Chapter VII

Objectification of Relationships

Terry Halpin, Neumont University, USA

ABSTRACT

Some popular information-modeling approaches allow instances of relationships or associations to be treated as entities in their own right. Object-role modeling (ORM) calls this process “objectification” or “nesting.” In the unified modeling language (UML), this modeling technique is called “reification,” and is mediated by means of association classes. While this modeling option is rarely supported by industrial versions of entity-relationship modeling (ER), it is allowed in several academic versions of ER. Objectification is related to the linguistic activity of nominalization, of which two flavors may be distinguished: situational and propositional. In practice, objectification needs to be used judiciously, as its misuse can lead to implementation anomalies, and those modeling approaches that permit objectification often provide incomplete or flawed support for it. This chapter provides an in-depth analysis of objectification, shedding new light on its fundamental nature, and providing practical guidelines on using objectification to model information systems. Because of its richer semantics, the main graphic notation used is that of ORM 2 (the latest generation of ORM); however, the main ideas are relevant to UML and ER as well.
INTRODUCTION

In this chapter, the terms “relationship type,” “association,” and “fact type” all denote relation types that may be identified by typed predicates. For example, “Person plays Sport” and “Sport is played by Person” are alternative readings for the same fact type. In many business domains, it is perfectly natural to think of certain relationship instances as objects about which we wish to talk; for example, Australia’s playing of cricket is rated world class. In object-role modeling (ORM) dialects, this process of making an object out of a relationship is called “objectification” or “nesting” (Bakema, Zwart, & van der Lek, 1994; De Troyer & Meersman, 1995; Halpin, 1998, 2001; ter Hofstede, Proper, & Weide, 1993). In the Unified Modeling Language (UML), this modeling technique is often called “reification,” and is mediated by means of association classes (OMG, 2003a, 2003b; Rumbaugh, Jacobson, & Booch, 1999). Although industrial versions of entity-relationship modeling (ER) typically do not support this modeling option (Halpin, 2001, Ch. 8; Halpin, 2004a), in principle they could be extended to do so, and some academic versions of ER do provide limited support for it (e.g., Batini, Ceri, & Navathe, 1992; Chen, 1976). As an example of partial support, some ER versions allow objectified relationships to have attributes but not to play in other relationships.

In practice, objectification needs to be used judiciously, as its misuse can lead to implementation anomalies, and those modeling approaches that do permit objectification often provide only incomplete or even flawed support for it. This chapter provides an in-depth analysis of the modeling activity of objectification, shedding new light on its fundamental nature, and providing practical guidelines on how to use the technique when modeling information systems. Because of its richer semantics, the main graphic notation used is that of ORM 2 (the latest generation of ORM), with some examples being recast in UML; however, the main ideas are also relevant to extended ER.

Objectification is closely related to the linguistic activity of nominalization. The next section distinguishes two kinds of nominalization (situational and propositional), and argues that objectification used to model information systems typically corresponds to situational nominalization. The section after that proposes an underlying theory for situational nominalization of binary and longer facts, based on equivalences and composite reference schemes. The subsequent section extends this treatment to unary facts and discusses other issues related to the objectification of unary relationships. Then, we consider what restrictions (if any) should be placed on uniqueness constraints over associations that are to be objectified, and propose a set of rules and heuristics to guide the modeler in making such choices. The subsequent section discusses what kind of modeling support is needed to cater to facts or business rules that involve propositional nominalization or communication acts. The conclusion summarizes the main results, suggests topics for future research, and lists references for further reading.

TWO KINDS OF NOMINALIZATION

In this chapter, we treat nominalization as the recasting of a declarative sentence using a noun phrase that is morphologically related to a corresponding verb in the original sentence. Declarative sentences may be nominalized in different ways. One common way is to use a gerund (verbal noun) derived from the original verb or verb
Related Content

Efficient Filtering of Branch Queries for High-Performance XML Data Services
www.igi-global.com/article/efficient-filtering-branch-queries-high/3404?camid=4v1a

Databases for Mobile Applications
www.igi-global.com/chapter/databases-mobile-applications/11140?camid=4v1a

Graph Mining Techniques: Focusing on Discriminating between Real and Synthetic Graphs
Ana Paula Appel, Christos Faloutsos and Caetano Traina Junior (2012). Graph Data Management: Techniques and Applications (pp. 239-259).
www.igi-global.com/chapter/graph-mining-techniques/58613?camid=4v1a
Database Pointers in Navigational and Object–Oriented Database Management Systems: A Comparison
www.igi-global.com/article/database-pointers-navigational-object-oriented/51155?camid=4v1a