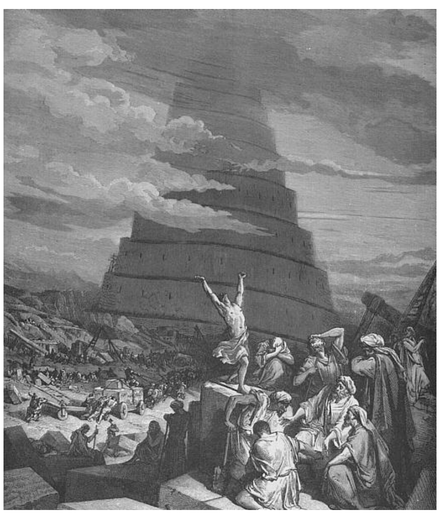
The Babel of the Semantic Web Tongues – In Search of the Rosetta Stone of Interoperability

Oliver Kutz, University of Bremen, Germany
Christoph Lange, University of Birmingham, UK
Till Mossakowski, DFKI GmbH, Bremen, Germany
C. Maria Keet, University of KwaZulu-Natal, South Africa
Fabian Neuhaus, National Center for Ontological Research, USA
Michael Gruninger, University of Toronto, USA

Of these were the isles of the nations divided in their lands,
every one after his tongue,
after their families, in their nations. (Genesis 10:5)

Abstract. We discuss a vision of the Semantic Web where ontologies, services, and devices can seamlessly interoperate across a multitude of protocols and languages. In particular, we discuss the importance of enabling interoperability for Semantic Web technologies in the knowledge representation layer, and give an overview of the Distributed Ontology Language DOL addressing this aspect, representing a first piece of a Rosetta Stone enabling overall interoperability.

The Babylonian Confusion



The Tower of Babel (image from Wikipedia)

The Semantic Web has led to an endless number of different standards: XML, RDF, RDFS and RDFa are used for the exchange of (possibly large) datasets. Ontological knowledge can be represented in the different profiles of OWL, which represent trade-offs between expressiveness of the languages and effectiveness of available tools for certain reasoning tasks. On one side of this spectrum there are large ontologies like SNOMED CT, expressed in logics of low expressivity such as OWL EL. In the opposite direction, the spectrum does not end with the relatively expressive, but still decidable, OWL 2 DL; rather, it is common practice (e.g. in bio ontologies) to intersperse OWL ontologies with first-order axioms in the comments or annotate them as having temporal behaviour [Smith *et al*., 2005, Beisswanger *et al., 2008*], although, unfortunately, these axioms will be ignored by tools. Foundational ontologies, such as DOLCE, BFO or SUMO, also use full first-order logic (and Common Logic is an ISO-standardised language with first-order expressivity) or even first-order modal logic. Even though such ontologies may not always be (considered as) part of the Semantic Web, many of them have the ontology's symbols identified by IRIs and partial OWL implementations are available (e.g. for DOLCE and BFO), and foundational ontologies serve as a methodological guideline for the design of (ontologically well-constructed) domain ontologies.

The large-scale commercial ontology Cyc provides a rich formalisation of common sense, and involves all kinds of logics, like first-order, higher-order, contextualised and non-monotonic logics. The latter also play a role in the W3C standard RIF (rule interchange format), which actually comprises a whole family of rule-based languages, trying to capture the important features of input languages of current industrial rule-based tools.

The result is a **Babylonian confusion**[1] of languages used in ontology engineering, and whilst certain relationships between some of the languages are well-studied, such as logical translations, others are just being begun to be investigated and understood, e.g. useful relations between OWL and RIF-PRD. Indeed, Tim Berners-Lee's original version of the Semantic Web layer cake has featured a rich number of heterogeneous logical languages involved in the Semantic Web, as depicted in Figure 1.

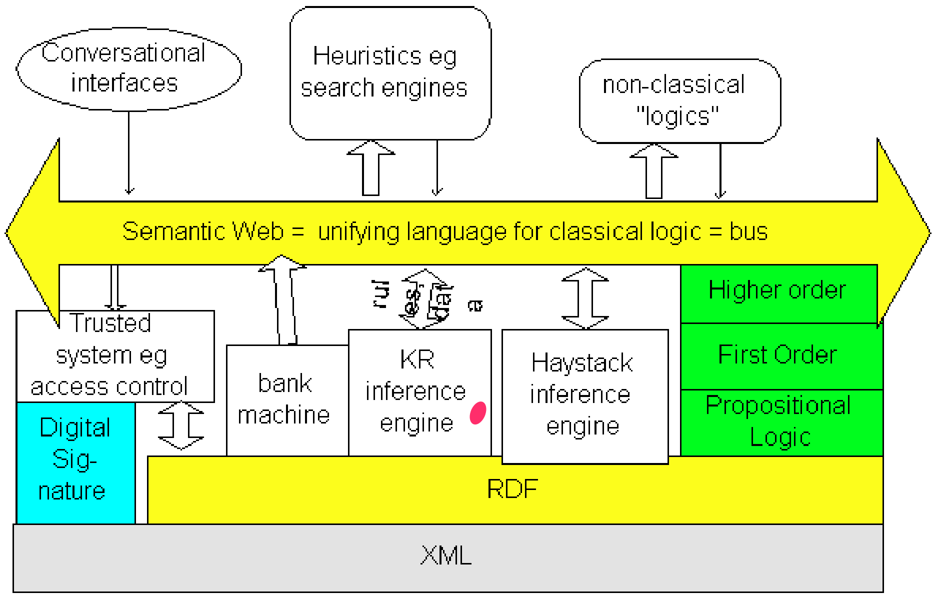


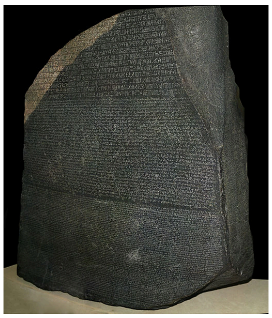
Figure 1. Tim Berners-Lee's early version of the semantic web layer cake

The original vision of the Semantic Web [Berners-Lee et al., 2001] emphasized the role of intelligent *agents*, which combine information found on the Web to assist with complex tasks such as making an appointment with a nearby doctor who specialises on the user's current disease. Agents can rely on *web services* that solve specialised problems and combine these in order to provide more powerful services. W3C standards and submissions for services include the Web Service Description Language (WSDL) for the specification of interfaces of single web services, the Web Service Choreography Description Language (WSCDL) for the specification of the interplay of decentralised services, and others such as WSCL, WS-Transfer, WS-Eventing etc., not to mention important non-W3C languages such as BPEL and BPMN. Again, this multitude of languages is only partly due to idiosyncrasies of companies and organisations, but more importantly also due to different intended characteristics and features of the languages, e.g. expressing different aspects of (composed) services.
Another reason for utilising different languages has become apparent with the expansion of the Linked Open Data cloud. The Linked Open Data cloud is particularly interesting in the context of so-called big data – i.e. data sets so large that their capturing, storage, management, and processing is a challenge in itself [Big Data, 2012]. Big data might, for example, originate from large-scale scientific experiments, social networks, or sensor networks. In the following, we restrict ourselves to big data that has been made available as linked data. With the current state of the art, this means that such datasets will be described using vocabularies with a weak semantics (typically, a subset of RDFS plus a few hand-picked OWL constructs). Agents consuming these data sets require stronger semantics (without sacrificing scalability

[1] Babel is also the name of an early conversion tool.

when used locally). Therefore, the required ontologies will be implemented and maintained in languages different from those used for the data sets. Again, different languages are used side-by-side to describe aspects of the same problem space.

Going beyond virtual agents and services, in the future, embodiment will play a greater role. A recently fast evolving area is *smart environments*, which provide embodied services. Here, the issue of service description arises in a similar way as for web services, and some standards such as the [‘universal remote console’ (URC)]() have been defined. Moreover, the (social) interactions between such embodied services and human agents acting in such environments (e.g. using intelligent dialogue systems) will clearly be a research topic of increasing importance in the future, especially in an aging society, and may be seen as the most challenging interoperability problem of all.



The Rosetta Stone
(image from Wikipedia)

The Vision of Interoperability

This multitude of languages and endeavours bring about an *interoperability* problem, which other activities try to counteract and overcome – including standardisation efforts. The diversity of current interoperability initiatives demonstrates, however, that there is currently no unified framework available within which the various interoperability efforts themselves could be synchronised and orchestrated. We expect that, by 2022, a **Rosetta stone of interoperability**, bridging this Babylonian confusion and extending Tim Berners-Lee’s original vision, will have been found. This Rosetta stone will ensure interoperability within and among the areas of knowledge engineering, services and devices. In each of these areas, interoperability will occur at various levels:

1. interoperability at the level of individual **data**, services, and devices;
2. interoperability at the level of **models**: ontologies (ontology alignment/integration), service descriptions (service matching), and device descriptions;
3. interoperability among different **metamodels**: ontology languages, service and device description languages.

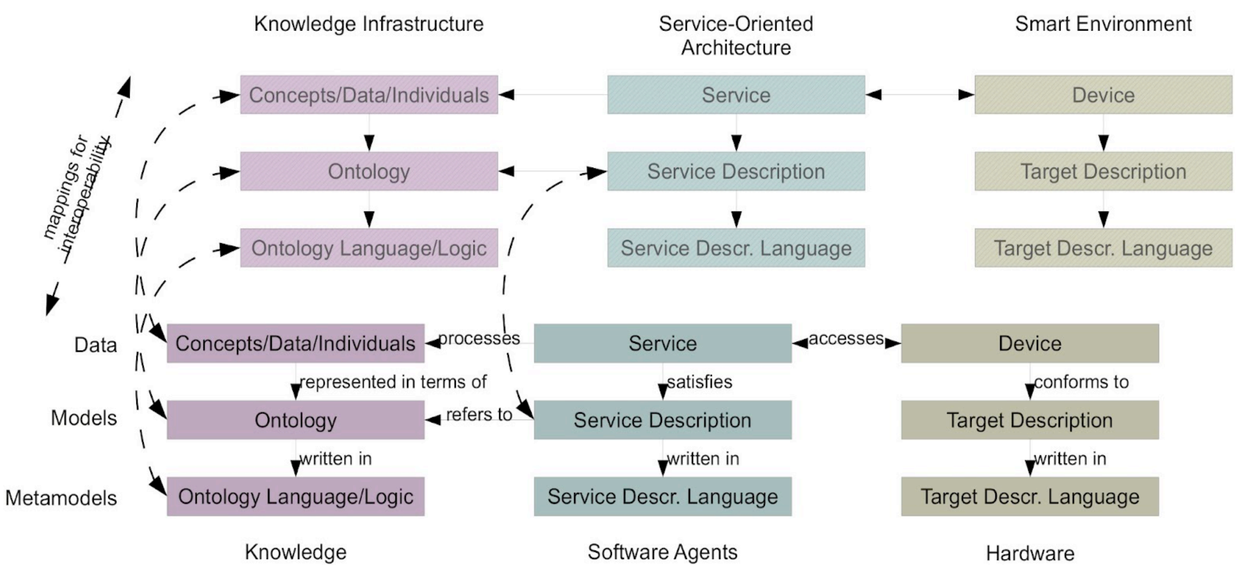


Figure 2. The interoperability stack

With such a *systematic* and *flexible* interoperability *at* all and *across* all of these levels, one can integrate data stemming from different sources, using different schemas, and formulated in possibly different schema languages. This also means that translations occur at all three levels: e.g. for the left column in the picture, we may need translation of data, translation of ontologies (ontology alignment), and translation of ontology languages. Then, much of the content written in hitherto unrelated languages can be connected. We can concentrate on the *content, services, and devices*, and find out more easily whether different pieces of content, different services and different devices can be related and integrated in a meaningful way or not.

The Distributed Ontology Language (DOL)

The vision depicted in Figure 2 is quite broad, and its realisation involves efforts in several areas. As a first step towards interoperability shown in the left column of Figure 2, we here sketch the **Distributed Ontology Language (DOL)**, a metalanguage for ontology integration and interoperability, which accepts the diverse reality found within the Semantic Web.[2] The process of standardising DOL within the ISO is to be finished in 2015, and tool support is under way. DOL allows for maintaining ***one connected, distributed ontology*** per application instead of two or more ***separate, disconnected*** implementations and ontologies.

Although the foundations of the Semantic Web – IRIs and RDF graphs – are likely to be sufficiently strong for integrating most desirable ontology languages in the foreseeable future, it is important to recognise that there will be a diversity of languages on top of RDF used to express ontologies. DOL is not “yet another ontology language”, but it provides a meta-level framework that integrates different ontology languages and makes them interoperable – and this regardless of whether their *syntax* is compatible with RDF or not, as long as their *semantics* can be formalised in a set-theoretic or institution-theoretic way [Mossakowski et al., 2012]. A distributed ontology consists of modules, which may be implemented in different DOL-conforming ontology languages, and which are interlinked by formal, logical links such as imports or theory interpretations, or informal, non-logical alignments (as returned, e.g., by statistical matching procedures) [Kutz et al., 2010]. DOL is to our knowledge the first language that systematically supports the expression of such a collection of links, indeed even the first language for the subcase of only homogeneous links (e.g. between two OWL ontologies). Heterogeneous logical links, i.e. across ontology languages, are *semantically* backed by a graph of logic translations (towards more expressive logics) and projections (towards less expressive logics). Links – logical as well as non-logical ones – across ontology languages are *syntactically* backed by the following abstraction, which the DOL data model makes over all ontology languages: a basic ontology (any single-language module in a distributed ontology) consists of *symbols* identified by IRIs, and of *sentences*. Where a basic ontology language does not natively use or enforce IRIs as identifiers, such as Common Logic, DOL provides the necessary “namespacing” constructs. The symbol abstraction has so far proven adequate across several standard ontology languages – which, of course, provide different *kinds* of symbols and sentences: OWL, e.g., supports classes, individuals, object properties, and data properties, and distinguishes declarations from axioms, whereas Common Logic supports names (of individuals, which can also be functions or relations, depending on context) and sequence markers (denoting sequences of such individuals) and does not require symbols to be declared before using them in sentences (“signature-free” approach). The basic ontologies in a distributed ontology can be given in place, using any DOL-conforming serialisation of the

[2] The issue of separate maintenance and linking ontologies is addressed only partially for so-called “networked ontologies” as investigated in the NeOn project. NeOn, however, mainly remains within the OWL framework, whereas even the alignment of two OWL ontologies may require facilities beyond standard OWL imports, even beyond the Alignment API [David et al., 2011], viz. much more expressive logics than OWL.

respective basic ontology language (e.g. RDF/XML for OWL, or CLIF for Common Logic), or they can be referred to by IRI.

The distributed ontology approach also promises to reconcile the knowledge representation requirements of intelligent agents and big data management software. In the past, tailor-made ontology languages have tried to mitigate the conflict between semantically rich descriptions for intelligent agents and the need to store huge amounts of data; consider OWL EL in biomedicine, which is sufficiently expressive to model basic relations between body parts while still scaling to large ontologies such as SNOMED CT. In general applications, such a compromise will be hard to find. This even holds true for applications in single but interdisciplinary domains; consider the ecology domain, which involves ontologies of plants, insects, soil, etc., and their interrelations, translational medicine for a particular disease, or the food domain, which involves ontologies of primary-produce agriculture, storage, processing, packaging, quality, safety, etc.[3] In DOL, the connection of a linked dataset to an expressive ontology that formalises the dataset's vocabulary is just a special case of a distributed ontology. This pattern makes linked datasets accessible to intelligent agents, and is supported as follows: 1. A dataset, represented as an RDF graph, is conceived as a basic ontology in the logic RDF. 2. The vocabulary used in this RDF graph is formalised in one or more basic ontologies in languages that are sufficient for expressing the semantics required by the respective agents. Complex settings may require complementary descriptions of different aspects of the same concepts in different basic ontology languages (see [Mossakowski et al., 2012] for an example from mereology, where some aspects of the parthood relation are modeled in OWL and others in Common Logic). 3. The link between the dataset and the formalisation of its vocabulary is established by a distributed ontology, which projects the vocabulary ontologies down to the RDF logic and then extends them with the RDF dataset (cf. [Lange et al., 2012]). With DOL, relations necessary for interoperability can be expressed in a semantically meaningful way, such that heterogeneous reasoning may infer new facts.

The DOL language is almost feature-complete at the time of this writing, with a committee draft standard (the first formal step after the working draft stage) expected in September 2012. The remaining ISO standardisation process will then focus on fine-tuning the details of syntax and semantics in accordance with the diverse group of stakeholders, seeking a consensual and comprehensible terminology of the standard document, specifying in detail the conformance of W3C- and ISO-standardised basic ontology languages with DOL, and spreading the word to communities around important non-standard ontology languages such as the schema.org data model. Actually, the latter will be integrated as a DOL-conformant language. Some use cases for DOL have been summarised in [Lange *et al.*, 2012].

Tools for Interoperability

The potential success of a language strongly depends on available tool support. In the following, we briefly describe the current state of the two main DOL-aware tools and outline further steps towards scaling them up to our interoperability vision. Hets, the **Heterogeneous Tool Set**, supports logically heterogeneous reasoning on modular ontologies; its native metalanguage HetCASL has influenced the design of DOL, and we are adding native DOL support at the moment. Like DOL, Hets is based on a graph of (implementations of) languages and translations. The **Ontohub.org** ontology repository engine [Lange et al., 2012] allows for managing (linking, annotating, discussing) ontologies in arbitrary languages by treating them as abstract collections of symbols and sentences; its database schema matches the abstract syntax of DOL, and we are developing an interface that serves all ontologies in the database as linked data at the moment. Ontohub employs Hets for structurally validating any ontology imported into the database, and for obtaining a logic-independent representation of the symbols and sentences found in any such ontology.

[3] On food, see, e.g., the ontologies of the SEAMLESS European project.

Several improvements can be made to Hets and Ontohub to better realise our vision of the future Semantic Web. Currently, Hets parses each module of a distributed ontology before being able to validate the distributed ontology as a whole. However, with the modular approach in the distributed ontology network that we envision, it should be able to fetch individual parts of an ontology on demand, in a follow-your-nose manner[4]. While Ontohub's data model is inherently linked data compliant, Ontohub needs to be turned into a linked data *client* in addition to the linked data server that it is now. Instead of first importing remote ontologies into its local database before being able to link, annotate and discuss them, it should be possible to just point to them. On the "intelligence" side, Ontohub needs a closer integration of Hets: beyond structural validation of basic ontologies, it should serve as a frontend to Hets for heterogeneous reasoning, and it should give access to operations supported by Hets, such as translating a basic ontology into a different ontology language. Additionally, Ontohub should offer language-specific ways of accessing ontologies: instead of just treating them as sets of symbols and sentences, it should be able to display a class hierarchy for OWL ontologies, and to generalise this notion of subsumption to other logics (e.g. first-order logic).

Conclusion

Facing the Babylonian confusion of the multitude of languages used in the Semantic Web, the need for interoperability arises at various levels: the syntactic and semantic level; data and services; object and meta level. We expect that in the increased presence of linked data and especially big data, this need will even become stronger due to the heterogeneity of the data. We have introduced the Distributed Ontology Language DOL as a means for providing interoperability among ontologies. In order to come close to a Rosetta Stone of interoperability, similar efforts at the other levels will be necessary.

Acknowledgments

We would like to thank the OntoIOp working group within ISO/TC 37/SC 3 for providing valuable feedback, in particular John Sowa and Sue Ellen Wright.

Bibliography

- Beisswanger, E.; Schulz, S.; Stenzhorn, H.; Hahn, U.: BioTop: An upper domain ontology for the life sciences - a description of its current structure, contents, and interfaces to OBO ontologies. Applied Ontology 3(4) (2008) 205-212.
- Berners-Lee, Tim; Hendler, James; Lassila, Ora. The Semantic Web, Scientific American, May 2001, p. 29-37.
- Big Data (anonymous authors). Challenges and Opportunities with Big Data. White paper. 2012.
- David, Jérôme; Euzenat, Jérôme; Scharffe, François and Trojahn dos Santos, Cássia. The Alignment API 4.0, *Semantic web journal* 2(1):3-10, 2011
- Floridi, Luciano. The Semantic Web vs. Web 2.0: A Philosophical Assessment, Episteme, 2009, 6, 25-37.
- Gärdenfors, Peter. How to Make the Semantic Web More Semantic, Proc. of FOIS 2004, IOS Press.
- Goguen, Joseph A. Data, Schema, Ontology and Logic Integration, Logic Journal of the IGPL, Vol. 13: 685-715, 2005.
- Kutz, Oliver; Mossakowski, Till; Luecke, Dominik. Carnap, Goguen, and the Hyperontologies, Logica Universalis: 4(2), Volume 4, Number 2, 255-333, 2010.
- Lange, Christoph; Mossakowski, Till; Kutz, Oliver; Galinski, Christian; Grüninger, Michael; Couto Vale, Daniel. The Distributed Ontology Language (DOL): Use Cases, Syntax, and Extensibility, *Terminology and Knowledge Engineering Conference (TKE)* 2012.
- Mossakowski, Till; Lange, Christoph; Kutz, Oliver. Three Semantics for the Core of the Distributed Ontology Language. In Michael Grüninger (Ed.), FOIS 2012: 7th International Conference on Formal Ontology in Information Systems
- Smith B; Ceusters W; Klagges B; Kohler J; Kumar A; Lomax J; Mungall CJ; Neuhaus F; Rector A; Rosse C. Relations in Biomedical Ontologies.Genome Biology, 2005, 6:R46

[4] http://patterns.dataincubator.org/book/follow-your-nose.html