D10 - Pragmatism versus formalism: the relation between Linked Open Data, semantics and ontologies

Auteurs

Laura Daniele (TNO)
Paul Brandt (TNO)

Clearly the central idea behind LOD as well as Ontologies is to facilitate data processing that is grounded in its semantics, e.g., establishing the meaning of data and act accordingly. Ontologies are engineering artifacts that represent the relevant state of affairs in the world, their interrelations and, of course, their significance to the application. Linked open data essentially represents a vision where applications can automatically consume and apply externally published data in a way similar to how humans consume information that has been published on the world wide web. Since both approaches can be considered a means to achieve a similar goal, and hence represent distinct tools, it is necessary to become aware of their differences and similarities before one can decide when, and how, both tools can be applied best. To that end we will investigate the balance between the precision of formalism, coming from the ontological approach, and the speed and readiness of pragmatism, coming from the LOD approach.

This is a preliminary version of this chapter – the final version will be available online.

The term ontology has been used (and often misused) in many different ways in the literature and on the Internet, so we first characterize ontologies here before we can put them in relation with Linked Open Data.

In a nutshell: what are ontologies?

We regard an ontology as an engineering artifact that consists of a set of concepts and definitions used to describe a certain reality, relations among these concepts, plus a set of axioms to constrain the intended meaning of these concepts [1]. Concepts are abstractions of relevant entities in the application domain under consideration and can represent either tangible entities, e.g., a person, a bridge or a piece of paper, or intangible entities, e.g., a color, a flu or a symphony. Moreover, concepts can represent a universal type, e.g., females, but can also be instantiated to represent individuals existing in reality, e.g., the queen Elizabeth II. Relations are relevant associations to be established between concepts and between instances of these concepts, e.g., a marriage between two people and the marriage between John and Alice, respectively. Axioms allow to specify constraints on the

usage of concepts and relations, e.g., the number of passengers that can board a Fokker 70 aircraft should not exceed a certain threshold.
Ontologies, then, capture mental images of the reality, the so-called conceptualizations in the Ullmann triangle [2] in Figure 1. Conceptualization are abstractions of reality that are based on a set of concepts. However, conceptualizations exist in principle in the mind of those who produce them, but they have to be unambiguously communicated to others. Therefore, conceptualizations require a language that allows them to be represented and communicated as concrete explicit specifications, which we call ontologies. This language should be suitable to represent instances of the concepts underlying the conceptualization and should have a formal semantics, which allows not only unambiguous interpretation but also rigorous analysis and reasoning.

Using ontologies

Ontologies are first of all a conceptual tool that can be used to establish a shared model of consensus among humans in a certain domain [3]. In such cases, an ontology should be represented using a general-purpose language that is expressive enough to represent the domain under consideration, regardless of specific technological platforms. In this way, one can abstract from the ways the ontology might be implemented in order to focus on the concepts, relations and axioms that need be specified, ignoring the issue of selecting the most suitable language to express them.

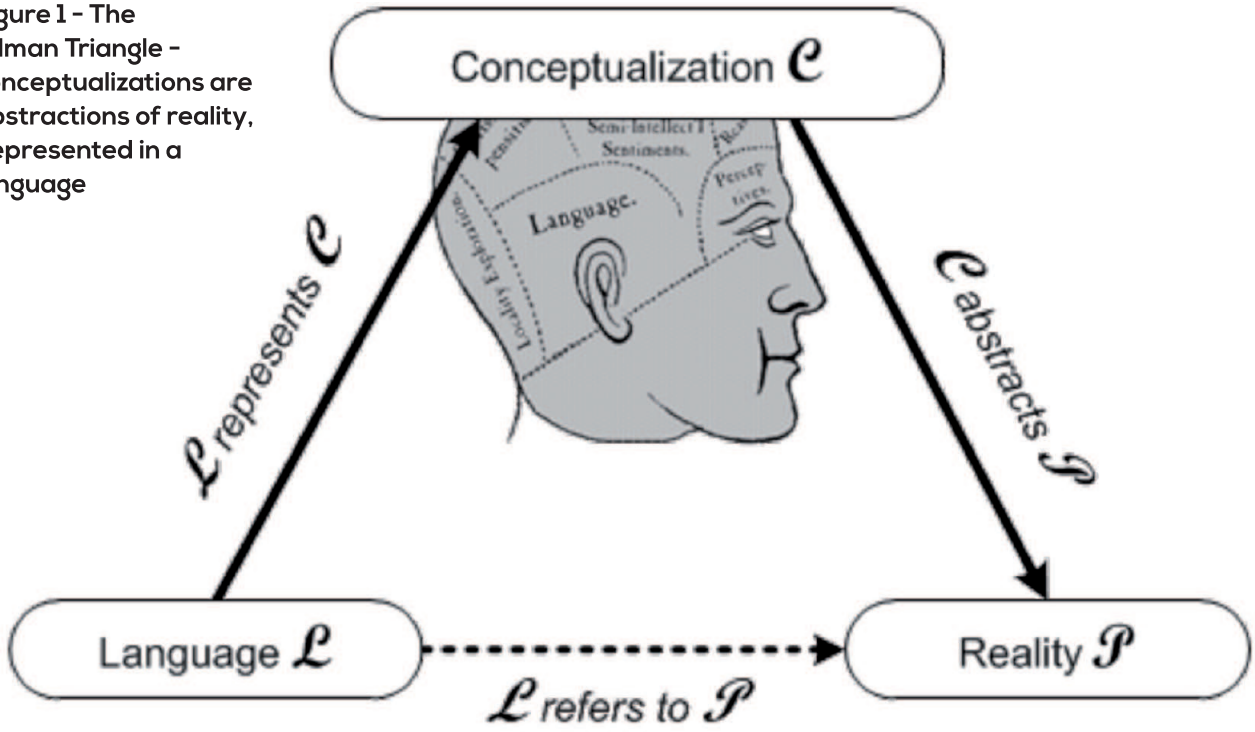

Figure 1 - The Ullman Triangle - conceptualizations are abstractions of reality, represented in a language

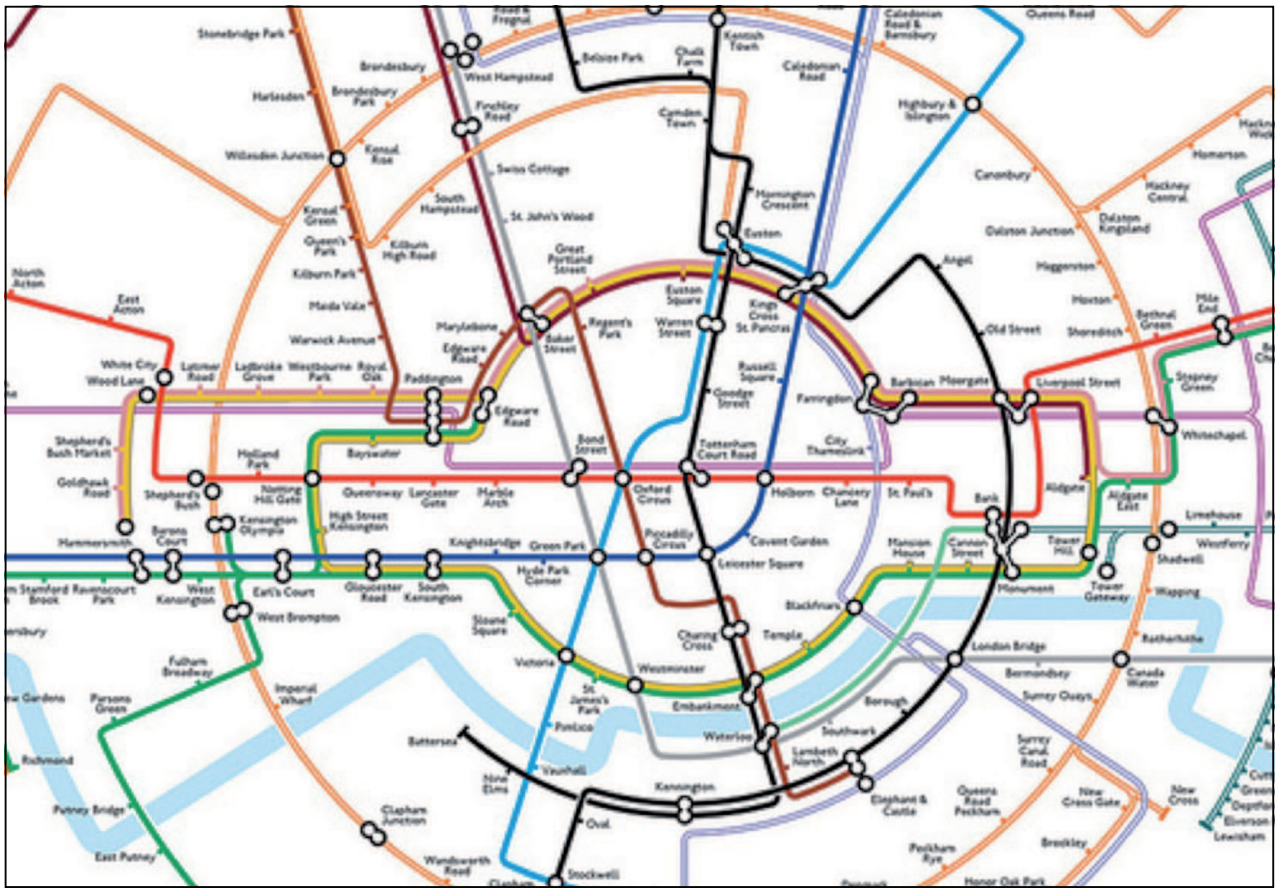

Figure 2 - A map of a city tube: an example of a representation of a conceptualization, using a (visual) language that refers to a certain portion of reality

Secondly, ontologies can be used as a means for establishing interoperability among multiple heterogeneous systems. In this case, an ontology provides a reference model that allows translation and matching, possibly automatically, among multiple heterogeneous systems [4, 5] that have been developed based on different semantic representations. Different systems should be integrated using interfaces that specify the information necessary for interoperability and these interfaces should be built upon a shared model of consensus that is specified by the ontology. In order to allow automated translation and matching, the ontology needs to be represented (implemented) using a language that can be processed by machines.

Finally, ontologies are used to specify and share knowledge about a certain domain in order to facilitate automated reasoning by inference engines {Uschold:2004dx}. This comes very close to application of artificial intelligence, and indeed a popular view is to consider ontologies a subarea of AI where they are applied as knowledge representation backbone, serving as vehicle for carrying domain knowledge.

The language underlying a model

Ontologies, like any other model, need a language for their representation. More precisely, the concepts underlying a certain conceptualization need to be mapped onto the elements of a suitable language. For example, the concept of a motor vehicle as a kind of object that moves on its own for the purpose of transporting people, can be referred to by using the term car (English), auto (Dutch), or macchina (Italian). The ontological commitment of a given language, namely the entities the primitives of a language commit to the existence of [6], influences the expressiveness, accuracy and complexity of the ontology that is being represented. For instance, the original entity-relationship model commits to a view that accounts for the existence, in the world, of entities, relations and properties; nothing more and nothing less. Consequently, temporal aspects cannot be distinguished. Although the database engineer will be able to create a database schema that can store relations to time, it is important to note that this is an artificial model, a codification artifact, as opposed to a conceptual model representing a truthful abstraction of time. The database schema, no matter the qualifications of the database engineer, can never be expected to describe time accurately since the language lacks a decent construct for it. Hence, in order to understand the possible applications for which ontologies can be used, one needs to understand their underlying representational language. Depending on that language, then, ontologies range from being highly informal, namely loosely expressed in natural language, to rigorously formal, namely strictly expressed using formal semantics, theorems and proofs. Informal ontologies may lead to ambiguities, and systems that are based on such ontologies are more error-prone than systems based on formal ontologies, which, in contrast, allow automated reasoning and consistency checking.

> 'The point here is that the choice of a particular codification language can only be justified as a design choice. To put it baldly, the question is not whether, for instance, OWL is good or not for representing ontologies. The question is whether OWL is justifiable as an adequate design choice in a specific design scenario.' - G. Guizzardi [12]

However, there is a trade-off between the expressiveness of a language and its computational power: the more expressive the language, the more complex to process it using machines, up to and beyond the point that its expressive power exceeds the computational power of the machines.

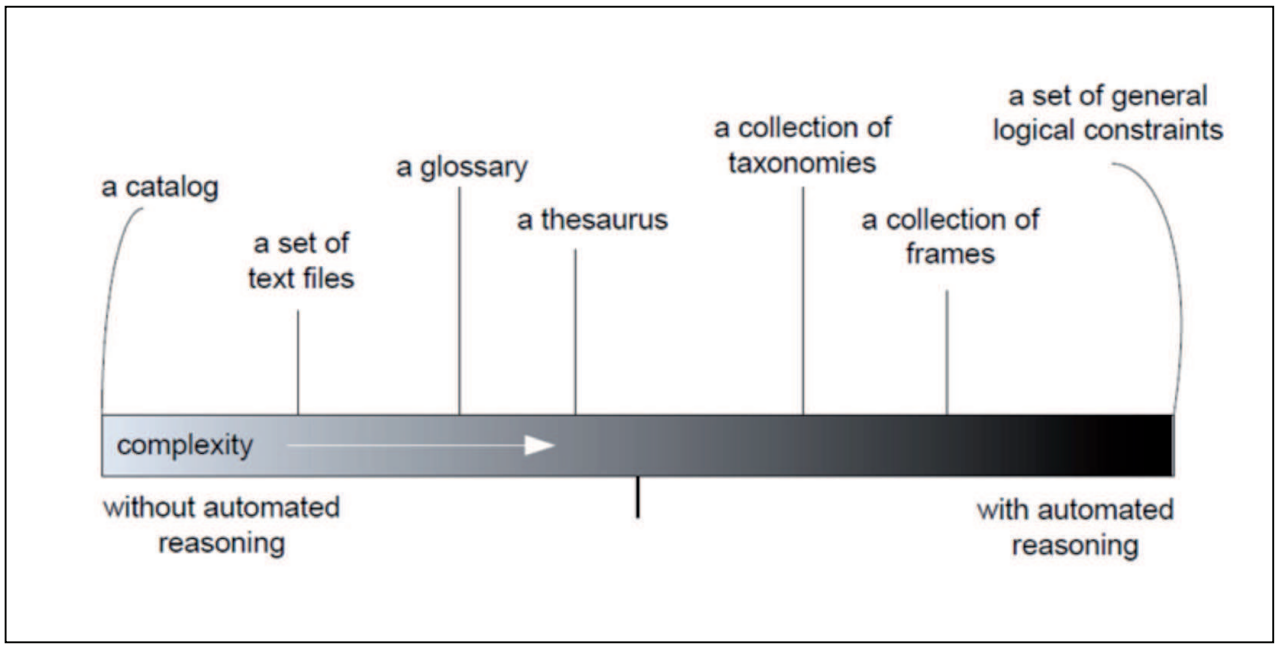

Figure 3 - A range of different kind of ontologies, their complexity and support for automated reasoning

Different kinds of ontologies

Several ontology forms are currently used in information systems and on the Web, with varying expressiveness and complexity. These ontologies span from simple catalogues of concepts related by subsumption relationships, to complete representations of concepts related by complex relationships, including axioms to constrain their intended interpretation. This has been depicted below (taken from [7]).

Going from left to right on this scale, with increasing complexity (i.e., increasing amount of meaning that can be specified), at some point an informal language becomes insufficient to unambiguously express the semantics with the required level of accuracy. Therefore, the degree of formality should increase, providing also more and better support for automated processing of the ontology. The so-called 'lightweight ontologies' as being used in the Semantic Web are usually less complex ontologies represented in the form of a catalog, thesaurus or taxonomy, possibly with a few logical constraints, and are expressed in languages that allow the ontology to be interpreted by machines, such as OWL and RDF. These languages have limited expressiveness that results from their humble ontological commitment, and can therefore carry large ontologies due to the mentioned trade-off with computational power. Hence lightweight ontologies can be easily and readily processed by machines, but remain limited in their expressiveness and their ontological commitment.

The case of Linked Open Data

LOD and RDF are powerful technologies conceived for publishing and exchanging data between machines. As a means for establishing interoperability, LOD are therefore commonly associated with ontologies. However, ontologies are not LOD. Ontologies can be implemented using LOD, making it one out of more possible technological platforms of choice.

LOD and RDF are a way to implement lightweight ontologies. In case of RDF as underlying modeling language, the statements that can be formulated express binary relationship instances. As explained in [8], these statements are represented as (subject, predicate, object) triples, where subjects and predicates are so-called resources, and objects are either resources or literals (constants). Resources are '(...) treated here as synonymous with 'entity', i.e. as a generic term for anything in the universe of discourse' (http://bit.ly/11Mn7wr, accessed June 14, 2013), and, can be identified by Uniform Resource Identifier (URI) references. Literals may be untyped (e.g., 'Netherlands') or typed (e.g., '42'^^xsd:Integer), and are considered to 'denote themselves'2, which implies that their semantics is considered to be self-explanatory.

The ontological commitment of a language is to be sought in its primitives, and in case of RDF we should consider resources and literals as its primitives. The RDF language, therefore, is suitable to represent lightweight ontologies, but is insufficient to cater for the accuracy and expressiveness that are required for more rigorous ontologies with high ontological commitment [9, 10]. Therefore, one should be aware that RDF is a powerful tool to link data using triples, and is driven by the pragmatism of providing timely technological solutions, rather than the precision coming from the world of formal ontologies. As a result, RDF is a suitable tool to improve the accessibility of the data that one wants to publish on the Web. Moreover, RDF enables interoperability to big data stores since it has been widely acknowledged as de-facto standard for semantic representations of data on the Web. However, one should be aware of the limits of the RDF semantic representation, which introduces risks of datasets redundancy, inconsistency, ambiguity and potential false agreements [6, 11]. These problems should not be underestimated because sooner or later they may grow out of proportion and, eventually harm an effective interoperability instead of enable it.

In conclusion: When to use ontologies, when to use LOD?

Formal ontologies are a powerful tool to represent a limited portion of reality (small/medium ontologies) with lots of details (high precision) using languages that allow to express concepts, relationships between these concepts and axioms to constrain the intended meaning of these concepts. Formal languages allow ontologies to be machine interpretable, enabling automated reasoning and consistency checking, while informal languages are more suitable and expressive for human communication, allowing an ontology to be a shared model of consensus among people rather than an artifact interpretable by machines. LOD, at the other hand, are more suitable to represent a big portion of reality (big, lightweight ontologies) with not much details (low precision) using subject/predicate/object triples. Since formal ontologies allow high precision and ontological commitment, there is a bigger chance to make mistakes and introduce inconsistencies, so one should carefully check the correctness and consistency of the ontology, possibly using automated tools. In contrast, the pragmatics of LOD allow for course-grained semantics that is not very accurate, and, therefore, leaves limited room for mistakes (in other words, what you do not state cannot be wrong). This is currently the reason for the success of LOD, however in future will represent the source of emergent false agreements. By then, hopefully, tooling will become available that will assist us in turning this massive LOD data set into something that is based upon a more formal ontological commitment to cater for fine-grained semantics, and that does so with high accuracy and without mistakes: 'The quality of any implementation artifact based on a model is ultimately bound by the quality of that model.' [12]

References

[1] N. Guarino, 'Formal Ontology in Information Systems', in Proceedings of the 1st International Conference on Formal Ontology in Information Systems (FOIS), 1998, pp. 3-15.

[2] S. Ullmann, 'Semantics: An Introduction to the Science of Meaning', Barnes & Noble, 1979.

[3] R. Burkhard and M. Meier, 'Tube map: Evaluation of a visual metaphor for interfunctional communication of complex projects', in Proceedings of I-KNOW'04, 2004.

[4] A. M. Ouksel and A. Sheth, 'Semantic interoperability in global information systems', ACM SIGMOD record, vol. 28, nr. 1, pp. 5-12, mrt. 1999.

[5] M. Uschold and M. Gruninger, 'Ontologies and semantics for seamless connectivity', ACM SIGMOD record, vol. 33, nr. 4, p. 58, dec. 2004.
[6] G. Guizzardi, Ontological foundations for structural conceptual models, CTIT, Centre for Telematics and Information Technology, 2005.
[7] B. Smith and C. Welty, 'Ontology: Towards a New Synthesis', in Proceedings of the international conference on Formal Ontology in Information Systems - FOIS '01, 2001, vol. 2001, pp. 3-9.
[8] C. Bizer, T. Heath, and T. Berners-Lee, 'Linked data-the story so far', International Journal on Semantic Web and Information Systems (IJSWIS), vol. 5, nr. 3, pp. 1-22, 2009.
[9] H. Kim, 'Pragmatics of the Semantic Web', in Semantic Web Workshop, 2002, pp. 1-2.
[10] P. Jain, P. Hitzler, P. Z. Yeh, K. Verma, and A. P. Sheth, 'Linked Data Is Merely More Data', in AAAI Spring Symposium on Linked Data Meets Artificial Intelligence, 2010, pp. 82-86.
[11] N. Guarino, D. Oberle, and S. Staab, 'What is an Ontology?', in in Handbook on Ontologies, 2009.
[12] G. Guizzardi, 'Theoretical foundations and engineering tools for building ontologies as reference conceptual models', Semantic Web, vol. 1, nr. 1, pp. 3-10, jan. 2010.