Ontology Integration Using $\mathcal{E}$-Connections

Bernardo Cuenca Grau$^1$, Bijan Parsia$^{2,3}$, and Evren Sirin$^2$

1 University of Oxford, UK
berg@comlab.ox.ac.uk
2 University of Manchester, UK
bparsia@cs.man.ac.uk
3 Clark & Parsia, USA
{bijan, evren}@clarkparsia.com

Summary. The standardization of the Web Ontology Language, OWL, leaves (at least) two important issues for Web-based ontologies unsatisfactorily resolved, namely how to represent and reason with multiple distinct, but linked ontologies, and how to enable effective knowledge reuse and sharing on the Semantic Web. In this paper, we present a solution for these problems based on $\mathcal{E}$-Connections. We aim to use $\mathcal{E}$-Connections to provide modelers with suitable means for developing Web ontologies in a modular way and to provide an alternative to the owl:imports construct.

With such motivation, we present in this paper a syntactic and semantic extension of the Web Ontology language that covers $\mathcal{E}$-Connections of OWL-DL ontologies. We show how to use such an extension as an alternative to the owl:imports construct in many modeling situations. We investigate different combinations of description logics for which it is possible to design and implement reasoning algorithms, well-suited for optimization. Finally, we provide support for $\mathcal{E}$-Connections in both an ontology editor, Swoop, and an OWL reasoner, Pellet.

11.1 Motivation

The Semantic Web architecture has been envisioned as a set of new languages that are being standardized by the World Wide Web Consortium (W3C). Among these languages, the Web Ontology Language (OWL) plays a prominent role, and Description Logics have deeply influenced its design and standardization [1][20]. Two of the three variants, or dialects, of OWL, namely OWL-Lite and OWL-DL, correspond to the logics $\mathcal{SHIF}(\mathcal{D})$ and $\mathcal{SHOIN}(\mathcal{D})$, respectively [20] [26] [28].

The acceptance of OWL as a Web standard will yield to the rapid proliferation of DL ontologies on the Web and it is envisioned that, in the near future, the Semantic Web will contain a large number of independently developed ontologies. The standardization of OWL, however, leaves two important issues for Web-based ontologies unsatisfactorily resolved, namely how to represent and reason with multiple distinct,
but linked ontologies, and how to enable effective knowledge reuse and sharing on the Semantic Web.

First, in order to provide support for integrating ontologies, OWL defines the \texttt{owl:imports} construct, which allows to include by reference in an ontology the axioms contained in another ontology, published somewhere on the Web and identified by a global name (a URI). The functionality provided by this construct, however, is unsatisfactory for a number of reasons [8]:

- The only way that the \texttt{owl:imports} construct provides for using concepts from a different ontology is to bring into the original ontology all the axioms of the imported one. Therefore, the only difference between copying and pasting the imported ontology into the importing one and using an \texttt{owl:imports} statement is the fact that with imports both ontologies stay in different files. This certainly provides some \textit{syntactic} modularity, but not a \textit{logical} modularity, which would be indeed more desirable.

- The entities of an ontology, such as classes and properties, are, as the ontologies themselves, identified by unique names (URIs) on the Semantic Web. For example, suppose that we are developing an ontology about “People” and we want to define the concept of a “Dog Owner”. It may seem natural for such a purpose to use the URI of a certain class “Dog”, that appears in an ontology about “Pets” that we have found on the Web. We may think then that we are committing to the meaning of “Dog” in that ontology (a dog is an animal, for example). Nevertheless, if we use the URI for “Dog” without importing the corresponding ontology, we are bringing \textit{nothing} from the meaning of the term in the the foreign ontology, while if we import it, we are bringing all the axioms of the “Pet” ontology to our logical space, even if we are only interested in dogs, and not in cats or hamsters.

- The use of \texttt{owl:imports} results in a flat ontology, i.e., none of the imported axioms or facts retain their context. While it is possible to track down the originator(s) of some assertions by inspecting the imported ontology, OWL reasoning does not take such context into account.

Second, enabling knowledge reuse and sharing has always been a major goal of the Web Ontology Working Group. Ontology engineering is a time-consuming task. As more ontologies are built and become available, and as the size of ontologies grows, knowledge sharing and reuse become important issues. On the one hand, when ontologies grow, they become harder for the reasoners to process and for humans to understand, and also harder to reuse. On the other hand, as more ontologies become available, the advantages of reusing existing ontologies become more apparent. In order to make reuse and sharing easier, ontologies should be designed as mostly independent and self-contained modules [30][29]. Intuitively, a module should contain information about a self-contained application domain. Then, suitable means should be provided for integrating and connecting those modular ontologies.

In this paper, we present an approach for tackling these two issues based on \textit{E}-Connections. The \textit{E}-Connections technique [25] [24] is a method for combining logical languages. The main motivation of \textit{E}-Connections is to combine decidable logics