e., the Ontology-based Modeling Patterns (OMP) (Athenikos & Song, 2008).

The main objective of CAM is to serve as a tool for teaching conceptual modeling to novice students in a systematic manner, rather than to provide a rigorous ontological analysis/system of classes and associations or an exhaustive collection of enterprise-level conceptual models. We thus do not argue for the completeness of the classification system. Our aim in this paper is to describe the CAM framework as well as to partially demonstrate its application.

**CAM (CLASS/ASSOCIATION-ANALYSIS-BASED MODELING)**

Here we describe the framework of Class/Association-analysis-based Modeling (CAM). (While the framework is intended to be generalizable across domains to a certain extent, its current content is geared towards the business domain, with a focus on the direct sales transactional aspect.) The CAM framework is based on three main components: (i) analysis/classification of entity types, (ii) analysis/classification of association types, and (iii) construction/classification of modeling patterns based on (i) and (ii). Below we describe each component in turn.

**Entity Classes**

We streamlined the taxonomy of entity classes from the OMP framework in order to make the organization of classes more easily understandable and usable by students. The resultant taxonomy in CAM consists of only 3 levels of subsumption (vs. 7 levels in OMP). Table 1 shows the simplified taxonomy of entity classes in CAM.

**Association Types**

The analysis/classification of association types in CAM was motivated by the examination of Coad et al.’s (1997) object models, Nicola et al.’s (2002) streamlined object models, as well as the modeling patterns in the OMP framework. The aim was to identify a set of most common association types that capture relevant activities in the business domain that need to be kept track of, in other words, those that satisfy the “need-to-know” requirements.

As a result of the analysis, a total of 28 association types were identified, which were organized into three groups: Primitive (see Table 2), Non-Primitive Single-Context (see Table 3), and Non-Primitive Multiple-Context (see Table 4). The primitive association types refer to those concerned with the familiar generalization/specialization hierarchy and various types of aggregation, which can be explicitly represented via the modeling primitives of UML. The non-primitive single-context association types refer to those that apply only in specific contexts, for example, between an entity class under the category Agent and another entity class under the category Non-Agent. The non-primitive multiple-context association types refer to those that can apply in multiple contexts, for example, between a Non-Agent entity and an Agent entity and also between a Non-Agent entity and a Non-Agent entity.

Table 2 shows 7 primitive association types. Table 3 shows 14 non-primitive single-context association types. Note that each association type represents (the predicate part of) a semantic triple, i.e., (Subject)—<Predicate>—(Object), and that each row in Table 3 corresponds to the Subject part while each column corresponds to the Object part. For example, the Affiliation association type holds between (a class belonging to the category) Agent, as Subject, and (a class belonging to the category) Non-Agent, as Object. Table 4 shows 7 non-primitive multiple-context association types. Note that, because these association types are those that have multiple contexts of application, each of these association types appears more than once in the table.

Table 5 specifies the meaning of each association type and the common entity classes involved with each association type. (Note that, for each association type, the numbered
Customizable Flexible Querying in Classical Relational Databases

www.igi-global.com/chapter/customizable-flexible-querying-classical-relational/20354?camid=4v1a

Evaluation of Quantified Statements Using Gradual Numbers

www.igi-global.com/chapter/evaluation-quantified-statements-using-gradual/20356?camid=4v1a