Adapt OWL as a Modular Ontology Language (A Position Paper)*

Jie Bao and Vasant Honavar

Artificial Intelligence Research Laboratory,
Department of Computer Science
Iowa State University, Ames, IA 50011-1040, USA
{baojie, honavar}@cs.iastate.edu

Abstract. OWL, the most commonly used web ontology language, has limited provision for handling modular ontologies. Specifically, the `owl:imports` construct for linking ontology modules lacks support for partial reuse of or localized semantics for the linked ontology modules. We propose an alternative to `owl:imports`, namely, *semantic importing*, for linking multiple modular ontologies with support for localized semantics, partial ontology reuse, and distributed reasoning.

1 Introduction

The semantic web relies on shared ontologies for interoperability of a network of loosely coupled, distributed, autonomous information sources and services. Building large ontologies requires collaboration among multiple autonomous individuals or groups. Because ontologies designed with specific domains, applications (e.g., medicinal chemistry), or users in mind often need significant changes or additions before they can be successfully deployed in a related but different setting (e.g., translational medicine), there is an urgent need for support for selective reuse of independently developed ontology modules.

OWL[1] provides the `owl:imports` construct for linking multiple OWL ontologies to form a larger OWL ontology. However, such a syntactic importing solution of OWL suffers from several limitations:

- Lack of support for *localized semantics*. As Bouquet et.al. [5] have previously observed, OWL provides only a *global semantics* for an ontology module that satisfies *all* the axioms and facts in *all* ontology modules that are(directly and indirectly) imported by that module. Thus, the importing of a module B by module A is tantamount to the copying of *all* statements of module B into module A.
- Lack of support for *partial reuse of ontology module*. The lack of support for selective reuse of parts of one ontology module limits the ability to reuse ontology modules because using the module in its entirety might introduce unwanted inconsistencies or impact performance (because of the presence of a large number of assertions that might be irrelevant in the context of the target application).

* This research was supported in part by the US NSF award 0639230

[1] http://www.w3.org/TR/2004/REC-owl-ref-20040210/

Consequently, there is a growing interest in OWL language constructs (or language extensions) to support modular ontologies, including several *syntactic* extensions to OWL including dOWL [1], C-OWL [5] and $\mathcal{E}$-Connections [6]. However, they are also limited in several ways:

– Limited expressivity [2]. C-OWL does not support linking two classes in different modules with roles, and $\mathcal{E}$-Connections has no direct support for inter-module class subsumption.
– Reasoning difficulties [2] arising from the lack of mechanisms to prevent arbitrary domain relations in C-OWL and the requirement of strict domain disjointedness in $\mathcal{E}$-Connections.

Against this background, we argue for a *semantic* extension to OWL, i.e. a new interpretation of `owl:imports` grounded in *modular semantics* which allows the connected ontology modules to have partially overlapping local interpretations to ovecome some of the existing limitations of existing approaches to linking OWL ontology modules.

2 A Modular Semantics for owl:imports

Instead of introducing syntactic extensions to OWL, we propose a new semantics for `owl:imports` based on an interpretation of `owl:imports` that provides direct support for modular ontologies. Our proposal differs from existing approaches in several respects:

1. support for both inter-module class subsumptions (e.g. $i : C \sqsubseteq j : D$) and inter-module role relations (e.g. $i : C \sqsubseteq \exists i(: r).(j : D)$); use of importing approach instead of the linking approach used in C-OWL and $\mathcal{E}$-Connections;
2. relaxation of the strict local domain disjointness that is required by $\mathcal{E}$-Connections (thereby avoiding the associated reasoning difficulties arising from such an assumption).

The result is an approach that offers a modular semantics for OWL ontologies that instead of requiring the ontology modules to share a completely overlapping model, allows them to have only partially overlapping models. The basic intuition is that when a symbol is shared by different modules, the restrictions on the interpretation of that symbol must also be shared by those modules.

Formally, given a set of OWL ontology modules $\{O_i\}_{i \in I}$, we say a module O_j imports a term $i : t$ if O_j (directly or indirectly) imports O_i and refers $i : t$ (denoted as $i \xrightarrow{t} j$). A modular semantics for OWL can be specified as:

Definition 1 (Modular OWL Interpretation) . *A modular OWL interpretation for a set of ontology modules* $\{O_i\}_{i \in I}$ *is a family* $\mathcal{I} = \{\mathcal{I}_i\}_{i \in I}$ *, where each* $\mathcal{I}_i = \langle \Delta_i, (.)^{\mathcal{I}_i} \rangle$ *is the local interpretation of* O_i*. For any term importing relation* $i \xrightarrow{t} j$*, we have* $t^{\mathcal{I}_i} = t^{\mathcal{I}_j}$*.*

The resulting semantics differs from the current OWL semantics in that it does not require the complete overlapping of local domains, i.e., $\Delta_i = \Delta_j$, for any i, j. It also relaxes the (explicit or de facto) domain disjointness adopted by $\mathcal{E}$-Connections and C-OWL. Thus, it offers a *selective* importing mechanism that allows the parts of an imported ontology module that are selected for reuse by another module to share their interpretation with that module whereas the other parts of the ontology (i.e., those that are not selected for reuse) retain their local interpretations.

Example: Consider two ontology modules modelling domain knowledge about wine and food, respectively. Suppose the food module contains the following terms and axioms:

$$\mathsf{food : Apple} \sqsubseteq \mathsf{food : Fruit} \quad (1)$$

$$\mathsf{food : Grape} \sqsubseteq \mathsf{food : Fruit} \quad (2)$$

Suppose the wine module imports the food module and contains axioms:

$$\mathsf{wine : WineGrape} \sqsubseteq \mathsf{food : Grape} \quad (3)$$

$$\mathsf{wine : Wine} \sqsubseteq \exists \mathsf{wine : madeFrom.(food : Grape)} \quad (4)$$

An interpretation of the ontology contains two local interpretations:

- $\mathcal{I}_1$: $\mathsf{food : Fruit}^{\mathcal{I}_1} = \{\mathsf{x}_1, \mathsf{x}_2\}, \mathsf{food : Apple}^{\mathcal{I}_1} = \{\mathsf{x}_1\}, \mathsf{food : Grape}^{\mathcal{I}_1} = \{\mathsf{x}_2\}$
- $\mathcal{I}_2$: $\mathsf{food : Grape}^{\mathcal{I}_2} = \mathsf{wine : WineGrape}^{\mathcal{I}_2} = \{\mathsf{x}_2\}$, $\mathsf{wine : Wine}^{\mathcal{I}_2} = \{\mathsf{x}_3\}$, $\mathsf{wine : madeFrom}^{\mathcal{I}_2} = \{\langle \mathsf{x}_3, \mathsf{x}_2 \rangle\}$

Hence, the importing relation from food to wine is *not* complete, but *partial* in that only the terms selected for reuse (e.g. food:Grape) are interpreted in the shared part of local domains ($\{x_2\}$ in the example), whereas the terms that are *not* selected for reuse (e.g. food:Apple) retain their *local* interpretations.

The example also shows that the semantic importing strategy supports both inter-module class subsumptions (e.g. axiom 3) and inter-module role relations (e.g. axiom 4).

3 Reasoning with Modularized owl:imports Semantics

Package-based Description Logics (P-DL) [2] adopts the semantic importing approach that allows local domains of ontology modules to be partially overlapping. A distributed reasoning algorithm for P-DL $\mathcal{ALCP_C}$, i.e. $\mathcal{ALC}$ extended with concept importing between ontology modules has been described in [3]. Recently, this approach to reasoning with modular ontologies has been shown to work with modular counterparts (e.g., $\mathcal{SHOIQP_C}$) of fairly expressive description logics $\mathcal{SHOIQ}$ wherein $\mathcal{SHOIQ}$ (OWL-DL) modules are connected using concept name importing [4]. Hence, if we were to adopt OWL with the proposed new `owl:imports` semantics as the syntax for P-DL, or equivalently, adopt the P-DL semantics for OWL-DL, we can offer support for selective reuse of modular ontologies, which ensures that individuals that are imported from one module into another module (through the `owl:imports` mechanism) to share a common interpretation. It thereby ensures that the inferences resulting from a distributed reasoning process are identical to those that are obtained from an integrated ontology that combines the selected parts of the relevant ontologies into a single centralized ontology [2] .

Pan et.al. [7] have recently proposed an extension to OWL that allows both syntactic importing (using the `owl:imports` construct) and semantic importing (using a new `owlx:semanticImports` construct). However, reasoning in such a setting, unlike in the case of our proposal [3], is limited to a pair of ontology modules with no cyclic importing.

4 Conclusions

The proposed modular semantics for `owl:imports` offers a promising approach to supporting partial (and selective, and hence context or application specific) reuse of existing ontology modules. In particular,

- It supports both inter-module class subsumption and inter-module property relations.
- Because it requires no new syntactic extensions to OWL, it ensures backward compatibility of the resulting modular OWL ontologies processable using existing tools in settings that do not require support for selective reuse of ontology modules).
- It allows parts of an ontology module that are not shared with another module to retain their local interpretations thereby offering a practical tradeoff between complete domain overlap and strict domain disjointness.
- It supports sound distributed reasoning among a federation of loosely coupled ontologies.

Work in progress is aimed at tool support for OWL-based modular ontologies, such as an ontology ontology editor that supports collaborative building of modular OWL ontologies, and a reasoner for distributed reasoning under the proposed OWL semantics.

References

1. Avery, J., Yearwood, J.: Dowl: A dynamic ontology language. In: ICWI. (2003) 985–988
2. Bao, J., Caragea, D., Honavar, V.: On the semantics of linking and importing in modular ontologies. In: I. Cruz et al. (Eds.): ISWC 2006, LNCS 4273. (2006) 72–86
3. Bao, J., Caragea, D., Honavar, V.: A tableau-based federated reasoning algorithm for modular ontologies. In: Accepted by 2006 IEEE/WIC/ACM International Conference on Web Intelligence (In Press). (2006)
4. Bao, J., Honavar, V.: Reasoning with expressive modular ontology languages. Technical report, Computer Sicence, Iowa State University (2006)
5. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: C-owl: Contextualizing ontologies. In: International Semantic Web Conference. (2003) 164–179
6. Grau, B.C., Parsia, B., Sirin, E.: Working with multiple ontologies on the semantic web. In: International Semantic Web Conference. (2004) 620–634
7. Pan, J., Serafini, L.: Semantic import: An approach for partial ontology reuse. In: WoMo 2006, co-located with ISWC. (2006)