On Importing Knowledge from DL Ontologies: some intuitions and problems

Alex Borgida
Dept. of Computer Science,
Rutgers University, New Brunswick, NJ, USA

Abstract. This paper argues for the benefits of distinguishing the notions of "ontology module" and "importing terms from an ontology", by sampling some papers on these topics in the AI and Database communities. It then proposes intuitions and a formal definition for "`importing terms S from KB under rules` $\mathcal{G}$", and looks at the problems of implementing this for very simple kinds of TBoxes.

1 Introduction

There has been considerable recent interest in the notion of "module" in ontologies, including a workshop on this topic at ISWC'06. We wish to consider modules not just as units of development, but also as sources of information used by other ontologies. In this regard, modern programming languages, such as Python provide interesting patterns of use: "`from YourModule import name1 as name2, name3 as name4, ...` ". Such an ability to selectively import ontology fragments will also be beneficial in ontology engineering. For example, the enormous medical ontology ON9.3, developed at CNR in Italy (`http://www.loa-cnr.it/medicine/`), documents each of its theories (modules) with a list of imported terms. Thus, *Anatomy*, which defines 55 classes, specifies in its documentation not just

```
  Theories included by Anatomy:
     Meronymy,  Positions,  Topo-Morphology
```

but also

```
  The following constants were used from included theories:
    * 3d-Area-Of defined as a relation in theory Topo-Morphology
    * > defined as a relation in theory Kif-Numbers
    ... (60+ other terms)
```

and even more interestingly

```
  The following constants were used from theories not included:
    * Anatomical-Abnormality defined as a class in theory Abnormalities
    ... (20+ other terms)
```

Given the decade-long experience of the scientists on the above project, one should not ignore their insight that such specifications are helpful in understanding, developing, and maintaining large ontologies.

To establish some intuitions concerning the desirable properties of "importing terms", we survey a small sample of relevant techniques that have been proposed in the literature. (Many additional papers are omitted for lack of space.)

2 Previous Approaches to Knowledge Import

A variety of papers provide more subtle approaches than importing entire ontology files, as in OWL. The first two categories (a-b) below, rely on (semi-automatically) fragmenting an ontology into modules, and then importing only relevant modules. The last two (c-d) directly address importing individual terms.

(a) Logical Specification of Modules A *logical module* KB1 of a theory KB is required to be *locally sound* (if KB1 $\models\psi$ then KB $\models\psi$) and *locally complete* (if KB $\models\psi$ for a formula ψ that uses only symbols from vocab(KB1), then in fact also KB1 $\models\psi$). Cuenca Grau et al [4] extend this idea, by requiring that **module**(N,KB) — the module of name N in KB, also be "a coherent and self-contained *subset* of KB" (which in this case is a description logic TBox). As such, it should contain N's subsuming and subsumed concept names in KB, and ensure "self-containment". A more recent proposal, "minimal S-modules" [6], will be reviewed later.

(b) (Automatic) Graphical Segmentation of Modules. Seidenberg and Rector [10] suggest that **module**(N,KB) start out with axioms specifying the (1) subclasses of N, (2) super-classes of N, (3) restrictions on the roles of N, and (4) super-roles of N in KB. One then repeatedly adds new identifiers and axioms according to steps 2-4 above, until a fixed point is reached. If one were to draw a graph G_{KB} with concept names as nodes connected by edges representing role restrictions or IsA relationships, then this can be described as a simple graph traversal algorithm. To reduce a large module, [10] allow limiting the depth of the traversal, resulting in "dangling" *boundary classes.*

(c) Importing Terms by Ontology Winnowing. A surprising number of papers argue for the development of *domain-specific* ontologies by reusing *fragments of generic, top-level ontologies* such as Cyc, WordNet, etc. In such cases, the portion of the top-level ontology KB to be imported is influenced by a set of "seed concepts" S, that are to be re-used in the domain-specific ontology. The key to each such technique are the principles which *automatically* derive the axioms and possibly additional concepts to be imported.

For example, Navigli [9] starts with WordNet, whose concepts are organized by hyponym subsumption. The elements of S are concepts corresponding to the roots of local ontology trees for domain specific terms. The algorithm first eliminates concepts not on a path from the top of the WordNet hierarchy to some element of S, in a "pruning phase"; it then eliminates, as uninteresting, concepts with only one child in the hierarchy left, in a "trimming phase". As a result, **import**(S,KB) is a taxonomy where every node has at least two children, so that long chains of uninteresting subsumptions are not present.

Conesa and Olive [2] elaborate Navigli's technique, to build database conceptual schemas by starting from OpenCyc as KB. The paper describes **import**(S,KB) as a minimal subset of KB whose vocabulary contains S and its superclasses, and the algorithm may eliminate concepts between the topmost classes in S and the root of the taxonomy in KB, as well as classes that only provide "redundant inheritance paths".

Note that in all proposals in this category the imported concept names are restricted to be used in the importing KB according to the following simple grammar $\mathcal{G}_{winnow}$ for TBox axioms

```
<TBox axiom> ::= <local axiom> | <connect up axiom>
<local axiom> ::= <local DL concept> ⊑ <local DL concept>
```

```
<connect up axiom> ::= <local DL concept> ⊑  <Imported concept identifier>
<local DL concept> ::= ...
```

(d) Importing Terms in Local-Model Semantics

The theory of binary $\mathcal{E}$-connections between description logics has been used in [3] to connect DL ontologies KB_1 and KB_2 (which are interpreted in disjoint "local" domains) through a number of binary relations ("links") p_k between objects in these interpretations. The result is that in KB_1 one can now construct concepts by restricting p_k with terms C_2 from KB_2, such as $\forall p_k.C_2$. (In KB_2, one can use p_k^-.) This is like importing concepts from KB_2 into KB_1, but restricting their syntactic occurrence to express value restrictions on the roles p_k — something that can obviously be expressed by another grammar, $\mathcal{G}_{\mathcal{E}}$, for subsumption axioms involving imported concepts.

Distributed Description Logics (DDL) [1] use "bridge rules" with approximately the meaning 1:A ⊑ 2:B to relate concepts A and B from ontologies KB_1 and KB_2 respectively. Such a rule can also be viewed as importing concept B into KB_1 and then highly restricting its syntactic occurence in an axiom.

3 Importing S: intuitions and definition

From the Anatomy example in Section 1, we start by using syntactic expressions of the form "`import S from` KB_{expt}" for **import**(S,KB_{expt}), where S is a set of identifiers $\{N_1, N_2, \ldots\}$ contained in vocab(KB_{expt}). For simplicity, when some KB_{impt} uses axioms relating the symbols in S to its own local identifiers, we assume that vocab(KB_{impt}) ∩ vocab(KB_{expt}) ⊆ S.

Based on the preceding survey, we take it that the purpose of importing some set of identifiers S and their related axioms from ontology KB, as opposed to including the entire KB as a file, is *to minimize the material* **import***(S,KB) required to understand S, in order to facilitate comprehension by humans, and possibly to help with local caching.* This philosophy is most evident above in the ontology winnowing work, but also appears in the work on automatic ontology modularization.

There is some sentiment that **import**(S,KB) should be a subset of KB, rather than its theorems [4–6]. This means that the syntactic presentation of axioms in KB is taken to matter, presumably since it helps humans understand the problem domain. We shall modify this requirement somewhat to say that *explanations* of reasoning in **import**(S,KB) should correspond to explanations in KB. The main reason for switching to this alternate requirement is that, as argued in [7], explanations need not always be complete logical proofs, since some obvious steps may be omitted. One example of this is simple inheritance: chaining of IsA in a classification hierarchy of primitives. For example, if KB contains {Dog :< Canine, Canine :< Animal} and S={Dog,Animal} then it should be sufficient to import {Dog :< Animal}[1]. Note that Navigli's proposal [9] omits exactly such kinds of trivial steps.

[1] Contrary to standard practice, we will use A:< B to indicate an axiom in the theory, and A⊑ B to indicate the subsumption judgment, entailed or proven in a theory.

Another implication of the need for explanations is that **import**(S,KB) may have to contain symbols other than those in S[2]. For example, if KB= {Married $\equiv$ Person $\sqcap \geq 1.spouse$, Unmarried $\equiv$ Person $\sqcap \leq 0.spouse$}, and S={Married, Unmarried}, then, to explain why they are disjoint, we will want **import**(S,KB) to contain {Married :< $\geq 1.spouse$, Unmarried :< $\leq 0.spouse$}. One could also make a case that the actual definitions should be included, since users of the term should appreciate that these are defined, as opposed to primitive concepts. A compromise might be to allow for definitions with ellipses: {Married $\equiv \ldots \sqcap \geq 1.spouse$, Unmarried $\equiv \ldots \sqcap \leq 0.spouse$}.

Once we admit the need for seeing additional symbols from vocab(KB), other than those in S, the question arises whether such symbols should become part of S, allowing the importer to use them in constructing new concepts/axioms. We suggest that this should *not* be the case, since the user has specified S as the set of concepts (s)he will be using. Therefore we will keep the set S unchanged, and consider vocab(**import**(S,KB))−S to be boundary concepts, used only in explanations. In line with our desire to reduce the need to understand all of KB, the set of such additional concepts should however be minimized.

From a logical point of view, we will obviously want **import**(S,KB) to be locally sound. We will not however insist on full "local completeness". The reason for this is that we have seen in both the work on upper-ontology prunning and local-model semantics that the imported identifiers might only be used in a limited way in KB_{impt}. (The use of external symbols is also limited in [5], in order to guarantee the desired property of "conservative extension".) For example, if imported names N_i can only appear in axioms of the form $\alpha :< N_i$, as per $\mathcal{G}_{winnow}$, then it might not matter whether **import**(S,KB) $\models \neg N_1 \sqsubseteq N_2$, since the syntax does not allow asking such questions directly of K3 = $\mathrm{KB}_{impt} \cup$ **import**(S,KB), and knowledge of this fact might not affect inferences from K3 in certain DLs. (See Sec. 4 for examples.) The same might happen if the importing ontology uses a different, weaker logical language than the exporting one. The limited use of imported concept identifiers can then be exploited to decrease the set of concepts and axioms from KB that need to be included in **import**(S,KB).

The `import` syntax should therefore reflect rules about the use of symbols in S in the importing ontology. An instruction of the form "`KB`$_{impt}$ `imports S from KB`$_{expt}$" would however seem to be too specific, since the material imported might change as the local ontology evolves. For this reason, we suggest characterizing the use of imported identifiers using a *grammar* $\mathcal{G}$, as we have done in (c) and (d) above.

To formalize the above discussion for the case of DL TBoxes, we assume that every description logic $\mathcal{D}$ provides, as usual, a syntax for the concepts and roles, as well as axioms allowed in a TBox, plus a semantic entailment relationship $\models$ of the logic for various kinds of judgements ψ, such as subsumption. In addition, we also need a specification $\mathcal{XPL}$ of acceptable explanations for judgements in the logic[3]. As part of $\mathcal{XPL}$, we assume that every DL $\mathcal{D}$ provides

[2] Of course, this is an integral part of all the proposals surveyed in Section 2.
[3] Normally, such a specification is based on a proof theory for the logic, e.g., [7].

an operator $expand_{\mathcal{D}}$(KB), which may add some redundant axioms to KB to avoid unnecessarily long explanations. For example[4], $expand()$ may collapse inheritance according to

$inherit$(KB)={ A :< α | A :< B_0 :< ... :< B_n :< α in KB}.

The following definition then summarizes the above intuitions

Definition 1. *Given (1) an (exporting) TBox KB of description logic $\mathcal{D}_1$, (2) a set of concept names $S \subseteq$ vocab(KB) to be imported, (3) a description logic $\mathcal{D}_2$ for importing Tboxes, and (4) a grammar $\mathcal{G}$ specifying the syntax of axioms in importing Tboxes, including the occurence of identifiers from S: We seek a minimal set of identifiers $\tilde{S}$ containing S, and a minimal subset K of axioms from expand(KB) involving only names from $\tilde{S}$ such that for every KB_{impt} satisfying $\mathcal{G}$, and every judgement ψ with vocab(ψ) $\subseteq$ vocab(KB_{impt}) $\cup$ S , we have that $KB_{impt} \cup$ KB $\models \psi$ iff $KB_{impt} \cup K \models \psi$, with all explanations in the latter being valid in the former.*

Such a TBox K will be referred to as $\mathbf{import}_{\mathcal{G}}$*(S,KB).* □

Note that such a K is guaranteed to exist, since one can start with $\tilde{S}$=vocab(KB), and K=KB, and then minimize from there. Of course, K may not be unique.

Independently, in a soon-to-appear paper [6], Cuenca Grau et al have proposed a definition for the notion of "minimal S-module", which can be viewed as a special case of the above, where $\mathcal{D}1 = \mathcal{D}2$, $\mathcal{G}$ does not officially restrict the occurence of imported names (though a sufficient syntactic test for it is proposed), explanations play no role, and the above-stated conditions must hold for *every* $\mathcal{D}$ with Tarski-style set-theoretic semantics. Please note that our more general definition was motivated by actual proposals in the prior literature (categories (c) and (d)), rather than just our own intuitions. The above cited paper, as well as [5] contain discussions concerning the related notion of "conservative extension".

4 Computing import in some simple cases

We propose to explore some computational consequences of the above definition. Because of the observation concerning the work in [6], their negative complexity results (e.g., undecidability for $\mathcal{ALCO}$) immediately transfer to our case. So, rather than jumping to consider very expressive DLs, we propose to see how various characteristics of DLs (e.g., primitive vs. defined concepts, ability to specify inconsistent concepts, role restrictions) interact with the above definition in the absence of other sources of complexity. Hence we restrict ourselves to "weak" DLs. In fact, $\mathcal{D}_{import}$ will be taken to be only atomic concepts; and in $\mathcal{D}_{export}$ subsumption will be determinable by a proof theory consisting of normalization rules followed by structural subsumption rules. As in [7], explanations of $\alpha \sqsubseteq \beta$ are provided by (i) decomposing β into a conjunction of "atomic descriptions" β_j[5], (ii) computing $normalize(\alpha)$, and (iii) then showing how $normalize(\alpha) \sqsubseteq \beta_j$

[4] Below, we will use A,B,C, ... to denote concept identifiers, and Greek letters α, β, ... to denote possibly complex concept expressions.

[5] An atomic description β_j cannot be expressed as the conjunction of two or more descriptions, each of which is smaller in size.

for each j. The decomposition of β into atomic description is not usually explained, since it is rather trivial.

We also restrict $\mathcal{G}$ throughout to very simple "top-level ontology" style imports, allowing B∈S to only appear in axioms A:< B for A∉S. (But note that axioms {F :< C, F :< B} will give the effect of conjoining elements of S in KB_{impt}!)

Throughout, we allow KB_{expt} to have axioms of the form A :< α, providing necessary conditions for primitive concepts, A. But we forbid recursion in axioms.

4.1 Conjunction in Necessary Conditions

For this, we decompose axioms involving conjunction into ones without them. Thus, if KB contained F:< (A ⊓G), but all we needed for a proof is F:< A, we will avoid the trivial steps of going from F:< (A ⊓G) to F:< A. For this purpose, we use an operator $expand_{\sqcap}()$ defined as

$expand_{\sqcap}(\alpha_1 \sqcap \ldots \sqcap \alpha_n) = \{\alpha_1, \ldots, \alpha_n\}$
$expand_{\sqcap}(\alpha{:<}\beta) = \{\alpha{:<}\gamma \mid \gamma \in expand_{\sqcap}(\beta)\}$
$expand_{\sqcap}(KB) = \{expand_{\sqcap}(\alpha{:<}\beta) \mid \alpha{:<}\beta \in KB\}$
and call the fixed point of this operator $expand_{\sqcap}{}^*()$.

Subsumption reasoning in $KB_{impt}\cup$ $expand_{\sqcap}{}^*$(KB) now consists solely of transitive chaining of axioms, which is abbreviated by *inherit*(). This yields

import(S,KB) = $reduce(select(S, inherit(expand_{\sqcap}{}^*(KB))))$

where $select(V, KB) = \{\psi \in KB \mid vocab(\psi) \subseteq V\}$ and reduce( ) removes redundant axioms — in this case, redundancy introduced earlier by *inherit*().

The complexity of this computation is clearly polynomial.

4.2 Disjoint Concepts and Necessary Conditions

If the exporting DL now also has atomic negation, say, one can specify disjoint concepts B and C, which means that if KB_{impt} has axioms {F:< B, F:< C}, then we must be able to conclude that F⊑ ⊥, in addition to *inherit*.

This may require considering identifiers, A, not in S, as in the case where S={B,C} but KB contains {B :< A, C :< ¬A}.

If we define T_S = {A ∈ vocab(KB) | there exist B,C∈ S, KB⊨B⊑ A, C ⊑ ¬A}, then it is sufficient to also include in **import**(S,KB) for all such A,B and C, axioms {B :< A , C:< ¬A}, testifying to the presence of A in T_S.

In and of itself this is not hard. However, the two concepts B and C may be disjoint for more than one reason: e.g., they may also be subsumed by $\hat{A}$ and $\neg\hat{A}$ respectively. According to our definition, and its intuitions, we should minimize the set of *additional* concepts introduced; hence vocab(**import**(S,KB)) should not contain both A and $\hat{A}$. Unfortunately, this minimization is a combinatorial problem:

Proposition 1. *There are simple KB with axioms of the form C :< D and C' :< ¬D', (C,C'∈S, D,D'∉S), such that the following problem is NP-hard: find the smallest set $W \subseteq vocab(KB) - S$ with the property that for all B,C∈ S: KB ⊨(B ⊓C) ⊑ ⊥ iff select(S∪ W,KB) ⊨(B ⊓C) ⊑ ⊥*

The proof is by reduction from the hitting set problem [Garey & Johnson, SP8].

Similar arguments will hold for any DL that has some way of describing inconsistent concepts, such as number restrictions; they also seem to apply to the minimal S-modules of [6].

4.3 Definitions with Conjunction

The novelty here is that KB can now have axioms of the form $D \equiv A \sqcap B$, as well as $C :< E$. In this case, definitions can no longer be replaced by simple subsumptions on atoms. In fact, if we have S ={A',B',D,H}, KB0 = $\{D \equiv A \sqcap B, A' :< A, B' :< B\}$, while KB_{impt} contains {E :< A', E :< B'}, then $\text{KB}_{impt} \cup$ KB0 $\models$E $\sqsubseteq$ D. Thus **import**(S,KB0) needs to support this inference by importing all of KB0.

Now note that since the conjunction of all concepts in S not subsumed by D ($\delta_D = \sqcap_{C \in S, \text{KB} \not\models C \sqsubseteq D} C$) is the strongest possible condition w.r.t. KB applicable to some concept F in KB_{impt}, if this is not sufficient to entail D (KB $\not\models \delta_D$ $\sqsubseteq$ D) then D's definition need not be considered, and hence can be omitted from **import**(S,DB), since it would never be needed as part of an explanation for why an F is subsumed by D. Therefore it would be sufficient to repeatedly add to **import**(S,KB) definitions for concepts D $\in$ vocab(KB) as long as KB$\models \delta_D$ $\sqsubseteq$ D.

Unfortunately, while the result will include enough axioms, it may include too many. For example, if concepts A and B in KB are subsumed by $\hat{A}$ and $\hat{B}$ individually, as well as n other concepts $C_1, \ldots, C_n$ jointly, then importing the defined concept D $\equiv \hat{A} \sqcap C_1 \sqcap \ldots \sqcap C_n \sqcap \hat{B}$, allows for 2^n possible minimal combinations of axioms to be imported (depending on how the C_i are alloted to A and B). The presence of other concepts and axioms will then tilt in favor of some of these choices.

Proposition 2. *If one allows necessary conditions on definitions, the problem of finding* **import***(S,KB) when KB has conjunctive definitions or axioms of the form $A :< B$, is NP-hard*

Proof is by reduction from the NP-hard problem of minimizing Horn proofs, or minimizing input to monotone boolean circuits. We strongly suspect that the theorem holds even if definitions cannot have necessary conditions.

4.4 Using $\mathcal{FL}^-$

We have seen so far the effect of allowing disjoint concepts and definitions. Let us consider now the other *sine qua non* of DLs, role restrictions.

When considering necessary conditions, restrictions of the form $\exists p.\top$ are treated as atomic, while nested $\forall$-restrictions need to be separated into atomic descriptions, which do not involve conjunction. For this purpose, extend $expand_{\sqcap}()$ as follows:

$expand_{\sqcap}(\forall p.\beta) = \{\forall p.\gamma \mid \gamma \in expand_{\sqcap}(\beta)\}$.

Now $inherit(expand_{\sqcap}{}^*(\text{KB}))$ again contains axioms abbreviating chains of subsumption from a concept A in S to atomic descriptions β_i appearing on the right hand side of axioms in S. **import**(S,KB) can now be computed as in 4.1 above.

As illustrated above, the general pattern for adding new constructors for DLs with structural subsumption seems to be to extend the notion of atomic description and $expand_{\sqcap}()$ so that $inherit(expand_{\sqcap}(\text{KB}))$ contains the axioms needed to find the normal forms of concepts in S, and detect conjunctions that can lead to $\perp$. One is then faced with a minimization problem for deciding which new identifiers and axioms to include, and this is likely to be difficult to solve precisely.

5 Conclusions

Starting from a sample of works on ontology modularization and reuse, we have argued for a set of desirable properties for the notion of *"KB_1 imports terms S from KB_2",* distinguishing this from the problem of ontology modularization by: allowing restrictions on the place where imported names can be used, and requiring both minimization of material imported and preservation of explanations — all properties motivated by prior examples of importing studied in the literature. We then investigated the difficulties encountered with implementing the corresponding formal definition in the case of TBoxes that use simple DLs, where subsumption itself is easy. Perhaps not surprisingly, attempts to *minimize* the set of axioms imported leads to combinatorial difficulties. It remains to be seen if the definition can be modified in a motivated manner (e.g., importing should provide *all* explanations in the exporting KB) and if approximate solutions to NP-hard problems would help. Forthcoming work with Fausto Giunchiglia will apply this framework to UML.

Acknowledgements. I am greatful to Fausto Giunchiglia for discussions starting down this path, and to the referees for pointing out unclear aspects. This work was supported in part by the U.S. DHS under ONR grant N00014-07-1-0150.

References

1. Alex Borgida, Luciano Serafini: Distributed Description Logics — Assimilating Information from Peer Sources. J. Data Semantics 1: 153-184 (2003)
2. Jorge Conesa, Antoni Olive: A Method for Pruning Ontologies in the Development of Conceptual Schemas of Information Systems. J. Data Semantics V: 64-90 (2006)
3. Bernardo Cuenca Grau, Bijan Parsia, Evren Sirin: Working with Multiple Ontologies on the Semantic Web. International Semantic Web Conference 2004: 620-634
4. Bernardo Cuenca Grau, Bijan Parsia, Evren Sirin, Aditya Kalyanpur: Modularity and Web Ontologies. KR 2006: 198-209
5. Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov and Ulrike Sattler, A Logical Framework for Modularity of Ontologies. IJCAI 2007.
6. Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov and Ulrike Sattler Just the Right Amount: Extracting Modules from Ontologies. WWW 2007.
7. Deborah L. McGuinness, Alexander Borgida: Explaining Subsumption in Description Logics. IJCAI 1995: 816-821
8. Carsten Lutz, Dirk Walther, and Frank Wolter: Conservative Extensions in Expressive Description Logics. IJCAI 2007.
9. Roberto Navigli: Extending, Pruning and Trimming General Purpose Ontologies. 2nd IEEE SMC 2002
10. Julian Seidenberg, Alan L. Rector: Web ontology segmentation: analysis, classification and use. WWW 2006: 13-22