Modeling of Linguistic Reference Schemes

Terry Halpin, INTI International University, Nilai, Malaysia

ABSTRACT

When using natural language, people typically refer to individual things by using proper names or definite descriptions. Data modeling languages differ considerably in their support for such linguistic reference schemes. Understanding these differences is important for modeling reference schemes within such languages and for transforming models from one language to another. This article provides a comparative review of reference scheme modeling within the Unified Modeling Language (version 2.5), the Barker dialect of Entity Relationship modeling, Object-Role Modeling (version 2), relational database modeling, and the Web Ontology Language (version 2.0). The author identifies which kinds of reference schemes can be captured within these languages as well as those reference schemes that cannot be. The author's analysis covers simple reference schemes, compound reference schemes, disjunctive reference and context-dependent reference schemes.

Keywords: *Data Modeling, Entity Identification, Information Modeling, Integrity Constraints, Reference Schemes*

INTRODUCTION

In this article we use the term "object" to mean any individual thing. If an object is currently in a person's view, the person may refer to that object simply by ostension (pointing at the object). Whether or not an object is in view, one may refer to it by using a linguistic expression. This allows one 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 linguistics).

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

In philosophy, many different 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/).

DOI: 10.4018/IJISMD.2015100101

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 2003 refers to George W. Bush, while uttering the same expression in 2013 refers to Barack Obama—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 this world history, the definite description "the 44th president of the USA" always refers to Barack Obama. Moreover, if we further restrict the UoD to the year 2013 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, as discussed later in the section on context-dependent reference schemes, we allow the same object to 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, and 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). So information models of business domains can be well formed even though they are not proper ontologies. For further discussion of such cases and temporal aspects of subtyping including mutability see Halpin (2009).

Computerized systems use 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) (Object Management Group, 2013), the Barker dialect of Entity Relationship modeling (Barker ER) (Barker, 1990), Object-Role Modeling (ORM) (Halpin, 2005), relational database (RDB) modeling, and the Web Ontology Language (OWL) (W3C, 2012c). Understanding the significant differences in the way these languages support reference schemes is important for modeling identification schemes within such languages and for transforming models from one language to another. Of these languages, ORM provides the most comprehensive coverage of reference schemes at the conceptual level, so will be used as a basis for comparison when discussing support for reference in the other languages.

Fact-oriented modeling approaches such as ORM, Natural Language Information Analysis Method (NIAM) (Wintraecken, 1990) and Fully Communication Oriented Information Modeling (FCO-IM) (Bakema, Zwart, & van der Lek, 2000) differ from ER, RDB, and class modeling in UML by uniformly modeling atomic facts as unary or longer relationships that are instances of *fact*

Related Content

Trans_Proc: A Reconfigurable Processor to Implement The Linear Transformations

Atri Sanyaland Amitabha Sinha (2022). *International Journal of Software Innovation (pp. 1-16).*

www.irma-international.org/article/transproc/303575

Towards an Integrated Personal Software Process and Team Software Process Supporting Tool

Ho-Jin Choi, Sang-Hun Lee, Syed Ahsan Fahmi, Ahmad Ibrahim, Hyun-Il Shinand Young-Kyu Park (2012). *Software Process Improvement and Management: Approaches and Tools for Practical Development (pp. 205-223).*

www.irma-international.org/chapter/towards-integrated-personal-software-process/61216

Software Effort Estimation for Successful Software Application Development

Syed Mohsin Saif (2022). *Research Anthology on Agile Software, Software Development, and Testing (pp. 123-164).*

www.irma-international.org/chapter/software-effort-estimation-for-successful-software-application-development/294463

Autonomous Communication Model for Internet of Things

Sergio Ariel Salinas (2021). *Handbook of Research on Software Quality Innovation in Interactive Systems (pp. 252-266).*

www.irma-international.org/chapter/autonomous-communication-model-for-internet-of-things/273572

A Communication Model Based on Fractal Geometry for Internet of Things

Sergio Ariel Salinas (2021). *Handbook of Research on Software Quality Innovation in Interactive Systems (pp. 192-212).*

www.irma-international.org/chapter/a-communication-model-based-on-fractal-geometry-for-internet-of-things/273570