The Modular Structure of an Ontology: Atomic Decomposition towards Applications

Chiara Del Vescovo

The University of Manchester, Oxford Road, Manchester, M13 9PL, UK
delvescc@cs.man.ac.uk

1 Introduction

Modularity in ontologies Modern ontologies can get quite large as well as complex, which poses challenges to tools and users when it comes to processing, editing, analyzing them, or reusing their parts. This suggests that exploiting modularity of ontologies might be fruitful, and research into this topic has been an active area for ontology engineering. Much recent effort has gone into developing *logically sensible* modules, that is, parts of an ontology which offer strong logical guarantees for intuitive modular properties. One such guarantee is called *coverage*. It means that a module captures all the ontology's knowledge about a given set of terms (signature). A module in this sense is a subset of an ontology's axioms that provides coverage for a signature, and each possible signature determines such a module. The minimal modules to provide coverage for a signature are those based on Conservative Extensions (CEs) [2], that are however not feasible to be computed for many expressive languages. Modules based on syntactic locality [5] also provide coverage because they are efficiently computable approximations of CEs; however, such modules are not in general minimal.

The extraction of such a module given a set of terms (*signature*) is well understood and starting to be deployed in standard ontology development environments, such as Protégé 4,[1] and online.[2] Locality-based modules have already been effectively used for ontology reuse [14] and as a subservice for incremental reasoning [3]. However, we think that by investigating the family $\mathfrak{F}_{\mathcal{O}}$ of all locality-based modules we can obtain information about topicality, connectedness, structure, superfluous parts of an ontology, or agreement between actual and intended modeling.

Previous work In [6] we investigated the number of (locality-based) modules that an ontology can have. There are examples of artificial ontologies with exponentially many w.r.t. their size: for example, the ontology $\mathcal{O} = \{\mathtt{A}_i \sqsubseteq \mathtt{B} \,|\, i = 1, \ldots, n\}$, where each subset of the ontology is a module. However, we tried to understand if real ontologies generate an exponential family $\mathfrak{F}_{\mathcal{O}}$ of modules. To this aim, we selected some ontologies from the TONES ontology repository[3] and tried to extract all of their modules. The results we obtained made us tend towards rejecting the hypothesis, but they were not strong enough for a clear rejection.

[1] http://www.co-ode.org/downloads/protege-x
[2] http://owl.cs.manchester.ac.uk/modularity
[3] http://owl.cs.manchester.ac.uk/repository/browser

In [7] we introduced a different approach to look at modules: we noticed that an ontology can be partitioned into building blocks, called *atoms*, that do not split over two modules. Atoms are interesting because they are in 1-1 correspondence with *genuine modules*, i.e., modules that are not the union of two uncomparable (w.r.t. set inclusion) modules. Moreover, they are computable in polynomial time (provided that the extraction of a locality-based module is polynomial). The set of atoms is called *Atomic Decomposition* (AD). The AD of an ontology is stable, in the sense that is well-defined given any (suitable) notion of locality-based module. This suggests us that we could use it for performing several tasks of interest for users of ontologies, as the estimation of the number of modules of an ontology. We started exploiting this matter in [8].

Following [7], in this paper we describe more extensively the AD of an ontology: its definition, its properties, its generation. We then introduce some tasks of interest for ontology engineers that could be performed by means of the ADs. For each such task, we discuss with the help of several examples some issues about the suitability of the sole AD to perform it. Hence, we introduce a family of refinements of AD, called *Labelled Atomic Decomposition* (LAD), useful to solve the issues raised. In particular, these issues are deeply discussed w.r.t. one such specific task: the *Fast Module Extraction* (FEM), t.i., the extraction of a single module without loading the ontology.

2 Preliminaries

We assume the reader to be familiar with Description Logics [1], and only briefly sketch here some of the central notions around locality-based modularity. We use $\mathcal{L}$ for a Description Logic, e.g., $\mathcal{SHIQ}$, and $\mathcal{O}, \mathcal{M}$, etc., for a knowledge base, i.e., a finite set of axioms. Moreover, we respectively use $\widetilde{\alpha}$ or $\widetilde{\mathcal{O}}$ for the signature of an axiom α or of an ontology $\mathcal{O}$, i.e., the set of concept, role, and individual names used in α or in $\mathcal{O}$.

Conservative extensions (CEs) capture the above described encapsulation of knowledge. They are defined as follows.

Definition 1. *Let $\mathcal{L}$ be a DL, $\mathcal{M} \subseteq \mathcal{O}$ be $\mathcal{L}$-ontologies, and Σ be a signature.*

1. *$\mathcal{O}$ is a* deductive Σ-conservative extension (Σ-dCE) *of $\mathcal{M}$ w.r.t. $\mathcal{L}$ if for all axioms α over $\mathcal{L}$ with $\widetilde{\alpha} \subseteq \Sigma$, it holds that $\mathcal{M} \models \alpha$ if and only if $\mathcal{O} \models \alpha$.*
2. *$\mathcal{M}$ is a* dCE-based module for Σ *of $\mathcal{O}$ if $\mathcal{O}$ is a Σ-dCE of $\mathcal{M}$ w.r.t. $\mathcal{L}$.*

Unfortunately, CEs are hard or even impossible to decide for many DLs [11,15,17]. Therefore, approximations have been devised. We focus on *syntactic locality* (here for short: locality). Given an ontology $\mathcal{O}$, and a set of terms, called *seed signature*, $\Sigma \subseteq \widetilde{\mathcal{O}}$, we say that an axiom $\alpha \in \mathcal{O}$ is $\bot$-*local* w.r.t. Σ if we can clearly identify it as a tautology when all the terms not in Σ are substituted by $\bot$ (the formal definition can be found in [5]). An analogous definition can be made for $\top$-locality. Then, a locality-based module is recursively computed as follows: starting from an empty set $\mathcal{M}$, each axiom $\alpha \in \mathcal{O}$ is tested whether it is local w.r.t. Σ; if not, α is added to $\mathcal{M}$, the signature Σ is extended with all terms in $\widetilde{\alpha}$, and the test is re-run against the extended signature. Then, $\mathcal{M} \subseteq \mathcal{O}$ and all axioms in $\mathcal{O} \setminus \mathcal{M}$ being local w.r.t. $\Sigma \cup \widetilde{M}$ is sufficient for $\mathcal{O}$ to be a Σ-dCE of $\mathcal{M}$. By alternating the extraction of $\bot$- and $\top$-module over the previously extracted module, we obtain various notion of modules. When the fixpoint is reached, the resulting notion is called $\top\bot^*$-*locality*. Locality-based modules can be efficiently computed and provide coverage; that is, they capture *all* the relevant entailments, but not necessarily *only* those [5,13]. A module extractor is implemented in the OWL API.[4]

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 following properties of locality-based modules will be of interest for our modularization [5,17].

Proposition 2. *Let $\mathcal{O}$ be an ontology, Σ a signature and $x \in \{\top, \bot, \top\bot^*\}$. Then the following properties hold:*
(a) for any Σ', x-mod$(\Sigma, \mathcal{O}) \subseteq x$-mod$(\Sigma \cup \Sigma', \mathcal{O})$ (monotonicity)
(b) for Σ' with $\Sigma \subseteq \Sigma' \subseteq \Sigma \cup \widetilde{\mathcal{M}}$, x-mod$(\Sigma', \mathcal{O}) = x$-mod$(\Sigma, \mathcal{O})$ (self-containedness)
(c) each axiom α entailed by $\mathcal{O} \setminus x$-mod$(\Sigma, \mathcal{O})$ and such that $\widetilde{\alpha} \subseteq \Sigma$ is a tautology (depletingness).

Proposition 3. *Any notion of locality-based modules satisfying the properties in Prop. 2 is such that any given signature generates a unique module.*

From now on, we focus on the $\top\bot^*$ notion of locality-based modules. However, we want to underline that what we discuss in the rest of the paper can be carried out for each notion of module satisfying monotonicity, self-containedness, and depletingness.

3 Atomic Decomposition

In [7] we introduced a new approach to represent the whole family $\mathfrak{F}_{\mathcal{O}}$ of locality-based modules of an ontology $\mathcal{O}$. The key point is observing that some axioms appear in a module only if other axioms do. In this spirit, we defined a notion of "logical dependence" between axioms: the idea is that an axiom α depends on another axiom β if whenever α occurs in a module $\mathcal{M}$ then β also belongs to $\mathcal{M}$.

To keep the formalization clean, we remove from the ontology *syntactic tautologies*, i.e. always-local axioms, and *global axioms*, i.e. axioms that belong to all modules. We can always remove these unwanted axioms and consider them separately. Then, for each axiom α is well-defined the *smallest* module containing it.

Proposition 4. *The module $\top\bot^*$-mod$(\widetilde{\alpha}, \mathcal{O})$ is the smallest containing α.*

Proof. We recall that $\top\bot^*$-mod satisfies the properties as in Prop. 2. Then:

1. α is non-local w.r.t. $\widetilde{\alpha}$ (because is not a syntactic tautology), hence $\mathcal{M}_\alpha$ is not empty
2. $\mathcal{M}_\alpha$ is the unique and thus smallest module for the seed signature $\widetilde{\alpha}$
3. by monotonicity, enlarging the seed signature $\widetilde{\alpha}$ results in a superset of $\mathcal{M}_\alpha$
4. $\mathcal{M}' = \top\bot^*$-mod$(\widetilde{\mathcal{M}'}, \mathcal{O}) = \top\bot^*$-mod$(\widetilde{\mathcal{M}'} \cup \widetilde{\alpha}, \mathcal{O}) \supseteq \top\bot^*$-mod$(\widetilde{\alpha}, \mathcal{O})$ by self-containedness and monotonicity, thus any module $\mathcal{M}'$ that contains α needs to contain also $\mathcal{M}_\alpha$. $\square$

Definition 5. *The module $\mathcal{M}_\alpha = \top\bot^*$-mod$(\widetilde{\alpha}, \mathcal{O})$ as in Prop. 4 is called α-*module.

[4] http://owlapi.sourceforge.net/

The dependency between axioms allows us to identify clumps of highly interrelated axioms that never split over two or more modules [7]; these clumps are called *atoms*.

Definition 6. *An* atom *is a maximal disjoint subset of an ontology such that their axioms either appear always together in modules, or none of them does.*

Definition 7. *The family of atoms of an ontology $\mathcal{O}$ is denoted by $\mathcal{A}(\mathfrak{F}_\mathcal{O})$ and is called* Atomic Decomposition (AD).

The AD is evidently a partition of the ontology, thus is linear w.r.t. the size of the ontology. Moreover, atoms are the building blocks of all modules [10].

Proposition 8. *Each module is the union of suitable atoms.*

We summarize in the following table the ontologies' fragments described so far.

Structure	$\mathcal{O}$	$\mathfrak{F}_\mathcal{O}$	$\mathcal{A}(\mathfrak{F}_\mathcal{O})$
Elements	axioms α	modules $\mathcal{M}$	atoms $\mathfrak{a}, \mathfrak{b}, \ldots$
Maximal size	baseline	exponential	linear
Mathem. object	set	family of sets	poset

Proposition 9. *Let $\mathfrak{a}$ be an atom in the AD $\mathcal{A}(\mathfrak{F}_\mathcal{O})$ of an ontology $\mathcal{O}$; then, for any selection of axioms $\mathcal{S} = \{\alpha_1, \ldots, \alpha_\kappa\} \subseteq \mathfrak{a}$ we have that* $\top\bot^*$-mod$(\widetilde{\mathcal{S}}, \mathcal{O}) = \mathcal{M}_\alpha$*. In particular, for each $\alpha_i \in \mathfrak{a}$, $\mathcal{M}_{\alpha_i} = \mathcal{M}_\alpha$.*

Proof. Let $\alpha \in \mathfrak{a}$ be an axiom, and let us consider the module $\mathcal{M}_\alpha$. Then:

1. $\mathfrak{a} \subseteq \mathcal{M}_\alpha$ by the definition of atoms
2. as a consequence, $\mathcal{M}_\alpha \supseteq \top\bot^*$-mod$(\widetilde{\mathcal{S}}, \mathcal{O})$ for every selection $\mathcal{S}$ of axioms from $\mathfrak{a}$
3. by Prop. 4, the inverted inclusion $\top\bot^*$-mod$(\widetilde{\alpha}_i, \mathcal{O}) \supseteq \mathcal{M}_\alpha$ also holds. □

A module $\mathcal{M}_\mathfrak{a} = \top\bot^*$-mod$(\widetilde{\mathfrak{a}}, \mathcal{O})$ is called *compact*. From Prop. 9 it is clear that the set of compact modules coincides with the one of α- modules. Hence, we can denote by $\mathcal{M}_\mathfrak{a}$ the module $\mathcal{M}_\alpha$ for each $\alpha \in \mathfrak{a}$. Now, we are ready to extend the definition of "logical dependency" to atoms.

Definition 10. *Let $\mathfrak{a}$ and $\mathfrak{b}$ be two distinct atoms of an ontology $\mathcal{O}$. Then:*

- $\mathfrak{a}$ *is* dependent *on* $\mathfrak{b}$ *(written $\mathfrak{a} \succeq \mathfrak{b}$) if $\mathcal{M}_\mathfrak{b} \subseteq \mathcal{M}_\mathfrak{a}$*
- $\mathfrak{a}$ *and* $\mathfrak{b}$ *are* independent *if $\mathcal{M}_\mathfrak{a} \cap \mathcal{M}_\mathfrak{b} = \emptyset$*
- $\mathfrak{a}$ *and* $\mathfrak{b}$ *are* weakly dependent *if, they are neither independent nor dependent; in this case, there exists an atom $\mathfrak{c}$ which both $\mathfrak{a}$ and $\mathfrak{b}$ are dependent on.*

Thanks to Def. 10, the AD inherites the mathematical structure of partially ordered set, thus can be represented by means of a Hasse diagram.

Proposition 11. *The binary relation "$\succeq$" as in Def. 10 is a partial order over the set $\mathcal{A}(\mathfrak{F}_\mathcal{O})$ of atoms of an ontology $\mathcal{O}$.*

Proof. This is true because $\succeq$ satisfies reflexivity, antisymmetry, and transitivity. □

Algorithm 1 Atomic decomposition

Input: An ontology $\mathcal{O}$.

Output: The set $\mathfrak{G}$ of α-modules; the poset of atoms $(\mathcal{A}(\mathfrak{F}_\mathcal{O}), \succeq)$; the set of generating axioms $\texttt{GenAxs}$; for $\alpha \in \texttt{GenAxs}$, the cardinality $\texttt{CardAtom}(\alpha)$ of its atom.

```
ToDoAxs ← ⊤⊥*-mod(Õ, O) \ ⊤⊥*-mod(∅, O)
GenAxs ← ∅
for each α ∈ ToDoAxs do
    Module(α) ← ⊤⊥*-mod(α̃, O)        {≠ ∅}
    new ← true
    for each β ∈ GenAxs do
        if Module(α) = Module(β) then
            Atom(β) ← Atom(β) ∪ {α}
            CardAtom(β) ← CardAtom(β) + 1
            new ← false
        end if
    end for
    if new = true then
        Atom(α) ← {α}
        CardAtom(α) ← 1
        GenAxs ← GenAxs ∪ {α}
    end if
end for
for each α ∈ GenAxs do
    for each β ∈ GenAxs do
        if β ∈ Module(α) then
            Atom(β) ⪯ Atom(α)
        end if
    end for
end for
A(F_O) ← {Atom(α) | α ∈ GenAxs}
G ← {Module(α) | α ∈ GenAxs}
return [(A(F_O), ⪰), G, GenAxs, CardAtom(·)]
```

Name	#logical axioms	#α-mods	#Con. comp.	#max. mod.	#max. atom
Koala	42	23	5	18	7
Mereology	44	17	2	11	4
University	52	31	11	20	11
People	108	26	1	77	77
miniTambis	173	129	85	16	8
OWL-S	277	114	1	57	38
Tambis	595	369	119	236	61
Galen	4,528	3,340	807	458	29

Table 1. Experiments summary

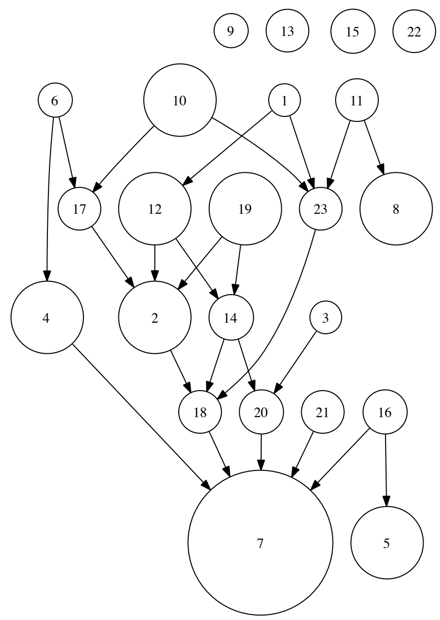

Figure 1. The AD of Koala

Prop. 9 and Prop. 11 provide the basis for our polynomial algorithm for the computation of the AD since it allows us to construct $\mathcal{A}(\mathfrak{F}_\mathcal{O})$ via α-modules only. The whole procedure is described in Alg. 1. A proof for its correctness can be found in [10].

We ran Algorithm 1 on a selection of ontologies, including those used in [6], and indeed managed to compute the AD in all cases, even for ontologies where a complete modularization was previously impossible. Table 1 summarizes ontology data: size, expressivity, number of compact modules (= number of atoms), number of connected components in the AD poset, size of largest compact module and of largest atom. Our tests were obtained on a 2.16 GHz Intel Core 2 Duo Macbook with 2 GB of memory running Mac OS X 10.5.8; each AD was computed within a couple of seconds (resp. 3 minutes for Galen).

We have also generated a graphical representation using GraphViz[5]. Our ADs show atom size as node size, see e.g. Fig. 1. It shows four isolated atoms, e.g., Atom 22, consisting of the axiom `DryEucalyptForest` $\sqsubseteq$ `Forest`. This means that, although other modules may use some (but not all) 22's terms, they do not "need" 22's axioms for any entailment. Hence, removing (the axioms in) isolated atoms from the ontology would not result in the loss of any entailments regarding other modules or terms. Of course, for entailments involving both `DryEucalyptForest` and `Forest` and possibly other terms, axioms in isolated atoms may be needed. A similar structure is observable in all ontologies considered, see the graphs at `http://bit.ly/i4olY0`.

The following results have a deep impact on the way we describe modules in ADs: the poset structure of an AD is a 1-1 representation of compact modules.

Definition 12. *The* principal ideal of an atom $\mathfrak{a}$ *is the set* $(\mathfrak{a}] = \{\alpha \in \mathfrak{b} \mid \mathfrak{b} \preceq \mathfrak{a}\} \subseteq \mathcal{O}$.

Lemma 13. *Principal ideals of atoms are modules.*

Proof. Given an atom $\mathfrak{a} \in \mathcal{A}(\mathfrak{F}_\mathcal{O})$, we want to compare its principal ideal $(\mathfrak{a}] = \bigcup_{\mathfrak{b} \preceq \mathfrak{a}} \mathfrak{b}$ with the module $\mathcal{M}_\alpha$. By the definition of atoms, $\mathcal{M}_\alpha \supseteq (\mathfrak{a}]$. We still need to prove that the equality holds. By contraposition, let $\mathcal{M}_\alpha$ be a proper superset of $(\mathfrak{a}]$. Then it contains at least one atom $\mathfrak{b}$ which $\mathfrak{a}$ is not dependent on. Let β be an axiom in $\mathfrak{b}$, and let us consider $\mathcal{M}_\beta$. By Prop. 4, $\mathcal{M}_\beta$ is the smallest module containing $\mathfrak{b}$. Then, $\mathcal{M}_\beta$ is contained in $\mathcal{M}_\alpha$, and since the latter is the smallest module containing $\mathfrak{a}$, this means that $\mathfrak{a}$ is dependent on $\mathfrak{b}$. This last fact contradicts the assumption. □

Prop. 9 implies that two axioms from the same atom generate the same compact module. The converse also holds.

Proposition 14. *Let α, β be two axioms such that $\mathcal{M}_\alpha = \mathcal{M}_\beta$. Then, an atom $\mathfrak{a}$ exists such that $\alpha, \beta \in \mathfrak{a}$.*

Proof. By contraposition, let $\mathfrak{a}$ and $\mathfrak{b}$ be two distinct (hence, disjoint) atoms such that $\alpha \in \mathfrak{a}$ and $\beta \in \mathfrak{b}$. Then, by Prop. 13 the principal ideals $(\mathfrak{a}]$ and $(\mathfrak{b}]$ are also distinct modules, and this contradicts the hypothesis. □

Another interesting property is the existence of a mapping, denoted by $r_\mathcal{O}$, between the family $\mathfrak{F}_\mathcal{O}$ of modules of the ontology $\mathcal{O}$ into the set of antichains of the poset structure of the AD, such that if $\mathcal{M} = \mathfrak{a}_1 \cup \ldots \cup \mathfrak{a}_n$, then $r_\mathcal{O}(\mathcal{M})$ is the minimum set of atoms such that $\mathcal{M} = (\mathfrak{a}_1] \cup \ldots \cup (\mathfrak{a}_\kappa]$. In particular, $\{\mathfrak{a}_1, \ldots, \mathfrak{a}_\kappa\}$ is a set of uncomparable atoms. Unfortunately, this mapping is not 1-1.

Corollary 15. *For each module $\mathcal{M}$ of an ontology $\mathcal{O}$, there are uncomparable (w.r.t. the dependency relation $\preceq$) atoms $\mathfrak{a}_1, \ldots, \mathfrak{a}_\kappa$ such that $\mathcal{M} = \bigcup_{i=1}^n (\mathfrak{a}_i]$.*

Proof. By the definition of atoms, if $\mathcal{M}$ contains one axiom from an atom $\mathfrak{a}_i$, then it contains all its axioms. By the definition of dependency, if $\mathcal{M}$ contains one atom $\mathfrak{a}_i$, then it contains all the atoms that $\mathfrak{a}_i$ depends on. Finally, we can consider only uncomparable atoms because in case $\mathcal{M}$ contains an $\mathfrak{b}$ such that $\mathfrak{a}_i \succeq \mathfrak{b}$, then $\mathfrak{b}$ is already included in the representation of $\mathcal{M}$ as $(\mathfrak{a}_1] \cup \ldots \cup (\mathfrak{a}_i] \cup \ldots \cup (\mathfrak{a}_\kappa]$. □

[5] `http://www.graphviz.org/About.php`

3.1 Genuine modules

Another notion of module that we want to describe consists of those that do not fall apart into more than one piece, and hence have a strong internal coherence.

Definition 16. *A module is called* fake *if there exist two uncomparable (w.r.t. $\subseteq$) modules $\mathcal{M}_1, \mathcal{M}_2$ with $\mathcal{M}_1 \cup \mathcal{M}_2 = \mathcal{M}$; a module is called* genuine *if it is not fake.*

Lemma 17. *The notions of α- and genuine modules coincide.*

Proof. Both directions are proven by contraposition.

α- $\Rightarrow$ *genuine* : Let $\mathcal{M}$ be a fake module. Then there are two uncomparable modules $\mathcal{M}_1$ and $\mathcal{M}_2$ such that $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$. In particular, there exist suitable atoms such that $\mathcal{M}_1 = \mathfrak{a}_1 \cup \ldots \cup \mathfrak{a}_\kappa$ and $\mathcal{M}_2 = \mathfrak{b}_1 \cup \ldots \cup \mathfrak{b}_\ell$. Since the modules are uncomparable, then there is at least one atom $\mathfrak{a}_k$ in $\mathcal{M}_1$ such that $\mathfrak{a}_k \notin \{\mathfrak{b}_1, \ldots, \mathfrak{b}_\ell\}$; similarly, there is at least one atom $\mathfrak{b}_l$ in $\mathcal{M}_2$ such that $\mathfrak{b}_k \notin \{\mathfrak{a}_1, \ldots, \mathfrak{a}_\kappa\}$. Finally, there is no atom $\mathfrak{c} \in \mathcal{M} = \{\mathfrak{a}_1, \ldots, \mathfrak{a}_\kappa, \mathfrak{b}_1, \ldots, \mathfrak{b}_\ell\}$ dependent both on $\mathfrak{a}_k$ and on $\mathfrak{b}_l$, otherwise these atoms would be both in $\mathcal{M}_1$ and in $\mathcal{M}_2$; hence, $\mathcal{M}$ is not compact.

genuine $\Rightarrow$ α- : Let $\mathcal{M}$ be a non compact module. By Cor. 15, there exist atoms $\mathfrak{a}_1, \ldots, \mathfrak{a}_\kappa$ such that $\mathcal{M} = (\mathfrak{a}_1] \cup \ldots \cup (\mathfrak{a}_\kappa]$, with $\kappa \geq 2$. By Lemma 13 we have that the principal ideal of every atom is a module. Hence $\mathcal{M} = (\mathfrak{a}_1] \cup \ldots \cup (\mathfrak{a}_\kappa]$ is the union of uncomparable modules, and more in specific, fake. □

A straightforward consequence of Cor. 15 and Lemma 17 is the set of genuine modules to be a base for all locality-based modules: more precisely, each module is the union of a combination of genuine modules. However, the converse does not hold: not all combinations are modules, and given an AD is non-trivial to determine which combinations of genuine modules generate a module.

Example 18. Let us consider the ontology $\mathcal{O} = \{\mathtt{A}_i \sqsubseteq \mathtt{A}_{i+1} \mid i = 0, \ldots, n-1\}$. Then, the AD of this ontology consists of n atoms pairwise independent. However, for each choice of two terms $\mathtt{A}_\kappa, \mathtt{A}_\ell$ with $\kappa < \ell$, the module for the seed signature $\Sigma = \{\mathtt{A}_\kappa, \mathtt{A}_\ell\}$ is the set $\top\bot^*(\Sigma, \mathcal{O}) = \{\mathtt{A}_\kappa \sqsubseteq \mathtt{A}_{\kappa+1}, \ldots, \mathtt{A}_{\ell-1} \sqsubseteq \mathtt{A}_\ell\}$. In other words, the atoms concerning the terms "between" $\mathtt{A}_\kappa$ and $\mathtt{A}_\ell$ are not really independent, because they are "pulled into" the module for Σ.

The reason for this to happen can be found in the overlapping of minimal seed signatures for genuine modules. We have seen in Sect.2 how modules are extracted, and how the seed signature is "enlarged" to include the signature of all non-local axioms. Hence, if the extended signature overlaps with the minimal seed signature of a different genuine module $\mathcal{M}'$, then $\mathcal{M}'$ is pulled into the module extracted.

4 Towards Applications

4.1 What for?

Fast Module Extraction (FME) : Ontologies are sometimes difficult even to load, so an interesting task to perform would be the off-line extraction of modules by using the

AD of an ontology; in practice, we want to be able to recognize which combinations of atoms generate a module, that is, find the inverse of the representing mapping $r_{\mathcal{O}}$. As briefly introduced in Ex. 18, this operation does not directly follow from the AD: we need more information concerning the minimal seed signatures of genuine modules, because their overlapping can cause other atoms to fall into the module we are extracting. Further in this section we are going to discuss some preliminary issues about FME.

Module Count (MC) : In [6] we tried to compute a full modularization for the ontologies of different size listed in Table 1 in order to test the hypothesis that the number of modules does not grow exponentially with the size of the ontology. Unfortunately, we managed to compute all modules for two ontologies only, namely Koala and Mereology. For the others, we sampled subontologies and extracted all of their modules. The results we obtained made us tend towards rejecting the hypothesis, but they were not strong enough for a clear rejection. From Cor. 15 we derive that one plausible application of ADs is an estimate of the number of modules of an ontology, as a first approximation by counting the number of antichains of the AD poset. However, this approach has been proven unsuccessful: the estimate is still too large, because not all antichains generate a module, as in Ex.18. So the problem remains open, and only preliminary though encouraging results are reported in [8].

Topicality for Ontology Comprehension (TOC) : The AD of an ontology derives from strong logical properties of locality-based modules, so we expect it to preserve, or indeed reveal, these properties. The first observation that we want to point out is that, given an ontology and a notion of module, the structure defined in its AD is uniquely determined. The stability of this structure implies that the issues described in what follows are well-defined. Since modules are defined as set of axioms providing coverage to a given set of terms Σ, it is natural to investigate the relations between terms and modules.

In [9] we have exploited different notions of topicality in ontologies (and, more in general, for logic-based theories) for notion of modules with strong logical properties. Clearly, a notion as AD is too loose to define topicality for ontologies, since the sole structure does not explain what the ontology is about. A refined suitable version of AD would also contain labels to describe the content of an atom. Preliminary results in this sense are reported in [9].

Beside the tasks described so far that we started addressing, we identified at least other two tasks of interest, that we briefly describe in what follows.

Suggesting axioms to Repair First (RF) : One task that ontology engineers perform commonly is maintaining and repairing ontologies. Interesting tools for this task make use of *Justifications* [12]. Justifications are minimal sets of axioms that explain why a specific entailment holds. Ontology engineers often search justifications for classes to be unsatisfiable. Unfortunately, justifications can be large, and numerous. The logical dependency of axioms defined in AD could be used to suggest which axioms of a justification to repair first, and in particular, those that the other axioms depend on. The hope is that mistakes in the modeling phase propagate within the ontology by means of the logical dependency as defined here.

Suggesting Seed Signatures (SigSug) : The users of ontologies are often interested in extracting a (possibly minimal) set of axioms that "know everything" about a specific set of terms. However, locality-based modules are designed to provide coverage for a given seed signature Σ. Even if related to what required, modules are often too small, because users are interested also in the relation that a term has with some of its sub- or super-classes, or sub- or super-roles. However, there is no trivial relation between the seed signature of input and the signature of the module extracted, so these relations are sometimes left out the module. The current solution for this problem is the extraction of a module for an enlarged signature, but the AD could be of interest for refining this approach.

Throughout the description of the various tasks, we mentioned that often the AD is too loose w.r.t. the actual modular structure of the ontology, hence adding information can be of help in real applications. One possible refinement of ADs consists of including information about seed signatures in the AD: the result is called *Labelled Atomic Decomposition* (LAD).

Definition 19. *Given: an ontology $\mathcal{O}$ and its AD $\mathcal{A}(\mathfrak{F}_{\mathcal{O}}) = \{\mathfrak{a}_1, \ldots, \mathfrak{a}_n\}$, a* labelling function $\mathsf{Lab}(.)$ *is a function from $\mathcal{A}(\mathfrak{F}_{\mathcal{O}})$ to the power set of $\widetilde{\mathcal{O}}$, that matches each atom with a suitable set of terms from the ontology.*

The information to be added depends on the task the users want to perform. Throughout the rest of this section we briefly discuss the suitability of a specific labelling function to perform FME directly from the LAD of an ontology.

4.2 LAD for Fast Module Extraction

Let us consider the labelling function $\mathsf{Lab}_{\mathrm{ssig}}$ such that to each atom $\mathfrak{a}$ is assigned the set of minimal seed signatures that generate the module $\mathcal{M} = (\mathfrak{a}]$. This labelling is useful to discover "hidden relations" between an atom and terms that do not occur in it.

Example 20. Let us consider the ontology $\mathcal{O} = \{\mathtt{A} \equiv \mathtt{B}, \mathtt{B} \sqsubseteq \mathtt{C}, \mathtt{B} \sqcap \mathtt{D} \sqsubseteq \mathtt{C} \sqcup \mathtt{E} \sqcup (\mathtt{G} \sqcup \neg\mathtt{G}), \mathtt{D} \sqsubseteq \mathtt{E}, \mathtt{E} \equiv \mathtt{F}\}$. Each axiom identifies an atom, and $\mathcal{O}$ equals the principal ideal of the atom $\mathfrak{a}_3$ consisting of the axiom $\mathtt{B} \sqcap \mathtt{D} \sqsubseteq \mathtt{C} \sqcup \mathtt{E} \sqcup (\mathtt{G} \sqcup \neg\mathtt{G})$. Although the signature of $\mathfrak{a}_3$ contains neither $\mathtt{A}$ nor $\mathtt{F}$, the set $\Sigma = \{\mathtt{A}, \mathtt{F}\}$ is indeed a minimal seed signature of the module $(\mathfrak{a}_3]$. The need of this axiom for the signature Σ is not evident at first sight.

On the other hand, $\mathsf{Lab}_{\mathrm{ssig}}$ does not include "irrelevant" terms: since under any interpretation of $\mathtt{G}$ the concept $\mathtt{G} \sqcup \neg\mathtt{G}$ is always $\top$, then $\mathtt{G}$ does not appear in any of the minimal seed signatures of the atom $\mathfrak{a}_3$. Although this can be seen as a good behaviour of $\mathsf{Lab}_{\mathrm{ssig}}$, we need to consider how $\top\bot^*$-modules are extracted: whenever an axiom α is non local, the seed signature Σ is extended with all terms in $\widetilde{\alpha}$. This means that in our example $\mathtt{G}$ belongs to the extended signature of the module $\mathfrak{a}_3$, and can interfere with other terms of the seed signature of input, even if it is logically irrelevant. We need to keep track of this information too. We define $\mathsf{Lab}_{\mathrm{FEM}}$ to be the refinement of $\mathsf{Lab}_{\mathrm{ssig}}$ by adding to the label of each atom also its irrelevant terms, i.e., all terms in the module that do not occurr in any minimal seed signature. In Fig. 2 we show such LAD for the ontology Koala. The refinement of $\mathsf{Lab}_{\mathrm{ssig}}$ affects only the atom labelled $\{\mathtt{Koala}\}$.

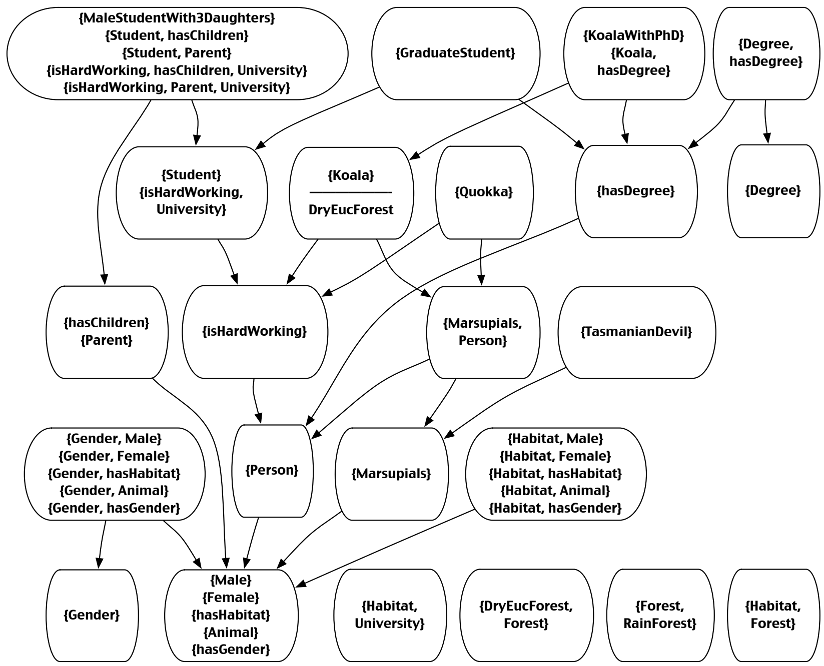

Figure 2. LAD for performing FME of the ontology Koala

A problem that arises with $\mathsf{Lab}_{\mathrm{FEM}}$ consists of the possibility of labels to be of exponential size w.r.t. the ontology size, as in the following example.

Example 21. Let us consider the the family of ontologies $\mathcal{O}_n = \{\mathtt{A}_i \equiv \mathtt{A}_{i-1} \sqcup \mathtt{A}'_{i-1}, \mathtt{B}_i \equiv \mathtt{B}_{i-1} \sqcup \mathtt{B}'_{i-1}, \mathtt{C}_i \equiv \mathtt{C}_{i-1} \sqcup \mathtt{C}'_{i-1}, \mathtt{D}_i \equiv \mathtt{D}_{i-1} \sqcup \mathtt{D}'_{i-1}, \mathtt{A}_i \sqcup \mathtt{B}_i \sqsubseteq \mathtt{C}_i \sqcap \mathtt{D}_i \,|\, i = 1, \ldots, n\}$. Then, $\#\mathcal{O}_n = 5n$, and each AD consists of $5n$ atoms with an axiom each. Now, some minimal seed signatures for the atom $\mathfrak{a}_i^n = \{\mathtt{A}_i \sqcup \mathtt{B}_i \sqsubseteq \mathtt{C}_i \sqcap \mathtt{D}_i\}$ contain 2 terms, one from $\{\mathtt{A}_i, \mathtt{B}_i\}$ and one from $\{\mathtt{C}_i, \mathtt{D}_i\}$. However, each term can be replaced by the two terms defining it (for example, $\mathtt{A}_i$ can be replaced by $\mathtt{A}_{i-1}, \mathtt{A}'_{i-1}$). Since this procedure can be recursively applied, the atom $\mathfrak{a}_i^n$ results to have at least 4^i minimal seed signatures.

Despite the discussion throughout this section, a procedure to perform FME is still not defined, and the exploitation of this matter is in our future work.

5 Outlook

We presented the *Atomic Decomposition* of an ontology, and showed its definition, its properties, and its tractable generation. The AD reveals the overall modular structure of an ontology, thus we expect to apply such decomposition in various scenario, from the module count, to the support to ontology engineers in the modeling phase. We have also introduced a family of refinements of AD, called *Labelled Atomic Decompositions*, justifying the need for labels in the specific task consisting of extracting a module without loading the ontology.

Future work includes the completion of the preliminary results described here. In particular, we want to explore suitable ADs/LADs for the tasks described in this paper. Then, we are open to investigate other tasks that could be of interest for users of ontologies and where a suitable LAD would be of help.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. F. Patel-Schneider, eds. The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
2. G.Antoniou, K. Kehagias. A note on the refinement of ontologies. Int. J. of Intelligent Systems, vol. 15, pp. 623–632 (2000)
3. Cuenca Grau, B., Halaschek-Wiener, C., Kazakov, Y.: History matters: Incremental ontology reasoning using modules. In: Proc. of ISWC/ASWC-07. LNCS, vol. 4825, pp. 183–196 (2007)
4. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: A logical framework for modularity of ontologies. In Proc. of IJCAI-07. pp. 298–304 (2007)
5. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. of Artif. Intell. Research, vol. 31, pp. 273–318 (2008)
6. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The modular structure of an ontology: an empirical study. In V. Haarslev, D. Toman, and G. Weddell (eds.) Proc. of DL 2010. `ceur-ws.org`, vol. 573. (2010)
7. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The Modular Structure of an Ontology: Atomic Decomposition. Accepted for IJCAI-11 (2011)
8. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The Modular Structure of an Ontology: Atomic Decomposition and Module Count. Accepted for WoMO-11 (2011)
9. Del Vescovo, C., Parsia, B., Sattler, U.: Topicality in Logic-Based Ontologies. Accepted for ICCS-11 (2011)
10. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The Modular Structure of an Ontology: Atomic Decomposition. Tech. rep., The University of Manchester (2011) Available at `http://bit.ly/i4olY0`
11. Ghilardi, S., Lutz, C., Wolter, F.: Did I damage my ontology? A case for conservative extensions in description logics. In: Doherty, P., Mylopoulos, J., Welty, C.A. (eds.) Proc. of KR-06. pp. 187–197. AAAI Press (2006)
12. Horridge, M.. Parsia, B., Sattler, U.: Laconic and Precise Justifications in OWL. In: Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) Proc. of ISWC-08. LNCS, vol. 5318, pp. 323–338 (2008)
13. 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)
14. Jimeno, A., Jiménez-Ruiz, E., Berlanga, R., Rebholz-Schuhmann, D.: Use of shared lexical resources for efficient ontological engineering. In: SWAT4LS-08. `ceur-ws.org` (2008)
15. Konev, B., Lutz, C., Walther, D., Wolter, F.: Formal properties of modularization. In: Modular Ontologies, LNCS, vol. 5445, pp. 25–66. Springer-Verlag (2009)
16. Kontchakov, R., Pulina, L., Sattler, U., Schneider, T., Selmer, P., Wolter, F., Zakharyaschev, M.: Minimal module extraction from DL-Lite ontologies using QBF solvers. In: Proc. of IJCAI-09. pp. 836–841 (2009)
17. Sattler, U., Schneider, T., Zakharyaschev, M.: Which kind of module should I extract? In: Cuenca Grau, B., Horrocks, I., Motik, B., Sattler, U. (eds.) DL 2009. `ceur-ws.org`, vol. 477. (2009)