Structural Aspects of Data Modeling Languages

Terry Halpin

LogicBlox, Australia and INTI International University, Malaysia
e-mail: terry.halpin@logicblox.com

Abstract: A conceptual data model for an information system specifies the fact structures of interest as well as the constraints and derivation rules that apply to the business domain being modeled. The languages for specifying these models may be graphical or textual, and may be based upon approaches such as Entity Relationship modeling, class diagramming in the Unified Modeling Language, fact orientation (e.g. Object-Role Modeling), Semantic Web modeling (e.g. the Web Ontology Language), or deductive databases (e.g. datalog). Although sharing many aspects in common, these languages also differ in fundamental ways which impact not only how, but which, aspects of a business domain may be specified. This paper provides a logical analysis and critical comparison of how such modeling languages deal with three main structural aspects: the entity/value distinction; existential facts; and entity reference schemes. The analysis has practical implications for modeling within a specific language and for transforming between languages.

1 Introduction

A *conceptual data model* includes a conceptual schema (structure based on concepts that are intelligible to business users) as well as a population (set of instances that conform to the schema). A conceptual schema specifies the fact structures of interest as well as the business rules (constraints or derivation rules) that apply to the relevant business domain. Various languages are used by modelers to capture or query the data model. These languages may be graphical or textual.

In *attribute-based approaches* such as Entity Relationship modeling (*ER*) [2] and the class diagramming technique within the Unified Modeling Language (*UML*) [18]), facts may be instances of attributes (e.g. Person.isSmoker) or relationship/association types (e.g. Person drives Car). UML's Object Constraint Language (OCL) [19, 21] provides a textual means to express class diagrams as well as many additional rules.

In *fact-oriented modeling* approaches, such as Object-Role Modeling (*ORM*) [10], all facts are treated as instances of fact types, which are represented using typed, logical predicates (e.g. Person smokes, Person drives Car). Referential facts also involve existential quantification (e.g. some Country has CountryCode 'AU'). For a detailed coverage of ORM and comparisons with ER and UML see [13]. Overviews of fact-oriented modeling approaches, including history and research directions, may be found in [9, 11]. The Semantics of Business Vocabulary and Business Rules (SBVR) initiative [20] and the Object-Oriented Systems Modeling (OSM) approach [6] are also fact-based in their requirement for attribute-free constructs.