Guiding the Representation of n-ary Relations in Ontologies through Aggregation, Generalization and Participation

Paula Severi
Department of Computer Science, University of Leicester
ps56@mcs.le.ac.uk

José Fiadeiro
Department of Computer Science, University of Leicester
jose@mcs.le.ac.uk

David Ekserdjian
Department of History of Art and Film, University of Leicester
dpme1@leicester.ac.uk

ABSTRACT

We put forward a methodological approach aimed at guiding ontologists in choosing which relations to reify. Our proposal is based on the notions of aggregation, generalization and participation as used in conceptual modelling approaches for database design in order to represent situations that, normally, would require non-binary relations or complex integrity constraints.

Keywords

Semantic web , ontology engineering, n-ary relations, aggregation, reification

1. INTRODUCTION

We explain the method of Ontology Engineering that we introduced in [**?**, **?**] for simplifying ontologies. Our method is inspired by notions previously proposed in database modelling and construction for increasing the "understandibility of relational models by the imposition of additional semantic structure" [**?**]: aggregation and generalization [**?**] and dependencies between relations [**?**].

Although, in ontologies, the technical problems that arise are not necessarily the same as those of relational databases, the methodological issues are similar in the sense that the solution to our problem lies first of all in helping modellers to conceptualize the real world in a way that can lead to a better representation, and then offering them a mechanism for implementing these semantic structures in ontologies. By 'better' we mean a more controlled use of reification and a closer fit between the resulting ontology and the real-world domain as perceived by an expert.

Our method also makes ontologies easier to extend, in particular to reuse existing reifications when adding new relations to an ontology, which is essential for supporting an incremental process of ontology engineering.

2. MOTIVATION

A well-known limitation of OWL 2 (Web Ontology Language) is that only binary relations between classes can be represented [**?**, **?**, **?**]. In practice, relations of arbitrary arity are quite common and they have to be represented in OWL in an indirect way by coding them as classes[1]. In the literature on Description Logic (DL) [**?**], the class codifying a relation ρ is called the *reification of* ρ.

As any codification, reification requires extra work in addition to 'simple' modelling, which can make it quite impractical (and unintuitive), especially when performed by people who are not 'experts': extra classes, predicates, individuals and axioms [**?**] need to be introduced and, as the number of classes increases, ontologies can become very difficult to read and understand, mainly because this additional information often masks the concepts and structures that it encodes. That is, there is a mismatch between the layer of abstraction at which domain modellers work and that of the representation where information is encoded, which is particularly harmful when we want to extend and reuse ontologies. Ontologies that are simple and easy to read are also more likely to be reused.

3. OUR METHOD

In order to illustrate some of the problems that may arise from the limitations of having to encode n-ary relations through reification and the method that we propose to minimize them, we use the Ontology of Altarpieces [**?**] — a joint project between the Departments of Computer Science and History of Art and Film at the University of Leicester. This case study is a good example of a domain in which n-ary relations arise quite naturally and frequently.

Suppose that we want to express the following knowledge as produced in natural language by an art expert: *Joseph is holding the flowering staff in the altarpiece called "The Marriage of the Virgin" by Raphael*

The natural representation of this domain property is in terms of a relation *holds* of arity 3. Figure **??** shows an entity-relationship (ER) diagram for the relationship *holds*. The corresponding relation cannot be represented in OWL unless we code it as a class `Reifholds` of individuals that represent the tuples — the reification of the relation [**?**]. However, reifying *holds* is not necessarily the right decision

[1]Similarly for RDF (Resource Description Framework)

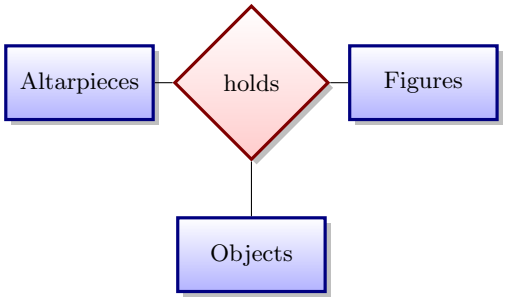

Figure 1: ER diagram: holds as a relationship of arity 3

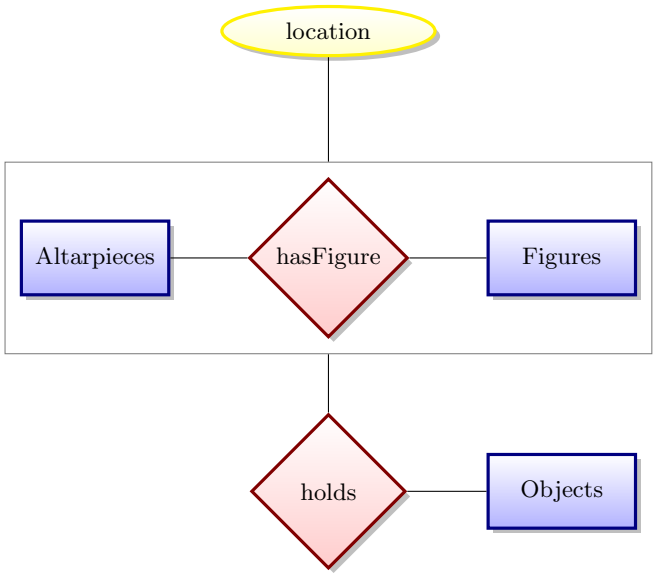

Figure 2: ER diagram: holds reduced to a binary relation through an aggregation.

that a modeller should make. Instead, we propose to transform the ternary relation *holds* into a binary relation $\widehat{holds}$, which does not need to be reified: it can be represented directly in OWL through a role. This simplification can be justified in methodological terms as it brings the representation closer to the domain.

Indeed, a conceptual model of the whole domain would reveal a richer semantic structure that is not captured in the simple diagram given in Figure **??**. More precisely, a wider conceptual model of the domain of altarpieces as depicted in Figure **??** shows that the entities *Altarpieces*, *Figures* and *Objects* are involved in more complex relationships. On the one hand, *holds* is actually a binary relationship between *Objects* and the 'aggregation' of a relationship *hasFigure* between *Altarpieces* and *Figures* (the aggregation is depicted by a box surrounding the relationship subdiagram, which is the notation usually adopted in conceptual modelling approaches). On the other hand, *hasFigure* has a 'descriptive attribute' (functional relationship) that returns the location of the figure in the altarpiece — one of `right`, `left`, `center`, `top`, `bottom`, `heaven` or `earth`.

In summary, the simplification discussed above corresponds, effectively, to reifying *hasFigure* and representing *holds* as a binary relation as depicted in Figure **??**. Our method explores, precisely, the fact that the reification of *hasFigure* can be taken to represent the aggregation of the relation as understood in conceptual modelling. It also explores the way other constructions (such as descriptive attributes) that have been proposed by the database community 30 years ago can be used for developing simpler and reusable ontologies from conceptual models.

To state the obvious, one should not take a blind approach to the representation of the domain and reify relations as they come: the complexity of the ontologies thus generated would be even beyond skilled computer scientists, let alone domain experts. As in database design, one should build a conceptual model of the domain before starting coding in OWL or any other language, and follow a sound methodology, as outlined in this paper, to generate or reuse code.

We are currently developing tool support for our methods in the form of Protégé plugins, one of which is being submitted as a demonstrator.