Computer Standards & Interfaces 44 (2016) 28–41

Contents lists available at ScienceDirect

Computer Standards & Interfaces

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

On the importance of truly ontological distinctions for standardizations: A case study in the domain of telecommunications

Pedro Paulo F. Barcelos [a,*], Cássio C. Reginato [b], Maxwell E. Monteiro [c], Anilton S. Garcia [a]

[a] Electrical Engineering Department, Federal University of Espírito Santo—UFES, Av. Fernando Ferrari, 514, Goiabeiras, 29075-910 Vitória, ES, Brazil
[b] Computer Science Department, Federal University of Espírito Santo—UFES, Av. Fernando Ferrari, 514, Goiabeiras, 29075-910 Vitória, ES, Brazil
[c] Federal Institute of Espírito Santo – IFES, Rodovia ES-010, Km 6,5, Manguinhos, 29173-087 Serra, ES, Brazil

A R T I C L E I N F O

Article history:
Received 20 June 2014
Received in revised form 8 August 2015
Accepted 26 September 2015
Available online 3 October 2015

Keywords:
Standard
Formal specification
Ontology
OntoUML
ITU-T Recommendation G.805

A B S T R A C T

Standards are documents that aim to define norms and common understanding of a subject by a group of people. In order to accomplish this purpose, these documents must define its terms and concepts in a clear and unambiguous way. Standards can be written in two different ways: by informal specification (e.g. natural language) or formal specification (e.g. math-based languages or diagrammatic ones). Remarkable papers have already shown how well-founded ontology languages provide resources for the specification's author to better distinguish concepts and relations meanings, resulting in a better specification. This paper has the objective to expose the importance of truly ontological distinctions for standardizations. To achieve this objective, we evaluate a math-based formal specification, in Z notation, using a well-founded ontology language for a telecommunications case study, the ITU-T Recommendation G.805. The results confirm that truly ontological distinctions are essential for clear and unambiguous specifications.

1. Introduction

According to the Oxford Dictionaries[1], a **standard** is "*an idea or thing used as a measure, norm, or model in comparative evaluations*". That is, by means of comparative evaluations, a standard is something used by human beings to provide a unique or equal interpretation over something in order to interoperate, communicate or deal about this thing. Groups of people usually define standards in order to represent a community consensus. These standards are typically defined in informal specifications – like the ones in natural language (e.g. English or German) – or in formal specifications, which are the specifications that use mathematical-based notation (usually logic-based), in a diagrammatic form or not, to create descriptions in a more precise way.

As stated by Guizzardi [1], the suitability of a language to create specifications in a given domain depends on how "*close*" the structure of the specifications constructed using that language resemble the structure of the domain abstractions they are supposed to represent. Further, Guizzardi [1] also presents that the *structure of a language* can be accessed via the description of the specification of the *conceptual model underlying the language*, i.e., a description of the worldview embedded in the language's modeling primitives. In Milton & Kazmierczak [2], this is called the *ontological metamodel of the language*, or simply, *the ontology of the language*.

Natural languages do not have a well-defined *underlying conceptual model*, hence, these languages are notoriously ambiguous [3]. This happens because this category of languages evolved by cognitive and social demand through the centuries. The usage of natural languages in standardizations may lead to a document with a series of deficiencies, undermining its comprehension and use in interoperation, in decision-making, or in problem solutions. Fig. 1 presents the different types of ontological deficiencies that can occur in standards.

Ontological deficiencies can occur when the languages are built over a not-well specified *underlying conceptual model* (the language's metamodel). Apart from the natural languages, formal languages can (and usually do) suffer from such a problem, even when the language has a formalized *underlying conceptual model*. It must be made clear that the existence of ontological deficiencies in a language is not only related to the presence or absence of a formalization of the language's *underlying conceptual model*: it is a matter of "how well specified" this formalization is. A well-specified *underlying conceptual model* should rely on a sound well-founded ontology (sometimes called *upper ontology*), like the Unified Foundational Ontology (UFO) [1], the Bunge–Wand–Weber Ontology (BWW) [6], or the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [7]. For

* Corresponding author.
E-mail addresses: pedropaulofb@gmail.com (P.P.F. Barcelos), cassio.reginato@inf.ufes.br (C.C. Reginato), maxmonte@ifes.edu.br (M.E. Monteiro), anilton@inf.ufes.br (A.S. Garcia).

[1] http://www.oxforddictionaries.com/us/definition/american_english/standard?q=standard.

Recurrent abbreviations: AF—Adaptation Function, AP—Access Point, CP—Connection Point, ITU—International Telecommunication Union, LC—Link Connection, OCL—Object Constraint Language, OWL—Web Ontology Language, TCP—Termination Connection Point, TPF—Transport Processing Function, TTF—Trail Termination Function, UML—Unified Modeling Language.

http://dx.doi.org/10.1016/j.csi.2015.09.011
0920-5489/

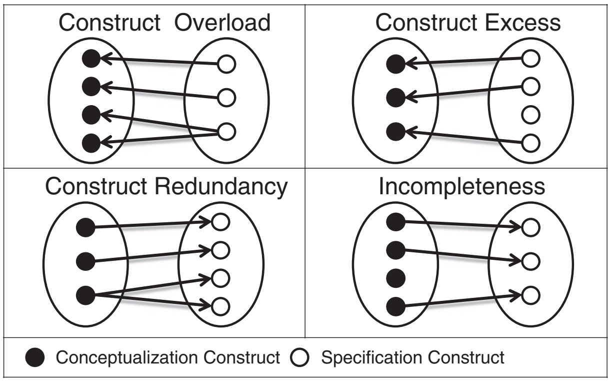

Fig. 1. Ontological deficiencies. From [4], based on [1,5].

instance, as results of ontological evaluations [1,5], deficiencies have already been identified in:

- diagrammatic languages, like the Unified Modeling Language (UML) [1,8];
- computational ontology representation languages, like the Web Ontology Language (OWL) [9]
- enterprise or business languages, like the Reference Model of Open Distributed Processing (RM-ODP) [10], or the Enterprise Systems Interoperability (ESI) [11];
- process modeling, like Petri Nets [12]; and
- objective modeling languages, like i* [13].

Nevertheless, some authors (e.g., Bowen [14] and Spivey [15]) claim that because a formal specification is precise (i.e., has a mathematical definition), this means that even if a certain specification is wrong, it is easier to identify and to correct the problem. The same authors claim that, since an informal specification is often ambiguous, it is more difficult to detect errors and subsequently put them right. Additionally, just like stated by Bowen [14], with the use of formal specifications, it is possible to reason about a system and detect inconsistencies in it far more easily than in the case where only an informal specification is available. The use of such languages allows the designer to verify if the system will behave as expected. In general, with the use of formal languages, the likelihood of errors in a design is reduced and errors may be pinpointed more easily [14].

Formal specifications may suffer from the ontological deficiencies presented in Fig. 1 especially when dealing with real-world domain modeling instead of design modeling. Making an analogy to the Model Driven Architecture [16], formal specification languages are more error-prone when dealing with Computational Independent Model (CIM, or the "analysis model") representation than when dealing with Platform Independent Model representation or Platform Specific Model representation. They are less error-prone because these two last models are an adaptation of the reality for the intended system – they are, respectively, the design model and the implementation itself – the CIM does not deal about systems, but about the system's real-world domain (e.g., a medical system is about diseases and treatments).

Using Guarino's classification of knowledge representation languages [17], math-based formal languages have primitives that can be categorized into the logical level (where primitives are propositions, predicates, logical functions, and operators) and into the epistemological level, which is the level of structure. Such categorization implies that these knowledge representation languages are neutral as concerns ontological choices. In fact, the ontological commitments of specifications that use math-based formal languages remain implicit, hidden in the mind of the specifications' authors [17], undermining comprehension and reuse. The ontological distinctions of real-world domain entities (e.g., rigidity, relational dependency) require a highly expressive language to be captured, thus, they are not captured by ontologically-neutral mathematical languages [1]. These expressive languages must be built with meta-properties that capture these ontological distinctions, and they must contain different constructs for different basic ontological categories [17]. Representation languages conforming to this view belong to Guarino's *ontological level*, which is the level of meaning [17]. Ontological level languages may commit to different ontological choices, resulting in different capabilities of identification and operation of ontological distinctions. However, the existence of such variation is interesting, as different domains may require different modeling approaches (e.g., a portion of a real-world domain may be represented statically or dynamically, depending on the modeling's objective and future application). In summary, especially in formalization of standards, the modeler must rely on an ontologically well-founded language instead of relying on a mathematical formal specification language. This happens because the latter does not provide mechanisms to deal with ontological issues in an appropriate way to represent complex domains.

According to Guizzardi et al. [9], the use of foundational concepts that take truly ontological issues seriously is becoming more and more accepted in the ontological engineering literature. In addition, the authors state that, in order to represent a complex domain, one should rely on engineering tools (e.g., design patterns), modeling languages, and methodologies that are based on well-founded ontological theories in the philosophical sense (see [18,19], for instance). Especially in complex domains – i.e., domains with complex concepts, relations, and constraints – and in domains with potentially serious risks of interoperability problems (the domain specified in the ITU-T Recommendation G.805 fits in both cases), a supporting ontology engineering approach should be able to:

a. allow the conceptual modelers and domain experts to be explicit, regarding their ontological commitments, which enables them to expose subtle distinctions between models to be integrated and to minimize the chances of running into a *False Agreement Problem* [20]
b. support the user in justifying their modeling choices and providing a sound design rationale for choosing how the elements in the universe of discourse should be modeled in terms of language elements [9].

This marks a contrast to practically all languages used to develop formal specification, including Z, B, Vienna Development Method (VDM), and Alloy. As stated by Guizzardi et al. [9], although these languages provide the modeler with mechanisms for building mathematical structures, they offer no support neither for helping the modeler on choosing a particular structure to model elements of the subject domain nor for justifying the choice of a particular structure over another. Finally, once a particular structure is represented, the ontological commitments that are made remain, in the best case, tacit in the modelers' mind. In the worst case, even the modelers and domain experts remain oblivious to these commitments [9].

An example of an ontologically well-founded modeling language is the version of UML 2.0 proposed in Guizzardi's doctoral thesis [1] and, thereafter, dubbed *OntoUML*. OntoUML real-world semantics is defined in terms of several ontological theories, such as theory of parts, of wholes, types and instantiation, identity, dependencies, unity, etc. However, in order to be as explicit as possible regarding all the underlying subtleties of these theories (e.g., modal issues, different modes of predication, higher-order predication), this language strives for having its formal semantics defined in a logical system as expressively as possible [9].

OntoUML has been successfully employed in a number of industrial projects in several different domains, ranging from Petroleum and Gas [9] to News Information Management [21]. In fact, it has been considered as a possible candidate for contributing to the Object Management Group (OMG) Semantic Information Model Federation (SIMF)