DeMoSt: a Tool for Exploring the Decomposition and the Modular Structure of OWL Ontologies

Chiara Del Vescovo[1], Pavel Klinov[2], Bijan Parsia[1],
Ulrike Sattler[1], and Thomas Schneider[3]

[1] University of Manchester, UK
{delvescc|bparsia|sattler}@cs.man.ac.uk
[2] University of Arizona, AZ, USA
pklinov@email.arizona.edu
[3] Universität Bremen, Germany
tschneider@informatik.uni-bremen.de

Motivation In relevant application fields ontologies are often maintained as huge monolithic collections of axioms in single files, as for some ontologies in the NCBO BioPortal repository[4], like the Gene Ontology ($\sim 60,000$ axioms). Such representation is not ideal for applications which only require access limited to individual fragments of the ontology. As an example, the reasoning service provided needs just a small part of it to be performed, and loading only the necessary bit to the task can reduce substantially time and memory required, improving the performance of the system. Hence it is important to investigate the possibility of maintaining ontologies in a more flexible form.

Recently the notion of *module* has been developed. Modules are subsets of ontologies that preserve all entailments over a set of terms Σ called *seed signature*, and they come in many flavours. We focus on modules based on *syntactic locality* [2] (from now on simply called modules), that are efficiently computable.

Modules are useful and used, for example in reuse scenarios, i.e. when a part of a well-established ontology is selected to be reused in a different ontology. However, it is really important to exactly know the seed signature needed, or a single extraction can generate undesired results, as, for example, a module that does not contain all axioms needed, or one that contains too many of them. Hence it is interesting to discover the relations between the extracted module with its super- and sub-modules, because a slightly different module could be more interesting than the one extracted. In order to look at these (logic-based) relationships, a succint representation, called *Atomic Decomposition*, of the modular structure of an ontology has been introduced in [5].

In our paper [3], accepted for the Research Track at ISWC2011, we present the first, to our knowledge, large-scale investigation into decomposability and modular aspects of the NCBO BioPortal ontologies. We consider the Atomic Decompositions of such ontologies and demonstrate that most of them split into *small* logically coherent parts (atoms), from which modules can be efficiently assembled during reasoning. We discuss such good (on average) decomposability of BioPortal ontologies, its implications for applications, and also comment on

[4] http://bioportal.bioontology.org/ontologies

occasional poor decomposability. Moreover, we describe a novel algorithm for decomposition-based module extraction (and the auxiliary algorithm for computing minimal seed signatures) and present evaluation results.

The theoretical foundation of Atomic Decomposition is quite abstract, so we intend to complete the presentation of our work by introducing a demo. The prototype described here, called DeMoSt (Decomposition & Modular Structure) is designed to enable an ontology engineer to analyse the AD of their ontology. During our demo, we plan to showcase DeMoSt and perform "on the fly" decompositions of ontologies, possibly selected by the interested audience.

Preliminaries We assume the reader to be familiar with OWL and the underlying Description Logics [1], and only briefly sketch here some of the central notions around locality-based modularity [2] and Atomic Decompositions [5,3]. We use $\mathcal{O}$ for ontologies, i.e. finite sets of axioms based on a Description Logic, e.g., $\mathcal{SHIQ}$, and $\mathcal{M}$ ($\subseteq \mathcal{O}$) for modules. Moreover, we use $\widetilde{\alpha}$ for the signature of an axiom α, i.e., the set of class, property, and individual names used in α.

Given a set of terms, or seed signature, Σ, a Σ-module $\mathcal{M}$ based on deductive-Conservative Extensions is a minimal subset of an ontology $\mathcal{O}$ such that for all axioms α with terms only from Σ, we have that $\mathcal{M} \models \alpha$ iff $\mathcal{O} \models \alpha$, i.e., $\mathcal{O}$ and $\mathcal{M}$ have the same entailments over Σ. Unfortunately, deciding if a set of axioms is a module in this sense is hard or even impossible for expressive DLs [6,8]. However, if we drop the minimality requirement we can define "good sized" approximations that are efficiently computable, as in the case of *syntactic locality*, which modules are extracted in polynomial time. We are not going into details and refer to [2] for a definition, but please note that such modules provide strong logical guarantees by capturing *all* the relevant entailments about Σ, although not necessarily *only* those [7]. A module extractor is implemented in the OWL API.[5]

Locality-based modules come in 3 flavours, namely $\top, \bot$, and $\top\bot^*$: roughly speaking, a $\top$-module for Σ gives a view "from above" because it contains all subclasses of class names in Σ; a $\bot$-module for Σ gives a view "from below" since it contains all superclasses of class names in Σ; and $\top\bot^*$-modules are the smallest, contained in both the corresponding $\top$- and $\bot$-module, containing all entailments to imply that two classes in Σ are in the subclass relation, but not necessarily their sub- or super-classes. Given a module notion $x \in \{\top, \bot, \top\bot^*\}$, we denote by x-mod$(\Sigma, \mathcal{O})$ the x-module of $\mathcal{O}$ w.r.t. Σ.

The modules of an ontology $\mathcal{O}$ can be exponential in the size of $\mathcal{O}$ [4]. However we can focus only on *genuine* modules, i.e. modules that are not the union of two "$\subseteq$"-uncomparable modules. Such modules define a base for all modules, and remarkably the family of genuine modules is linear in the size of $\mathcal{O}$ [5].

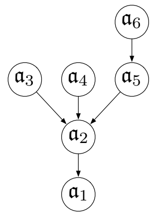
Some sets of axioms never split across two modules [5], revealing a strong logical interrelation. For $x \in \{\top, \bot, \top\bot^*\}$, we call *$x$-atom* a maximal set $\mathfrak{a}^x \subseteq \mathcal{O}$ such that, for each x-module $\mathcal{M}^x$, either $\mathfrak{a}^x \subseteq \mathcal{M}^x$, or $\mathfrak{a}^x \cap \mathcal{M}^x = \emptyset$. The family of x-atoms of $\mathcal{O}$ is denoted by $\mathcal{A}(\mathfrak{F}^x_{\mathcal{O}})$ and is called *x-Atomic Decomposition (x-AD)*. If x is clear from the context, or irrelevant, we drop it.

[5] `http://owlapi.sourceforge.net`

Since every atom is a set of axioms, and atoms are pairwise disjoint, the AD is a partition of the ontology, and its size is at most linear w.r.t. the size of the ontology. Moreover, there is a 1-1 correspondence between atoms and genuine modules: for each atom $\mathfrak{a}$ we denote by $\mathcal{M}_{\mathfrak{a}}$ the corresponding genuine module, that is also the smallest module containing $\mathfrak{a}$. Then we can define a second logical relation between atoms: an atom $\mathfrak{a}$ is *dependent* on a distinct atom $\mathfrak{b}$ (written $\mathfrak{a} \succeq \mathfrak{b}$) if $\mathcal{M}_{\mathfrak{b}} \subseteq \mathcal{M}_{\mathfrak{a}}$. Note that this property then holds for all modules containing $\mathfrak{a}$. The dependence relation $\succeq$ on AD is a partial order (i.e., transitive, reflexive, and antisymmetric), thus can be represented by means of a Hasse diagram. Moreover, it is computable in polynomial time [5].

Atoms can be seen as building blocks for modules: for each x-module $\mathcal{M}$ of an ontology $\mathcal{O}$, there are atoms $\mathfrak{a}_1, \ldots, \mathfrak{a}_\kappa$ in $\mathcal{A}(\mathfrak{F}^x_{\mathcal{O}})$ such that $\mathcal{M} = \bigcup_{i=1}^n \mathfrak{a}_i$. The converse does not hold, since not all combinations of atoms are modules. However, an algorithm to perform the extraction of a module from the AD of an ontology is described in [3].

Example 1. Consider the following ontology and its $\bot$-AD:
$\mathcal{O} = \{\alpha_1 =$ '$\texttt{Animal} \sqsubseteq \exists\texttt{hasGender.Thing}$',
$\alpha_2 =$ '$\texttt{Animal} \sqsubseteq \geq 1\texttt{hasHabitat.Thing}$',
$\alpha_3 =$ '$\texttt{Person} \sqsubseteq \texttt{Animal}$',
$\alpha_4 =$ '$\texttt{Vegan} \equiv \texttt{Person} \sqcap \forall\texttt{eats.}(\texttt{Vegetable} \sqcup \texttt{Mushroom})$',
$\alpha_5 =$ '$\texttt{TeeTotaller} \equiv \texttt{Person} \sqcap \forall\texttt{drinks.NonAlcoholicThing}$',
$\alpha_6 =$ '$\texttt{Student} \sqsubseteq \texttt{Person} \sqcap \exists\texttt{hasHabitat.University}$',
$\alpha_7 =$ '$\texttt{GraduateStudent} \equiv \texttt{Student} \sqcap \exists\texttt{hasDegree.}(\{\text{BA, BS}\})$'$\}$



Here the $\bot$-atoms in the AD contain the following axioms respectively: $\mathfrak{a}_1 = \{\alpha_1, \alpha_2\}$, $\mathfrak{a}_2 = \{\alpha_3\}$, $\mathfrak{a}_3 = \{\alpha_4\}$, $\mathfrak{a}_4 = \{\alpha_5\}$, $\mathfrak{a}_5 = \{\alpha_6\}$, $\mathfrak{a}_6 = \{\alpha_7\}$.

DeMoSt The prototype DeMoSt aims to enable users in the discovery of the modular structure of their ontologies, and the logical dependence between ontologies' axioms. We want to show how this is possible by working out a cognitive walkthrough for an ontology engineer who designed the ontology $\mathcal{O}$ as in Ex. 1. We assume them to have knowledge and experience with OWL ontologies, and with their features and representations, like class- and property-hierarchies. Moreover, we assume them to have some basic notions of posets, as, for example, how to represent them via Hasse diagrams. This last requirement enables the users to understand the meaning of the representation shown, but it is not necessary in order to navigate the tool.

DeMoSt is implemented as a plugin for Protégé, a popular editor of ontologies. DeMoSt takes an OWL ontology $\mathcal{O}$ as input and it outputs an interactive window divided into 4 panes:
- a *minimap* of the default (in this case, $\bot$-) AD of $\mathcal{O}$
- a *terminology* pane, showing the class hierarchy (or, optionally, the property hierarchy) of $\mathcal{O}$
- a *navigation* pane, showing the AD of $\mathcal{O}$ over a selected area, with a control bar to zoom in/out and to change the centre of the picture
- an *info* pane, either summarizing statistics about $\mathcal{O}$ or showing information concerning single terms or atoms whenever one of these is selected.

Now, let us suppose that the user wants to verify if the modular structure of $\mathcal{O}$ reflects her intentions. By opening $\mathcal{O}$ with DeMoSt, she obtains:
- in the navigation pane, a view of the whole ontology AD, as shown in Ex.1, at the default zoom rate
- the class-hierarchy in the terminology pane
- in the info pane, some data, like number of axioms (7), number of atoms (6), average size of genuine modules (~ 3.67), and so on.

Next, she wants to look closer at the term `Vegan`. She selects it in the terminology pane; then the minimap pane highlights all atoms where this term occurs (in our case, only in atom $\mathfrak{a}_3$), whilst the info pane shows the atoms' IDs (this function is useful for the case that the change in highlighting cannot be observed in a big graph). So she selects that area of the ontology in the minimap, the navigation pane recenters the picture around the selected area and shows $\mathfrak{a}_3$ and its dependence on $\mathfrak{a}_2$. She selects the atom $\mathfrak{a}_3$, and sees in the info pane that it contains only axiom α_4. However, she knows that a vegan has limitations also in what to drink (no milk, for example). She selects the term `drinks` in the terminology pane, and sees that the only atom where it occurs is $\mathfrak{a}_4$. So she understands that the ontology is not well modelled since it does not state any logical connection between α_4 and α_5, and that at least it should include the axiom $\alpha_8 =$ '$\mathtt{Vegan} \sqsubseteq \forall \mathtt{drinks}.\neg\mathtt{Milk}$', as well as replace the "$\equiv$" symbol with the "$\sqsubseteq$" symbol in axiom α_4.

Acknowledgements This material is based upon work partially supported by the National Science Foundation (NSF) under grant #0943879 and the NSF Plant Cyberinfrastructure Program (#EF-0735191).

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
2. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. of Artif. Intell. Research 31, 273–318 (2008)
3. Del Vescovo, C., Gessler, D., Klinov, P., Parsia, B., Sattler, U., Schneider, T., Winget, A.: Decomposition and modular structure of bioportal ontologies. In: Proc. of ISWC-11 (2011), to appear
4. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The modular structure of an ontology: an empirical study. In: Proc. of DL 2010. `ceur-ws.org` (2010)
5. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The modular structure of an ontology: Atomic decomposition. In: Proc. of IJCAI-11. pp. 2232–2237 (2011)
6. Ghilardi, S., Lutz, C., Wolter, F.: Did I damage my ontology? A case for conservative extensions in description logics. In: Proc. of KR-06. pp. 187–197 (2006)
7. Jiménez-Ruiz, E., Cuenca Grau, B., Sattler, U., Schneider, T., Berlanga Llavori, R.: Safe and economic re-use of ontologies: A logic-based methodology and tool support. In: Proc. of ESWC-08. LNCS, vol. 5021, pp. 185–199 (2008)
8. Konev, B., Lutz, C., Walther, D., Wolter, F.: Formal properties of modularization. In: Stuckenschmidt, H., Parent, C., Spaccapietra, S. (eds.) Modular Ontologies, LNCS, vol. 5445, pp. 25–66. Springer (2009)