Computers in Industry 65 (2014) 1242–1252

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

An ontological approach for reliable data integration in the industrial domain

Stefano Borgo *

Laboratory for Applied Ontology, ISTC CNR, via alla cascata 56C, 38123 Trento, Italy

ARTICLE INFO

Article history:
Received 5 July 2013
Accepted 22 December 2013
Available online 26 February 2014

Keywords:
Formal ontology
Product data
Design data
Information integration

ABSTRACT

Ontologies are structural components of modern information systems. The taxonomy, the core of an ontology, is a delicate balance between adequacy considerations, minimal commitments and implementation concerns. However, ontological taxonomies can be quite restrictive and entities that are commonly used in production and services might not find room in a official or *de facto* standard or ontological system. This mismatch between the company's view and the ontological constraints can limit or even jeoparize the adoption of modern formal ontologies in industry. We study the roots of this problem and individuate a general set of principles to relate the ontology and those non-ontological entities that are yet important for the core business of the company. We then introduce a theoretically sound and formally robust approach to expand a given ontology with new dependency relations, which make available information regarding the non-ontological entities without affecting the consistency of the overall information system.

1. Introduction

Imagine you pick a book at a bookstore, read parts of it, find it interesting and finally buy the book. Any learned ontologist and knowledge engineer would distinguish at least two types of entities here: a physical entity (the physical-book that is bought at the bookstore and carried home in a bag) and an information entity (the information-book that that changes the knowledge or state of mind of the reader). The physical vs. information (or functional) distinction, blurred in natural language, is only one of the targets of ontological analysis and is frequently undetected in everyday life. Yet, it helps to answer questions like: what do we pay for when we by a stone sold as a paperweight? why do we recognize cars in a wreckage deposit?

Modeling domain knowledge in an ontologically consistent way is a fascinating enterprise and pushes the knowledge engineer to deal with a series of important distinctions that go often undetected in commonsense as well as in professional life. The goal is to model information as used in a domain application while avoiding possible sources of confusion and ambiguities. The problem we address in this paper is part of the large effort to improve today's organization and management of the information generated in engineering design and manufacturing and, more generally, in the production life cycle at large. More specifically, let us call *domain entities* the things a company talks about in its everyday business. It is well known that some of these entities may be ontologically unsound, the goal of this paper is to find a methodology that allows to recognize and process information regarding these domain entities even if not consistent with the adopted ontology. If we succeed, give the possibility to companies to embrace modern information systems, and their underlying ontologies, without giving up its own language and, more importantly, business perspective.

We started with the classical example of the book as a physical thing vs the book as an information object. This example is quite intuitive and has been analyzed at length in the literature. In linguistic semantics notions like *constructional polysemy* and *co-predication* [2] have been proposed to make sense of the double meaning of the term book. Semanticists analyze the use of these terms as an indication of the need for combinations of categories in different branches linguistic ontology. From this view, an expression like "That book is 500 pages long and is difficult to read" [2, p. 257] is a case of category overdetermination and special constructs are proposed to justify the existence of mixed categories whose elements collect all the needed properties. The idea is that an expression like "This car runs 120 mhp and is selling well" should be understood as talking about a new type of object which *combines* a physical object (with its physical properties), and a type (to which the statistics on the selling events on the market are referred). The distinction between physical and abstract or

* Tel.: +39 0461314873; fax: +39 0461314875.
E-mail address: stefano.borgo@cnr.it

0166-3615/$ – see front matter © 2014 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.compind.2013.12.010

between instance and type is essential but the proposal to introduce mixed categories is not acceptable in ontology since incompatible properties like ''being abstract'' (property of the book content) and ''being concrete'' (property of the physical book) cannot coexist in the same entity. This shows that the linguistic approach is not a solution for our case.

Since the mixture of incompatible properties often result in logical inconsistency, one can see if logicians have found a suitable way to deal with the physical vs. information (and analogous) distinctions. In formal logic, one aks whether the term 'book' is actually denoting something. Logic allows to choose between three options: the term does not denote, it denotes one entity or it denotes more than one entity [35, chapter 8]. We have already seen the problems raised by assuming it denotes just one thing. Instead, in *free logic* a term may have no denotation at all. Roughly speaking, this view says that the notion of book does not correspond to a real entity. Although formally correct, this is not an ontologically suitable solution since it is hard for a book producer to believe that what she calls book, her business object for which she makes models and production plans, is a non-existing entity. The third option, a term can have multiple denotations, is the view taken in paraconsistent logic [35, p. 160]. The knowledge engineer might be satisfied with multiple denotations since all domain entities, including books and cars, are recognized as meaningful under this perspective. Furthermore, even the ontologist would agree with this view since it allows to distinguish domain entities that are ontologically unsound: indeed *ontologically sound entities have a unique denotation while the others do not*. Unfortunately, the multiple denotation approach makes the relationship between domain entities cumbersome: in paraconsistent logic identity is not transitive and even basic logical principles, like *particular generalization*, fail. An information system based on this view might conclude that the same product *A*, which has label *HD-Id*32 for the handling department and *PD-Id*121 for the production department, may very well be two different things leading to unacceptable errors in counting how many products are in stock.

We have seen that natural language and logic propose unsatisfactory solutions to our problem and, to the best of our knowledge, no better solution comes from other domains. We believe this is due to the fact that the problem we are after is a truly ontological problem and we are now going to discuss it in these terms.

The paper is structured as follows. Section 2 introduces ontological taxonomies and explains why they are essentially tree-shaped. Section 3 clarifies the role of the ontology module in an information system and concludes that some entities are necessarily left out. Section 4 discusses the principles on which our approach relies. Section 5 briefly introduces two ontologies exploited in the industry domain and used later to provide examples. Section 6 presents our methodology divided in four phases and gives examples for each. Finally, Section 7 adds further observations and points to future work.

2. Taxonomies and criteria

An information system based on an ontology adopts, at the minimum, the taxonomic structure of the ontology, that is, the basic hierarchy of categories (aka concepts, types, classes)[1] and the criteria for category membership. The taxonomy impacts issues like the (types of) entities that can possibly exist, the properties they share, how entities and properties are related, and which relationships hold among entities and among properties.

Ontologists have exploited taxonomies in different formats. There are essentially two ontological structuring relations suitable to building articulated taxonomies, namely the relation of *part* and the relation of *subsumption* (or *ISA*). Other ontological relations like *participation* and *causation* do not partition the domain, or are non-iterative like the *instance-of* relation. Furthermore, relations that are linguistically (or even psychologically) motivated like *similar-to*, *opposite-of* and *kind-of*, have ambiguous interpretations and thus are ontologically unclear. Since parthood is actually a collection of different relations [40,45,27] that are not easy to keep apart, practically any ontological system of broad scope relies primarily on the subsumption relation. We will thus focus on these subsumption-based systems.

In principle, ontological taxonomies can use subsumption in different ways [44]. The taxonomies used in industrial applications are mostly *tree*-shaped, we will see an example of this in Section 5.2. In most of the other cases the requirement about disjointedness of subcategories is relaxed, which allows a category to be a specialization of two or more larger categories. This phenomenon is known as *multiple inheritance*. Technically, these taxonomies have the structure of *directed acyclic graphs* (DAG), we will see a case in Section 5.1.

The preference for tree-shaped taxonomies is grounded in a series of methodologies, e.g., OntoClean [25] and variants like [39,46], which analyze the ontological meta-properties of categories and guide a coherent subsumption implementation.

Ontological taxonomies implement a series of principles among which the following:

> First, a category system should strive to be *exhaustive*, providing a *complete* list of highest kinds so that there is a category for everything there might be. [. . .] The categories provided (at any given level) should be mutually exclusive, so that we avoid redundancy (and retain efficiency) and ensure that whatever there is can be uniquely located in exactly one category. [44, p. 9]

The first requirement for *structural adequacy*, known as the principle of *classification completeness* (''there is a category for everything there might be''), is particularly hard to achieve and has an important consequence: any entity not belonging to one of the categories in the chosen ontology, does not exist (relatively to that ontology). The second requirement, the principle of *ontology clearness*, amounts to say that ontological categories are disjoint.

Ontological taxonomies are also bound to more philosophical principles called ontological commitments. Two criteria are due to Quine [36,37]: *ontological minimality* and *ontological adequacy*. According to the first criterion the ontology should commit to the existence of a smallest number of categories. The selection of the categories to include in the ontology follows some *respectability* guidelines, where respectability is the result of ontological considerations; for instance, the possibility of entities that do not have clear identity criteria, is ontologically questionable and thus these kinds of entities should not be accepted. The minimality criterion has been considered by many authors although with different motivations as in the following snippet:

> An ontology should require the minimal ontological commitment sufficient to support the intended knowledge sharing activities. An ontology should make as few claims as possible about the world being modeled, allowing the parties committed to the ontology freedom to specialize and instantiate the ontology as needed. [22, p. 910]

[1] These terms are often associated with different meanings, which however vary across research communities. In ontology concepts are mostly understood as mental entities, types as abstractions from individuals, classes as extensions of properties, and categories as ontologically motivated classifiers. In this paper these distinctions are not crucial, thus we take these terms as synonyms.