On Importing Knowledge from DL Ontologies: some intuitions and problems

Alex Borgida
Dept. of Computer Science, Rutgers University, 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 both the AI and Database community. It then proposes intuitions and a formal definition for "`importing terms S from KB under rules` $\mathcal{G}$", and looks at some problems in implementing this for very simple kinds of exporting ontologies KB, requiring only structural subsumption.

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 (ONIONS), one should not ignore their intuition that such specifications are helpful in understanding/developing/maintaining large ontologies.

To further refine our intuitions we briefly survey a variety 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 three categories (a-c) below, rely on automatically fragmenting an ontology into modules, and then importing only relevant modules. The last two (d-e) 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) 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": absence of concept names D, E such that D$\in$ vocab(**module**(N,KB)), E$\notin$vocab(**module**(N,KB)), and either KB $\models$D $\sqsubseteq$E or KB $\models$E $\sqsubseteq$D. Impressively, there is an efficient algorithm for partitioning most interesting OWL DL ontologies into such modules.

(b) Automatic Graphical Segmentation of Modules. Based on experience with building large biomedical ontologies, for which technique (a) above is unsuccessful, Seidenberg and Rector [8] suggest that **module**(N,KB) start out with axioms specifying the (1) subclasses of N in KB, (2) super-classes of N in KB, (3) restrictions on the roles of N in KB, 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 restrictions or IsA relationships, then this can be described as a simple graph traversal algorithm. To reduce a large module, [8] allow disregarding some roles, and limiting the depth of the traversal. Concept names whose roles are not included are called boundary classes. Note that several aspects of this empirically-based proposal contravene the "self-containment" requirement of (a) above.

(c) Ontology Modules as View Specifications. Noy and Musen [6] exemplify the case where the domain expert is given the opportunity to specify explicitly the subset of the ontology which is to be imported. In this case, the tool offered is a language for giving directives for traversing the graph of the ontology, particularly the length of the paths to be followed along different kinds of relationships. As in [8], boundary concepts show up at the end of such paths.

(d) 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 additional concepts and axioms in the sub-ontoly **import**(S,KB) to be imported.

For example, Navigli [7] 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 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 in fact describes **import**(S,KB) as a minimal subset of KB whose vocabulary contains S and its superclasses, and its 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".

(e) Importing Terms in the Context of Local Model Semantics

Distributed Description Logics [1] use "bridge rules" of roughly the form i:A $\sqsubseteq$j:B to relate concepts A and B from ontologies KB_i and KB_j respectively. Inside KB_k, one is however restricted to using subsumption axioms involving only symbols prefixed by k: . Here, a bridge rule like i:A $\sqsubseteq$j:B can be viewed as "importing concept B from ontology KB_j", and then highly restricting its use in KB_i to only top-level subsumption axioms.

The theory of binary $\mathcal{E}$-connections between description logics has been used in [3] to connect ontologies KB_1 and KB_2, which are again interpreted in independent 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 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.

Finally, Stuckenschmidt & Klein [9] combine the semantics used in DDL with a view-based module definition and add as a novel feature the local compilation/caching of reasoning from the exporting module.

3 The intuitions of importing

We start by using syntactic expressions of the form "`import S from KB`$_{expt}$", where S is a set of identifiers $\{\mathrm{N}_1, \mathrm{N}_2, \ldots\}$ contained in vocab(KB_{expt}). When some ONT_{impt} uses axioms relating the symbols in S to its own local identifiers, we assume that vocab(ONT_{impt}) $\cap$ vocab(KB_{expt}) $\subseteq$ S.

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, and ontology view specification.

From a logical point of view, we will want **import**(S,KB) to be "locally sound". We will not however want full "local completeness". The reason for this is that we have seen in both the work on upper-ontology pruning, and especially local-model semantics that the concepts in **import**(S,KB) might only be used in a limited way in ONT_{impt}. Independently, this might be the case if the importing ontology uses a different, weaker logical language than the exporting one. This limited use of imported concepts 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 S in the importing ontology. An instruction of the form "$\texttt{ONT}_{impt}$ `imports S from` $\texttt{KB}_{expt}$" 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 importing ontology and its use of imported symbols using a *grammar* $\mathcal{G}$.

There is a considerable agreement that **import**(S,KB) should be a subset of KB, rather than its theorem. This means that the syntactic presentation of axioms in KB is taken to matter, presumably since it helps humans understand the 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 [5], 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. So, 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 [7] omits exactly such trivial steps.

Another implication of the need for explanations is that **import**(S,KB) may have to contain symbols other than those in S. For example, if KB= {Married $\equiv$ Person $\sqcap \geq 1.\mathit{spouse}$, Unmarried $\equiv$ Person $\sqcap \leq 0.\mathit{spouse}$}, and S={Married, Unmarried}, then, to explain why they are disjoint, we will want **import**(S,KB) to contain {Married :< $\geq 1.\mathit{spouse}$, Unmarried :< $\leq 0.\mathit{spouse}$}. 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 $\equiv \ldots \sqcap \geq 1.\mathit{spouse}$, Unmarried $\equiv \ldots \sqcap \leq 0.\mathit{spouse}$}.

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

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

of S, allowing the importer to use them in concept/axiom construction. 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. So we will keep the set S unchanged, and consider vocab(**import**(S,KB))$-$S to be boundary concepts used only in explanations. Note however that in line with our desire to reduce the need to understand all of KB, the set of such additional concepts should be minimized.

4 Importing from DL TBoxes

Henceforth, we will let both ONT_{impt} and KB be DL TBoxes, S be a set of concept identifiers, and explore some computational consequences of the above intuitions. We will use A,B,C,... to denote concept names, and Greek letters such as α, β to denote possibly complex concept expressions.

We assume here that KB contains at least axioms of the form A $:<\alpha$, describing *primitive concepts* A, and possibly axioms of the form B$\equiv\beta$, introducing *defined concepts* B. We do not allow here recursive concepts.

We also assume an operator *expand*(KB), which adds some axioms to KB to avoid unnecessarily long explanations. For example, it may collapse inheritance according to *inherit*(KB)=$\{$ A $:<\alpha$ $\mid$ A $:<$ B_0 $:<\ldots:<$ B_n $:<\alpha$ in KB$\}$

Gathering the intuitions from above, we suggest the following

Definition. *Given (i) an (exporting) TBox KB, (ii) a set of concept names $S\subseteq$ vocab(KB) to be imported, (iii) grammatical rules $\mathcal{G}$ specifying the form of all possible importing knowledge bases ONT_{impt} (i.e., a set of concept constructors and kinds of axioms allowed), which may provide for different ways in which symbols from S can be used in constructing concepts and axioms in ONT_{impt}.*

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 ONT_{impt} satisfying $\mathcal{G}$, and concepts α and β with vocab(α), vocab(β) $\subseteq$ vocab(ONT_{impt}), we have that $ONT_{impt}\cup KB \models\alpha\sqsubseteq\beta$ iff $ONT_{impt}\cup K\models\alpha\sqsubseteq\beta$ with all explanations in the latter being valid in the former.

Such an ontology 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 $\tilde{\mathrm{S}}$=vocab(KB), and K=KB, and then minimize from there. Of course, K may not be unique.

To examine some of the properties of this definition, we limit ourselves here to very simple situations, where there are no other sources of complexity: (1) $\mathcal{G}$ only allows B$\in$S to appear in axioms like A$:<$B for A$\notin$S.[2] (2) In the DLs used in KB, subsumption must be determinable efficiently by a proof theory consisting of normalization rules followed by structural subsumption rules. Following [5], explanations of $\alpha\sqsubseteq\beta$ are provided by (i) decomposing β into a conjunction of

[2]But axioms $\{$F $:<$ C, F $:<$ B$\}$ will give the effect of conjoining elements of S.

"atomic descriptions" β_j[3], (ii) computing *normalize*(α), and (iii) then showing how normalize$(\alpha) \sqsubseteq \beta_j$ for each j. The decomposition of β into atomic description is not usually explained, since it is rather trivial.

4.1 Primitive concepts, with conjunction in axioms

To reason with conjunction in necessary conditions about primitive concepts we decompose axioms involving conjunction into ones without them, so that if KB contained F:< (A ⊓G), but all we needed for a proof is just F:< A, we would avoid the trivial steps of going from F:< (A ⊓G) to F:< A. For this, we provide an operator $expand_\sqcap()$
$expand_\sqcap(\alpha_1 \sqcap \ldots \sqcap \alpha_n) = \{\alpha_1, \ldots, \alpha_n\}$
$expand_\sqcap(\alpha{:<}\,\beta) = \{\alpha{:<}\,\gamma \mid \gamma \in expand_\sqcap(\beta)\}$
$expand_\sqcap(KB) = \{expand_\sqcap(\alpha{:<}\,\beta) \mid \alpha{:<}\,\beta \in KB\}$
and call the fixed point of this operator $expand_\sqcap{}^*()$.

Subsumption reasoning in $\mathrm{ONT}_{impt} \cup$ $expand_\sqcap{}^*$(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 Primitive concepts, with $\mathcal{AN}$ (conjunction, atomic negation, number restrictions). We can now specify disjoint concepts, whose conjunction is subsumed by everything. One must therefore consider axioms involving symbols not in S since some F in ONT_{impt} might subsumed by both B and C in S, where in turn, KB contains B :< A, C :< ¬A for some A ∉ S.

If we define the set 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) axioms B :< A and C:< ¬A testifying to the presence of A in T_S. Similarly, we can define sets T_S^n = { $\geq n.p$ ∈ subterms(KB) | there exist B,C∈ S, KB⊨B⊑ $\geq n.p$, C ⊑ $\leq (n-1).p$ }, dealing with conflicting number restrictions.

However, 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 4.1. *There are simple KB with axioms of the form B :< A and B' :< ¬A', (B,B'∈S, A,A'∉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 hitting set problem [Garey & Johnson, SP8].

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

4.3 Primitives and definitions, with conjunction in axioms

The novelty here is that KB can now have axioms of the form $D\equiv A\sqcap B$, as well as {A' :< A, B' :< B}, and these definitions can no longer be replaced by simple subsumptions on atoms. In fact, if we have S ={A',B',D,H}, and ONT_{impt} contains {E :< A', E :< B'}, then $\mathrm{ONT}_{impt}\cup$ KB $\models$E $\sqsubseteq$D, so **import**(S,KB) needs to support this inference by also containing the axiom $D\equiv A\sqcap B$. On the other hand, note that if in KB we have H $\equiv$A $\sqcap$Y, then there is no reason to add this definition to **import**(S,KB), since it could not possibly have been used in inferring a relationship involving concepts in ONT_{impt}.

There is a relatively simple test whether D $\in$ S need *not* have its definition expanded: check if the concept $\delta_D = \sqcap_{C\in S,\mathrm{KB}\not\models C\sqsubseteq D} C$ has the property that KB $\not\models$ δ_D $\sqsubseteq$D. An algorithm could then repeatedly add definitions for concepts D as long as KB$\models$ δ_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 4.2. *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 Primitives and definitions, using $\mathcal{FL}^-$

There is actually no need to import additional information from KB in order to reason about axioms for primitive concepts in $\mathcal{FL}^-$, since their conjunction cannot be inconsistent.

To deal with definitions, restrictions of the form $\exists p.\top$ are treated as primitive names, while nested $\forall$-restrictions need to be separated into atomic descriptions, which do not involve conjunction. For this purpose, extend $expand_\sqcap()$

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

$expand_\sqcap(D\equiv\beta) = \{D\equiv \sqcap_{\beta_i\in expand_\sqcap(\beta)}\beta_i\}$

Now $inherit(expand_\sqcap{}^*(\mathrm{KB}))$ contains axioms abbreviating chains of subsumption from a concept A in S to atomic descriptions $beta_i$ appearing on the right hand side of definitions in S. As before, minimizing the set of such axioms is a combinatorial problem.

As illustrated above, the general pattern for any new DL is to extend the notion of atomic description and $expand_\sqcap()$ so $inherit(expand_\sqcap(\mathrm{KB}))$ contains the axioms needed to find the normal forms of concepts in S, and detect conjunctions that can lead to ⊥. 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. We have 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.

References

[1] A. Borgida, L. Serafini: Distributed Description Logics: Assimilating Information from Peer Sources. J. Data Semantics 1: 153-184 (2003)

[2] J. Conesa, A. 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, B. Parsia, E. Sirin: Working with Multiple Ontologies on the Semantic Web. International Semantic Web Conference 2004: 620-634

[4] Bernardo Cuenca Grau, B. Parsia, E. Sirin, Aditya Kalyanpur: Modularity and Web Ontologies. KR 2006: 198-209

[5] Deborah L. McGuinness, Alexander Borgida: Explaining Subsumption in Description Logics. IJCAI 1995: 816-821

[6] Natalya Fridman Noy, Mark A. Musen: Specifying Ontology Views by Traversal. International Semantic Web Conference 2004: 713-725

[7] Roberto Navigli: Extending, Pruning and Trimming General Purpose Ontologies. 2nd IEEE SMC 2002

[8] Julian Seidenberg, Alan L. Rector: Web ontology segmentation: analysis, classification and use. WWW 2006: 13-22

[9] Heiner Stuckenschmidt, Michel C. A. Klein: Integrity and Change in Modular Ontologies. IJCAI 2003: 900-908