ABSTRACT

The tremendous demand for software productivity has led to the idea of reuse of solutions that have worked successfully in the past. The notion of a design pattern is now well accepted in software design, and research in the area of data modeling has also begun. Although two books have explicitly attempted to cover this area, the representations provided in the books seem to be focused on specific applications and do not provide a generic and comprehensive set of templates. Another book attempts to address the problem but provides patterns at a level of granularity too small to be useful. This paper teases out underlying structures that tend to occur frequently in these books and provides patterns at an abstract and more useful level of granularity. It describes 11 data modeling patterns commonly found in business scenarios. The patterns are then validated by checking the frequency of occurrence of each pattern in the data representations included in three comprehensive texts of reference models. Two of these sources are targeted mainly at practitioners, and the third is academic oriented and targeted at students learning data modeling. Results indicate that although certain patterns are used more frequently than others, most of the 11 structures occur with adequate frequency to qualify as patterns. A comparison reveals that the frequency distribution of patterns is different among these sources. Further, the academic-oriented source distinctly focuses on different patterns as compared to the other two sources. The paper discusses the differences and provides specific recommendations on improving pedagogy in conceptual data modeling.

Keywords: data classes; database conceptual design; entity diagrams; heuristic development; unified modeling language

INTRODUCTION

One thing expert designers know not to do is to solve every problem from first principles (Chi, Glaser & Farr, 1988; Gamma, Helm, Johnson & Vlissides, 1994). Tremendous demand for software productivity has led to the idea of reuse of solutions that have successfully worked in the past. Although the notion of a design pattern is now well accepted in software design, work in the area of conceptual data modeling has just started. The purpose of this paper is to foster this progress by providing patterns for conceptual data modeling at a level of abstraction and granularity that encourage pattern recognition and reuse. The patterns are subjected to valida-
tion by determining the frequency of their occurrence in data models found in comprehensive texts of reference models.

Gamma et al. (1994), whose work on patterns pertains to object modeling in software design, have defined a design pattern as a description of communicating objects and classes that are customized to solve a general design problem in a particular context. In the context of data modeling, a pattern can be defined as a description of objects (or entities), relationships, and attributes that are customized to solve a general conceptual or logical database design problem. Gamma et al. specify four essential elements of a pattern—Pattern name, Problem, Solution, and Consequences. The pattern name is a handle to describe a design problem, its solution, and consequences in a few words. The problem describes when to apply the pattern. In a data-modeling context, the solution would describe the conceptual and/or logical design of the pattern. The consequences are the results and trade-offs of applying the pattern. In this paper, we propose variation as a fifth element of a pattern. The variation of a pattern describes a limited deviation to the basic structure of the pattern. A variation of a pattern must bear resemblance to both the problem and the prescribed solution for the normative form of the pattern. Naming the variation provides us with a richer vocabulary of templates and alternatives within a given scenario.

Pattern recognition is a well-researched topic in cognitive psychology. Almost three decades ago, Reynolds and Flagg (1977) devoted a complete chapter to pattern recognition and discussed research in the area of speech and visual recognition. They state:

*Template-matching is the simplest process by which pattern recognition can occur.*

According to this theory a large number of internal representations, or templates, of various objects are stored within the permanent, or long-term memory. Associated with each representation is a label or meaning. When an external stimulus is presented it is compared to the various internal templates until a match is found. (p. 61)

The main premise in our paper is that a limited number of basic templates can cover a large number of data modeling scenarios.

The theoretical basis of data modeling patterns can also be expressed using three notions: production/schema, analogy, and abstraction. Production system theories are based on the claim that underlying human cognition is a set of condition-action pairs called productions (Anderson, 1996). The conditions specify some data patterns, and if elements matching these patterns are in memory, then the productions can apply. Analogy is a common problem-solving approach. With this method, the problem solver attempts to use the structure of the solution of one problem to guide solutions to other problems (Anderson, 1985; Metsker, 2002). For example, a designer may note that an order form and the invoice generated have a similar structure, and the data model of the order can be employed to model invoice by using a “transfer in problem solving” (Kotovsky & Fallside, 1989) approach. This is feasible because order and invoice are examples of a deeper structure called transaction. Abstraction facilitates defining the deeper structure of a pattern (Greeno, 1989). Any template structure is based on the notion of abstraction. Abstraction achieves parsimony in the number of structures to be recalled but relies on designer skill to accurately map an application description into a deeper structure. This mapping is depen-
A Deterministic Approach to XML Query Processing with Efficient Support for Pure and Negated Containments
www.igi-global.com/chapter/deterministic-approach-xml-query-processing/8020?camid=4v1a

Database Engineering Focusing on Modern Dynamism Crises
www.igi-global.com/chapter/database-engineering-focusing-modern-dynamism/11136?camid=4v1a