Enhanced Verbalization of ORM Models

Matthew Curland\textsuperscript{1} and Terry Halpin\textsuperscript{2}

\textsuperscript{1} ORM Solutions, USA
\textsuperscript{2} INTH International University, Malaysia and LogicBlox, Australia
mcurland@live.com, terry.halpin@logicbox.com

Abstract. Fact-oriented modeling approaches such as Object-Role Modeling (ORM) validate their models with domain experts by verbalizing the models in natural language, and by populating the relevant fact types with concrete examples. This paper extends previous work on verbalization of ORM models in a number of ways. Firstly, it considers some ways to better ensure that generated verbalizations are unambiguous, including occasional use of lengthier verbalizations that are tied more closely to the underlying logical form. Secondly, it provides improved verbalization patterns for common types of ORM constraints, such as uniqueness and mandatory role constraints. Thirdly, it provides an algorithm for verbalizing external uniqueness and frequency constraints over roles projected from join paths of arbitrary complexity. The paper also includes some discussion of how such verbalization enhancements were recently implemented in the Natural ORM Architect (NORMA) tool.

1 Introduction

When designing an information system, it is important to ensure that the data model accurately reflects the data requirements for the relevant business domain by validating the model with someone who understands the business domain, i.e. a domain expert. Since domain experts are often nontechnical, the data model should be communicated to them in language that is intelligible to them, without requiring mastery of a complex, technical syntax such as that of the Object Constraint Language (OCL) [17]. Fact-oriented modeling approaches are specifically designed to facilitate communication between modelers and domain experts by verbalizing the data model in natural sentences and populating the model’s fact types with concrete examples of fact instances. A fact type corresponds to a set of typed predicates of arity one (e.g. Person smokes), two (e.g. Person was born on Date), or higher (e.g. Item contains Item in Quantity). This approach differs from Entity relationship (ER) modeling [3] and class diagramming within the Unified Modeling Language (UML) [19], which encode some facts in attributes (e.g. Person.isSmoker, Person.birthdate).

Fact-orientation’s attribute-free nature promotes semantic stability (e.g. no remodeling is needed to talk about an attribute) and its graphical constraint notation for data modeling is much richer than that of industrial ER or UML. The family of fact-oriented modeling approaches include various dialects such as Object-Role Modeling (ORM), Natural-language Information Analysis Method (NIAM) [20],