Chapter 10
Reference Scheme Modeling

Terry Halpin
INTI International University, Malaysia

ABSTRACT

In natural language, individual things are typically referenced by proper names or definite descriptions. Data modeling languages differ considerably in their support for such linguistic reference schemes. Understanding these differences is important both for modeling reference schemes within such languages and for transforming models from one language to another. This chapter provides a comparative review of reference scheme modeling within the Unified Modeling Language (version 2.5.1), the Barker dialect of entity relationship modeling, Object-Role Modeling (version 2), relational database modeling, the Web Ontology Language (version 2.0), and LogiQL (an extended form of datalog). The authors identify which kinds of reference schemes can be captured within these languages as well as those reference schemes that cannot be captured. The analysis covers simple reference schemes, compound reference schemes, disjunctive reference, and context-dependent reference schemes.

INTRODUCTION

In this article, the term "object" means any individual thing. If an object is currently in our view, we may refer to that object simply by ostension (pointing at the object). Whether or not an object is in view, we may refer to it by using a linguistic expression. This allows us to reference concrete objects from the past (e.g. Einstein), the present (e.g. this article), or the future (e.g. the next solar eclipse), as well as intangible objects (e.g. a specific course in logic).

An information system models a specific *universe of discourse* (UoD), also known as a business domain (a world about which users wish to discourse within the business). For example, one UoD might concern a company's product sales, while another UoD might deal with flight bookings. In natural language, linguistic expressions used to reference objects within a given UoD are typically *proper names* (e.g. "Donald Trump") or *definite descriptions* (e.g. "the president of the USA") (Allen, 1995).

DOI: 10.4018/978-1-5225-7271-8.ch010

In philosophy, many proposals exist regarding the precise nature of proper names (e.g. see http://plato.stanford.edu/entries/names/) and definite descriptions (e.g. see http://plato.stanford.edu/entries/descriptions/). One popular account treats proper names as rigid designators, where “A rigid designator designates the same object in all possible worlds in which that object exists and never designates anything else” (http://plato.stanford.edu/entries/rigid-designators/). The term “possible world” may be assigned different meanings. In this article, a *possible world is treated as a state of the UoD being modeled by an information system*, and proper names are treated as rigid identifiers within the UoD of interest. Definite descriptions are often characterized as non-rigid, since some of them may refer to different objects in different possible worlds. For example, if we take “the president of the USA” as shorthand for “the current president of the USA”, then uttering this expression in 2016 refers to Barack Obama, while uttering the same expression in 2018 refers to Donald Trump—a simple example of deixis where the denotation of a term depends on its context (in this case, the time of utterance). However, given our sense of possible world, some definite descriptions are rigid designators (within a given UoD). For example, if we restrict the UoD to our world history, the definite description “the 45th president of the USA” always refers to Donald Trump. Moreover, if we further restrict the UoD to the year 2018 then “the president of the USA” is a rigid designator within that UoD.

The information models discussed in this article use both proper names and definite descriptions for identification, so each usage refers to just one object within the given UoD. However, the same object may take on different preferred identifiers in different contexts. As noted by Guizzardi (2005, ch. 4), for the same identifier to apply to an object throughout its lifetime, the object must belong to a *rigid type*. We define a type to be rigid if and only if each instance of that type must belong to that type for its whole lifetime *in the business domain being modeled*. Consider a UoD in which a person is identified by a student number while at a given university and is later identified by an employee number while working for a given company. Here, the type Person is rigid, but the types Student and Employee are not. To model this situation, we use a global, rigid identifier based on Person (e.g. PersonNr) that always applies, then introduce Student and Employee as “role subtypes” of Person, along with their local identifiers (StudentNr and EmployeeNr) for recording facts specific to their context as a student or employee. Setting up a 1:1 correspondence between the global and local identifiers allows history to be maintained about persons who migrate from one role subtype to another. If instead the UoD records facts about persons only while they are students at a given university, then for our purposes of information modeling, the type Student is rigid, even though it is not rigid in the ontological sense (since a person may enter and leave studenthood throughout his/her actual lifetime). Hence information models of business domains can be well formed even if they are not proper ontologies. For further discussion of such cases and temporal aspects of subtyping including mutability see Halpin (2009).

Computerized systems model linguistic reference schemes, either directly or indirectly. However, there are major differences in the way that popular data modeling and semantic web languages support such reference schemes. This article provides a comparative review of how such reference schemes are supported in current versions of the following modeling languages: the Unified Modeling Language (UML) version 2.5.1 (Object Management Group, 2017), the Barker dialect of Entity Relationship modeling (Barker ER) (Barker, 1990), Object-Role Modeling (ORM) version 2 (Halpin, 2005; Halpin, 2015b), relational database (RDB) modeling, the Web Ontology Language (OWL) version 2 (W3C, 2017), and LogiQL, an extended form of datalog (Halpin & Rugaber, 2014). Understanding the significant differences in the way these languages support reference schemes is important for modeling identification schemes