Decomposition and Modular Structure of BioPortal Ontologies*

Chiara Del Vescovo[1], Damian D. G. Gessler[2], Pavel Klinov[2], Bijan Parsia[1],
Ulrike Sattler[1], Thomas Schneider[3], and Andrew Winget[4]

[1] University of Manchester, UK
{delvescc|bparsia|sattler}@cs.man.ac.uk
[2] University of Arizona, AZ, USA
dgessler@iplantcollaborative.org,pklinov@email.arizona.edu
[3] Universität Bremen, Germany
tschneider@informatik.uni-bremen.de
[4] St. John's College, NM, USA
andrewwinget@gmail.com

Abstract We present the first large scale investigation into the modular structure of a substantial collection of state-of-the-art biomedical ontologies, namely those maintained in the NCBO BioPortal repository.[5] Using the notion of Atomic Decomposition, we partition BioPortal ontologies into logically coherent subsets (atoms), which are related to each other by a notion of dependency. We analyze various aspects of the resulting structures, and discuss their implications on applications of ontologies. In particular, we describe and investigate the usage of these ontology decompositions to extract modules, for instance, to facilitate matchmaking of semantic Web services in SSWAP (Simple Semantic Web Architecture and Protocol). Descriptions of those services use terms from BioPortal so service discovery requires reasoning with respect to relevant fragments of ontologies (i.e., modules). We present a novel algorithm for extracting modules from decomposed BioPortal ontologies which is able to quickly identify atoms that need to be included in a module to ensure logically complete reasoning. Compared to existing module extraction algorithms, it has a number of benefits, including improved performance and the possibility to avoid loading the entire ontology into memory. The algorithm is also evaluated on BioPortal ontologies and the results are presented and discussed.

Keywords: OWL, modularity, atomic decomposition, semantic Web services, SSWAP

1 Introduction

State-of-the art biomedical ontologies, e.g., those provided by the NCBO BioPortal, are often maintained as monolithic collections of axioms in single files or

* This is an author version of the contribution to Proc. ISWC 2011. The original publication is available at www.springerlink.com.

[5] http://bioportal.bioontology.org/

in a few files. This is not ideal for applications which require access to individual fragments of ontologies, for example, axioms relevant for a particular term. One example is use of ontology terms in descriptions of Semantic Web services or requests for their discovery. In such cases it is undesirable to load the entire ontology into memory (or transfer it over the network) in order to reason about a limited signature.

Semantic Web Services, such as SSWAP[6] (Simple Semantic Web Architecture and Protocol [8]) or SADI (Semantic Automated Discovery and Integration [14]), offer particular challenges for monolithic ontologies. In this application, semantic Web services reference (and dereference) ontological terms at transaction time–often requiring only a few terms from numerous ontologies in order to complete a transaction between two agents. This creates two challenges specific to ontology decomposition and modularity: 1) Semantic Web services operate under both AAA (Anyone can say Anything about Anything[7]) and the OWA (Open World Assumption). Thus even if service providers had complete knowledge of all BioPortal ontologies before transaction, this could become incomplete at transaction time because service providers could be presented with new terms from new ontologies where said terms could imply arbitrarily complex relations with cached ontologies (e.g., class subsumption or equivalence). This implies new, on-demand reasoning, which places a premium on minimizing the size and complexity of relevant ontologies to those components necessary and sufficient for the transaction at hand; 2) memory and hard disk resources are not limiting for virtually all biological ontologies. But network bandwidth and latency is limiting: large, monolithic ontologies can exceed 10 Mbytes when serialized as RDF/XML. Therefore it is important to investigate the possibility of maintaining ontologies in a more flexible form which supports reasoning over small (from the network's or the reasoner's viewpoint) fragments.

This paper presents the first, to our knowledge, large-scale investigation into decomposability and modular aspects of the NCBO BioPortal ontologies and demonstrates that most of them can be split into *small* logically coherent parts (atoms), from which modules can be efficiently assembled before reasoning. We discuss such good (on average) decomposability of BioPortal ontologies, its implications for applications, and also comment on occasional poor decomposability (Section 3). Finally, we describe a novel algorithm for decomposition-based module extraction (and the auxiliary algorithm for computing minimal seed signatures) and present evaluation results in Section 4.

2 Modularity and Atomic Decomposition

We assume the reader to be familiar with OWL and the underlying Description Logics [1], and sketch here some of the central notions around locality-based modularity [2] and Atomic Decomposition [6]. We use $\mathcal{L}$ for a Description Logic,

[6] http://sswap.info

[7] For details see paragraph 2.2.6 of the "RDF: Concepts and Abstract Syntax' document at http://www.w3.org/TR/rdf-concepts/#section-anyone

e.g., $\mathcal{SHIQ}$, and $\mathcal{O}, \mathcal{M}$, etc., for an ontology, 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 class, property, and individual names used in α or in $\mathcal{O}$.

Given a set of terms, or *seed signature*, Σ, a Σ-module $\mathcal{M}$ based on deductive-Conservative Extensions [9] 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. Deciding if a set of axioms is a module in this sense is hard or even impossible for expressive DLs [12], but if we drop the minimality requirement we can define "good sized" approximations, as in the case of *syntactic locality*, or *locality* for short, which can be efficiently extracted. Such modules provide strong logical guarantees by capturing *all* the relevant entailments about Σ, despite not necessarily being minimal subsets of $\mathcal{O}$ with this property [11]. A module extractor is implemented in the OWL API.[8]

Given an ontology $\mathcal{O}$ and a seed signature Σ, we say that an axiom $\alpha \in \mathcal{O}$ is $\bot$-*local* w.r.t. Σ if we can "clearly identify" the result of replacing all terms in α not in Σ with $\bot$ as a tautology; see [2] for a formal definition. Then, a $\bot$-module for Σ contains all axioms that are non-$\bot$-local w.r.t. Σ, plus all those needed to preserve the meaning of terms occurring in these axioms. Similarly we can define $\top$-modules. Additionally, by nesting these two notions until a fixpoint is reached we obtain $\top\bot^*$-modules. Hence, 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; a $\top\bot^*$-module is a subset of both the corresponding $\top$- and $\bot$-modules, containing all entailments to imply that two classes in Σ are in the subclass relation, but not necessarily all 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.

In [6] we have introduced a new approach to represent the whole family $\mathfrak{F}^x_{\mathcal{O}}$ of locality-based x-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 have defined a notion of "logical dependence" as follows: an axiom α depends on another axiom β if, whenever α occurs in a module $\mathcal{M}$, then β belongs to $\mathcal{M}$, too. Next, we observe that, for each axiom α, the x-module for the signature $\widetilde{\alpha}$ is the smallest x-module containing α; we call *α-module* a module x-mod$(\widetilde{\alpha}, \mathcal{O})$ and denote it by $\mathcal{M}^x_\alpha$.

The dependence between axioms allows us to identify clumps of highly interrelated axioms that are never split across two or more modules [6]; these clumps are called *atoms*. More precisely, for $x \in \{\top, \bot, \top\bot^*\}$ an *x-atom* of an ontology $\mathcal{O}$ is a maximal subset of $\mathcal{O}$ which is either contained in, or disjoint with, any x-module of $\mathcal{O}$. 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, we drop it.

Since every atom is a set of axioms, and atoms are pairwise disjoint, the AD is a partition of the ontology $\mathcal{O}$. Hence, the number of atoms is at most linear

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

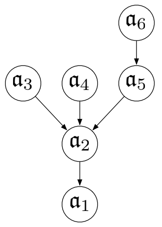
w.r.t. the size of $\mathcal{O}$. Moreover, atoms are the building blocks of all modules [7]. For an atom $\mathfrak{a} \in \mathcal{A}(\mathfrak{F}^x_{\mathcal{O}})$, the module $\mathcal{M}^x_{\mathfrak{a}} = x\text{-mod}(\widetilde{\mathfrak{a}}, \mathcal{O})$ is called *compact.*

Proposition 1. *Let* $\mathfrak{a}$ *be an atom in the AD* $\mathcal{A}(\mathfrak{F}^x_{\mathcal{O}})$ *of an ontology* $\mathcal{O}$ *and* $\alpha \in \mathfrak{a}$*; then, for any selection of axioms* $\mathcal{S} = \{\alpha_1, \ldots, \alpha_\kappa\} \subseteq \mathfrak{a}$ *we have that* $x\text{-mod}(\widetilde{\mathcal{S}}, \mathcal{O}) = \mathcal{M}^x_\alpha$*. In particular, for each* $\alpha_i \in \mathfrak{a}$*,* $\mathcal{M}^x_{\alpha_i} = \mathcal{M}^x_\alpha$*. Vice versa, if* $\mathcal{M}^x_\alpha = \mathcal{M}^x_\beta$*, then there exists some* $\mathfrak{a}$ *such that* $\alpha, \beta \in \mathfrak{a}$*.*

As a consequence of Prop. 1, the set of compact modules coincides with the set of α-modules, and we 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 dependence to atoms. 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 $\mathcal{M}_{\mathfrak{b}} \subseteq \mathcal{M}_{\mathfrak{a}}$. The dependence relation $\succeq$ on AD is a partial order (i.e., dependence is transitive, reflexive, and antisymmetric) and thus can be represented by means of a Hasse diagram, i.e. a graph showing the dependencies between its nodes. Moreover, $\succeq$ provides the basis for a polynomial-time algorithm for computing the AD, since they allow us to construct $\mathcal{A}(\mathfrak{F}^x_{\mathcal{O}})$ via α-modules only [6].

Given the Hasse diagram of an AD, it is easy to get all compact modules of an ontology by considering the *principal ideal of an atom* $\mathfrak{a}$, i.e. the set $(\mathfrak{a}] = \{\alpha \in \mathfrak{b} \mid \mathfrak{a} \succeq \mathfrak{b}\} \subseteq \mathcal{O}$.

Example 2. Consider the ontology $\{\alpha_1, \ldots, \alpha_7\}$ and its $\bot$-AD:
$\alpha_1 =$ '$\texttt{Animal} \sqsubseteq (= \texttt{1hasGender}.\top)$',
$\alpha_2 =$ '$\texttt{Animal} \sqsubseteq (\geq \texttt{1hasHabitat}.\top)$',
$\alpha_3 =$ '$\texttt{Person} \sqsubseteq \texttt{Animal}$',
$\alpha_4 =$ '$\texttt{Vegan} \equiv \texttt{Person} \sqcap \forall\texttt{eats}.(\texttt{Vegetable} \sqcup \texttt{Mushroom})$',
$\alpha_5 =$ '$\texttt{TeeTotaller} \equiv \texttt{Person} \sqcap \forall\texttt{drinks}.\texttt{NonAlcoholicThing}$',
$\alpha_6 =$ '$\texttt{Student} \sqsubseteq \texttt{Person} \sqcap \exists\texttt{hasHabitat}.\texttt{University}$',
$\alpha_7 =$ '$\texttt{GraduateStudent} \equiv \texttt{Student} \sqcap \exists\texttt{hasDegree}.\{\text{BA, BS}\}$'



Here the $\bot$-atoms in the AD contain the following axioms respectively: $\mathfrak{a}_1 = \{\alpha_1, \alpha_2\}$, $\mathfrak{a}_2 = \{\alpha_3\}$, $\mathfrak{a}_3 = \{\alpha_4\}$, $\mathfrak{a}_4 = \{\alpha_5\}$, $\mathfrak{a}_5 = \{\alpha_6\}$, $\mathfrak{a}_6 = \{\alpha_7\}$. The compact module for the atom $\mathfrak{a}_6$ is $\mathcal{M}_{\mathfrak{a}_6} = \mathfrak{a}_1 \cup \mathfrak{a}_2 \cup \mathfrak{a}_5 \cup \mathfrak{a}_6$.

Next, we are interested in modules that do not "fall apart", and thus can be said to have an internal logical coherence. A module is called *fake* if there exist two $\succeq$-uncomparable 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. Interestingly, the notions of α-modules, principal ideals of atoms, and genuine modules coincide [6], so from now on we refer to them simply as Genuine Modules (GMs). Note that fake modules are represented in the Hasse diagram of an AD as union of principal ideals of atoms; the converse does not hold: not all combinations of principal ideals of atoms are fake modules.

Whilst getting GMs is an easy task to perform via ADs, extracting a module for a general signature is more complicated. This happens because axioms can pull in a module terms that are not "strictly necessary" for them to be non-local. For example, only axiom α_4 in Ex. 2 is non-$\bot$-local w.r.t. $\Sigma = \{\texttt{Vegan}\}$. However, each module containing α_4 contains also α_1, α_2, and α_3, because in order to preserve the meaning of `Vegan` we need first to preserve the meaning of the other terms occurring in this axioms. To guarantee this condition, we need to enlarge Σ with the terms pulled in by relevancy, and then re-check the axioms against relevancy w.r.t. the new signature.

We formalize this idea as follows. We define a *minimal seed signature* for a module $\mathcal{M} = x\text{-mod}(\Sigma, \mathcal{O})$ to be a $\subseteq$-minimal signature Σ' such that $\mathcal{M} = x\text{-mod}(\Sigma', \mathcal{O})$. We denote the set of all minimal seed signatures of a module by $x\text{-}\mathtt{mssig}(\mathcal{M}, \mathcal{O})$. We call an atom $\mathfrak{a}$ *relevant for a signature* Σ if there exists $\Sigma' \in x\text{-}\mathtt{mssig}(\mathcal{M}_{\mathfrak{a}}, \mathcal{O})$ such that $\Sigma' \subseteq \Sigma$.

Proposition 3. *Let $x \in \{\bot, \top\}$ and Σ_0 the input signature. Let us consider $\mathcal{M}^x_0 = \{\alpha \in (\mathfrak{a}] \mid \mathfrak{a}$ is relevant for $\Sigma_0\}$ and, for $i \geq 1$, $\mathcal{M}^x_i = \{\alpha \in (\mathfrak{a}] \mid \mathfrak{a}$ is relevant for $\widetilde{\mathcal{M}^x_{i-1}} \cup \Sigma_0\}$. Then, the chain of inclusions $\mathcal{M}^x_0 \subsetneq \mathcal{M}^x_1 \subsetneq \ldots$ eventually stops, and denoted by $\mathcal{M}^x_*$ the fixpoint, we have that $\mathcal{M}^x_* = x\text{-mod}(\Sigma_0, \mathcal{O})$.*

The procedure described in Prop. 3 is equivalent to the standard extraction of a module only for the two notions $\top$ and $\bot$, because the $\top\bot^*$-AD only partially reflecting dependencies between atoms; see [3] for an example.

In summary, x-atoms and related genuine modules form a basis for all x-locality-based modules. Next, we analyse ADs of existing ontologies and discuss their decomposability.

3 Decomposability of BioPortal Ontologies

Decomposing ontologies into suitable parts is clearly beneficial when it comes to processing, editing, and analyzing them, or to reusing their parts. When ontologies are decomposed automatically, e.g., by computing an AD, it is interesting to discuss and evaluate the suitability of such decomposition for different scenarios, and whether all or which ontologies decompose "well", what it means to decompose well, and which properties of an ontology lead to "good" decomposability.

In this paper we discuss and evaluate the performance of ADs w.r.t. a specific task, i.e. Fast Module Extraction. Suitable application and maintainance scenarios for this task are, as stated before, semantic Web services, or SADI services. We prove that in these cases the AD is generally a good decomposition. On the other hand, "good decomposability" may have a different meaning in other scenarios.

A first such scenario, called Collaborative Ontology Development and Reuse, involves different ontology engineers working on different modules of an ontology. The aim is to minimize the risk of conflicts which could result from two or more ontology engineers making changes to logically related parts of the ontology (i.e., one engineer could be changing the semantics of terms used by another). Modularity provides the notion of "safety" which defines conditions under which there is no such risk [2]. We assume that each engineer works within *their* module and uses other terms in a safe way, and that modules different engineers work on do not overlap. Here, a fine-grained decomposition is desirable.

Another scenario, called Topicality for Ontology Comprehension, is based on the assumption that, in order to enable the understanding of what the ontology deals with, we can search for its "topics" and their interrelations [4]. In this case, a good decomposition should provide a "bird's-eye" view of the topical structure of an ontology. This means that a very fine-grained decomposition is undesirable because it does little to help understanding. On the other hand, large clumps of axioms could aggregate, hence hide, specific topical relations. In this scenario, a good decomposition should be only modestly fine-grained.

We now present the results of decomposing BioPortal ontologies w.r.t. our notions of locality. Due to space restrictions we present only summaries of this results, but full decompositions, spreadsheets with metrics and other data is available online at `http://tinyurl.com/modbioportal`.

The 3 notions of ADs we use are strongly related since $\top\bot^*$-AD is a refinement w.r.t. set inclusion of both $\bot$- and $\top$-AD, see [3]. As a consequence, we expect ontologies to have more, smaller $\top\bot^*$-atoms than $\bot$- or $\top$-atoms.

Proposition 4. *The* $\top\bot^*$-AD *is finer than both the* $\bot$-AD *and* $\top$-AD*, i.e., for any* $\top\bot^*$*-atom* $\mathfrak{a}$*, there exists a* $\bot$*-atom* $\mathfrak{b}$ *and a* $\top$*-atom* $\mathfrak{c}$ *with* $\mathfrak{a} \subseteq \mathfrak{b}$ *and* $\mathfrak{a} \subseteq \mathfrak{c}$*.*

The NCBO BioPortal ontology repository contains over 250 bio-medical ontologies, of which 218 are OWL or OBO ontologies. Among these, we filtered out those whose file was corrupted, those that do not contain any logical axioms, and some very large ontologies.[9] The result is a corpus of 181 ontologies, designed and built by domain experts, that vary greatly in size and expressivity [10].

We have decomposed these 181 BioPortal ontologies according to all three notions of syntactic locality: $\bot$, $\top$, and $\top\bot^*$. For each decomposition, we compute a basic set of metrics: for each ontology, we compute the average and maximal size of atoms and Genuine Modules (GM) measured in numbers of axioms (axs. in the table), and then we take the average of the resulting numbers over all 181 ontologies. The results are presented in the following table.

Notion of locality	Average average axs./atom	Average maximum axs./atom	Average average axs./GM	Average maximum axs./GM	Average nr. of conn. components
$\top\bot^*$	1.73	86	66	143	826
$\bot$	2.19	93	73	156	45
$\top$	330.45	1,417	1,166	2,093	1.64

It can be seen that the $\top\bot^*$-AD is generally quite fine-grained: the average size of an atom is less than 2 axioms; indeed, only 54 ontologies out of 181 have at least one atom greater than 10 axioms. Next, $\bot$-AD is fairly, even suprisingly close in granularity to $\top\bot^*$-AD as the average atom is only slightly larger than 2 axioms, and all other metrics are surprisingly close. This remark is supported by the Spearman's coefficient [13] comparing the number of atoms per ontology in the $\bot$-AD with the one in the $\top\bot^*$-AD. It has a value of $\rho \cong 0.9946$, showing a strong, monotonic correlation between the two measures. Moreover, closer inspection reveals that these two ADs even coincide in 34/181 ontologies. This is interesting for FME, as we will see later.

[9] See the technical report [3] for statistics for ontologies with over 20K axioms.

In contrast, $\top$-AD is substantially coarser than both $\top\bot^*$ and $\bot$-ADs as the average atom is two orders of magnitude larger, and all other metrics are much larger as well. Given the nature of $\top$-locality [2], this is not surprising, and it supports our general understanding that $\top$-ADs are not a good choice when small size of atoms and modules are relevant. Also, observe that the connectivity of $\top\bot^*$-AD is much looser than that of the other two ADs: this reflects the fact that the dependency relation, for $\top\bot^*$-AD, only reflects one kind of dependency, which is the reason why a $\top\bot^*$ version of Prop. 3 does not hold.

In the majority of the ontologies investigated, we observe rather good decomposability in terms of atom size. There are, however, ontologies that contain abnormally huge atoms even for $\top\bot^*$-AD, e.g., over 6K axioms. This is of concern since a module of these ontologies is likely to be of at least that size. For example, in the context of Web services, an attempt to discover a service whose description uses terms from such an atom may require transmitting and reasoning with thousands of axioms, which is undesirable. We observe these huge atoms both in absolute terms, i.e., with more than 200 axioms, and in relative terms, i.e., with more than 50% of axioms of the ontology. In the following table, we list ontologies whose $\top\bot^*$-ADs have a huge atom, absolute, relative, or both. We report their size, the size of the maximal atoms, plus some other data that is explained in what follows.

Ontology $\mathcal{O}$ (ID in BioPortal)	$\#\mathcal{O}$	#max Atom	#Eq. axs.	#Disj. axs.
Nanoparticle Ontology (1083)	16,267	6,425	42	6,106
Breast Tissue Cell Lines Ontology (1438)	2,734	2,201	0	7
IMGT Ontology (1491)	1,112	729	38	594
SNP Ontology (1058)	3,481	598	30	210
Amino Acid Ontology (1054)	477	445	8	190
Comparative Data Analysis (1128)	804	434	8	190
Family Health History (1126)	1,091	378	0	1
Neural Electromagnetic Ontologies (1321)	2,286	259	21	0
Computer-based Patient Record Ontology (1059)	1,454	238	18	20
Basic Formal Ontology (1332)	95	89	13	41
Ontology of Medically-related Social Entities (1565)	138	100	17	41
Ontology for General Medical Science (1414)	194	102	17	41
Cancer Research and Mgmt Acgt Master (1130)	5,435	3,796	16	42

We carried out a preliminary investigation of ontologies with huge atoms, trying to understand the reasons for the existence of huge atoms. It turns out that some huge atoms are due to the abundance of Disjoint Covering Axioms (DCAs) and we assume that their abundance is due to a specific usage pattern of ontology editors. More precisely, one version of DCAs is a pair of axioms of the form $\{\mathtt{A} \equiv (\mathtt{B}_0 \sqcup \ldots \sqcup \mathtt{B}_n), \mathrm{PairwiseDisjoint}(\mathtt{B}_0, \ldots, \mathtt{B}_n)\}$. Since our notion of modularity is based on axioms and subsets of an ontology and is self-contained, any module that mentions $\mathtt{B}_i$ contains both axioms, and thus pulls in all axioms about $\mathtt{B}_j$ as well. When DCAs occur on many classes on all levels in the class hierarchy of an ontology, then this results, unsurprisingly, in a huge atom. Moreover, note that not only disjointness causes axioms to tie together, as the explicit covering axiom shows the same behaviour. For disjointness, however, this "pulling-in" effect does not occur if we rewrite the n-ary disjointness axiom into equivalent pairwise disjointness axioms or even make the disjointness implicit, as in the following example: $\{\mathtt{B}_0 \sqsubseteq \mathtt{A} \sqcap (= 0\mathtt{R}.\top), \ldots, \mathtt{B}_{n-1} \sqsubseteq \mathtt{A} \sqcap (= n-1\,\mathtt{R}.\top), \mathtt{B}_n \sqsubseteq \mathtt{A} \sqcap (\geq n\,\mathtt{R}.\top)\}$.

In the previous table, we see that ontologies with huge atoms often have a large number of DCAs in these atoms, as indicated by the number of equivalence class and disjointness axioms in the last two columns: e.g., in the first ontology, which also has the largest atom, almost all axioms in this atom are disjointness axioms; additionally, upon inspection, it turns out that some of the equivalence axioms in this atom are covering axioms involving 10 or more classes. Also, in the second ontology, even though the largest atom only contains 7 disjointness axioms, it turns out that one disjointness axiom contains 52 terms.

The numbers for Comparative Data Analysis and Amino Acid ontologies look very similar because the first ontology imports the second. Trivially, large atoms persist also in the imports closure of an ontology: they can only grow. This is particularly relevant for ontologies that are used as base for others. In our corpus, we indeed find such a basis, which causes other ontologies to decompose badly in the sense described above: the Basic Formal Ontology consists of 95 axioms, 89 of which form an atom, which is due to the abundant usage of DCAs. Among the "relative huge atoms" ontologies, two import the Basic Formal Ontology, and their decomposability is affected.

Other patterns also lead to huge atoms, and an investigation of possible patterns is part of future work.

The last remark about this data concerns its analysis under the viewpoint of scenarios different from semantic Web services. For Collaborative Ontology Development and Reuse, these results are promising since they show a seemingly good decomposability of ontologies for $\top\bot^*$-AD and $\bot$-AD, i.e., the existence of small, disjoint sets of axioms that can be safely updated in parallel. In contrast, in the Topicality for Ontology Comprehension scenario we observe that, when the number of atoms is comparable with the number of axioms, then atoms do not provide any summarization over axioms and we cannot hope that considering atoms can provide any summarization benefit. In this case, the atoms reflect only very fine-grained topics of an ontology [4]. However, the dependency structure reflects the logical dependency between atoms, and thus can be used to consider, e.g., dependent components which, in turn, may better reflect the topics of an ontology. Of course, to really support ontology comprehension, we might have to consider "most relevant" atoms of an ontology [5] and, definitely, suitable labeling of modules. Both directions are part of future work.

4 Labeled Atomic Decomposition and Decomposition-Based Module Extraction

One particular application of atomic decomposition explored in this paper is module extraction. In this section we describe a module extraction algorithm, called FME for "Fast Module Extraction", which is (a) usually faster than the standard ME algorithm and (b) does not require loading the entire ontology into memory.

As explained in Section 2, every module is a union of atoms, however, not every union of atoms is a module. In general, it is non-trivial to determine which atoms the module for a given seed signature Σ consists of. In particular, a seemingly irrelevant atom, whose signature is disjoint with Σ, may turn out to be a part of the module. One way to help determining relevant atoms is to *label* them, i.e., associate them with extra information regarding seed signatures. In this paper we consider a particular kind of labels which, for each atom $\mathfrak{a}$, contains the set of the *Minimal Seed Signatures* $\mathsf{MSS}((\mathfrak{a}])$ (recall that each $(\mathfrak{a}]$ is a module).

Labelling each atom $\mathfrak{a}$ with the minimal seed signatures of its module $\mathsf{MSS}((\mathfrak{a}])$ can have several uses. First, every $\Sigma \in \mathsf{MSS}((\mathfrak{a}])$ can be regarded as a (minimal) topic that determines $(\mathfrak{a}]$ and $\mathfrak{a}$. In this sense, all MSSs of all atoms constitute all relevant minimal topics about which the ontology speaks. This can be exploited for comprehension. The case where atoms have too many MSSs—$(\mathfrak{a}]$ could have up to $2^{\#(\mathfrak{a}]}$ many—is the subject of a representation method that allows the adjustment of granularity and is deferred to future work. Second, the collection of all MSSs guides the extraction of a single module by suggesting possible topics (MSSs as inputs of the extraction algorithm). Again, the number of topics needs to be controlled by adjusting the granularity of the presentation.

4.1 Labeling Algorithm and Evaluation

First, we present an AD-driven algorithm for computing, for each atom $\mathfrak{a}$ in the decomposition, the set of its minimal seed signatures $\mathsf{MSS}((\mathfrak{a}])$. Currently, the algorithm is limited to $\top$ or $\bot$-locality. We plan to extend it to $\top\bot^*$-locality in the future.

Note: in Algorithm 1 the symbol $\cup^*$ means "union and minimization w.r.t. set inclusion". This operator guarantees that every set S of seed signatures does not contain Σ' if $\Sigma \subseteq \Sigma'$ for some $\Sigma \in S$. For example, $\{\Sigma_1, \Sigma_2\} \cup^* \{\Sigma_3, \Sigma_4\}$, where $\Sigma_2 \subset \Sigma_3$, is equal to $\{\Sigma_1, \Sigma_2, \Sigma_4\}$.

Algorithm 1 first computes the set $\mathsf{MGS}(\mathfrak{a})$ (minimal globalizing signatures) for all axioms in $\mathfrak{a}$ (Line 4). For an axiom α and a given notion of locality x, $\mathsf{MGS}(\alpha)$ is the set of all $\Sigma \subseteq \widetilde{\alpha}$ such that α is x-non-local w.r.t. Σ and α is x-local w.r.t. all proper subsets of Σ. For bottom atoms $\mathfrak{a}$ (i.e., atoms which do not depend on other atoms) the sets $\mathsf{MSS}((\mathfrak{a}])$ and $\mathsf{MGS}((\mathfrak{a}])$ coincide.

Now, every signature $\Sigma \in \mathsf{MGS}(\mathfrak{a})$ is necessarily a seed signature for $(\mathfrak{a}]$ but, unless $\mathfrak{a}$ is a bottom atom, is not necessarily minimal. The reason is that $\Sigma' \subset \Sigma$ could be a seed signature for a module $(\mathfrak{b}]$, for some atom $\mathfrak{b} \preceq \mathfrak{a}$ if $\Sigma \subseteq \Sigma' \cup \widetilde{(\mathfrak{b}]}$. In that case, informally, Σ' first "pulls" $(\mathfrak{b}]$ into the module (Σ' being a seed signature for $(\mathfrak{b}]$) and then the extended seed signature $\Sigma' \cup \widetilde{(\mathfrak{b}]}$ "pulls" the axioms of $\mathfrak{a}$ and the rest of $(\mathfrak{a}]$. With "extended seed signature", we mean the seed signature against which locality is checked at some iteration of the standard ME algorithm. Even worse, there could be MSSs for $(\mathfrak{a}]$ which are

Algorithm 1 Computing MSSs for a principal ideal

```
 1: Input: Ontology $\mathcal{O}$; its AD $x$-mod-AD, $x \in \{\top, \bot\}$; atom $\mathfrak{a}$
 2: Output: $\mathsf{MSS}(\mathfrak{a})$, the set of all MSSs for $(\mathfrak{a}]$
 3: $\mathsf{MSS}(\mathfrak{a}), \mathsf{PreMSS}(\mathfrak{a}) \ \leftarrow\ \emptyset$
 4: $\mathsf{MGS}(\mathfrak{a}) \ \leftarrow\ \bigcup^{*}_{\alpha \in \mathfrak{a}} \mathsf{MGS}(\alpha)$
 5: $\mathsf{DD}(\mathfrak{a}) \ \leftarrow$ the set of atoms that $\mathfrak{a}$ non-transitively depends on
 6: if $\mathsf{DD}(\mathfrak{a}) = \emptyset$ then
 7:    return $\mathsf{MGS}(\mathfrak{a})$
 8: end if
 9: for each $\mathfrak{b} \in \mathsf{DD}(\mathfrak{a})$ do
10:    $\mathsf{MSS}(\mathfrak{b}) \ \leftarrow$ recursively compute MSSs for $(\mathfrak{b}]$
11: end for
12: for each $\Sigma \in \mathsf{MGS}(\mathfrak{a})$ do
13:    $\mathsf{RC}_{\Sigma}(\mathfrak{a}) \ \leftarrow\ \{\mathfrak{b} \in \mathsf{DD}(\mathfrak{a}) \mid \Sigma \cap \widetilde{(\mathfrak{b}]} \neq \emptyset\}$
14:    for each $\{\mathfrak{b}_1, \ldots, \mathfrak{b}_n\} \in \wp(\mathsf{RC}_{\Sigma}(\mathfrak{a}))$ do
15:       $\Sigma_{\mathfrak{a}} \ \leftarrow\ \Sigma \setminus \bigcup_{i=1,\ldots,n} \widetilde{(\mathfrak{b}_i]}$
16:       for each $X \in \mathsf{MSS}(\mathfrak{b}_1) \times \cdots \times \mathsf{MSS}(\mathfrak{b}_n)$ do
17:          $\mathsf{PreMSS}(\mathfrak{a}) \ \leftarrow\ \mathsf{PreMSS}(\mathfrak{a}) \cup^{*} \{\Sigma_{\mathfrak{a}} \cup X\}$
18:       end for
19:    end for
20: end for
21: for each $\Sigma \in \mathsf{PreMSS}(\mathfrak{a})$ do
22:    $\mathsf{MSS}(\mathfrak{a}) \ \leftarrow\ \mathsf{MSS}(\mathfrak{a}) \cup^{*} \{\{\Sigma'\} \mid \Sigma' \subseteq \Sigma$ and $x$-mod$(\Sigma', \mathcal{O}) = (\mathfrak{a}]\}$
23: end for
24: return $\mathsf{MSS}(\mathfrak{a})$
```

not subsets of any signature in $\mathsf{MGS}(\mathfrak{a})$ – or not even subsets of $\widetilde{\mathfrak{a}}$, as illustrated in Example 5.

Example 5. Let $\mathcal{O} = \{\alpha, \beta, \gamma\}$ with $\alpha =$ '$\mathtt{A} \equiv \mathtt{B} \sqcap \mathtt{C}$', $\beta =$ '$\mathtt{B} \equiv \mathtt{D} \sqcup \mathtt{E}$', and $\gamma =$ '$\mathtt{C} \equiv \mathtt{F} \sqcup \mathtt{G}$)'. Then the following hold:

$$
\begin{aligned}
\bot\text{-mod}(\{\mathtt{A}\}, \mathcal{O}) &= \bot\text{-mod}(\{\mathtt{B}, \mathtt{C}\}, \mathcal{O}) & &= \{\alpha, \beta, \gamma\} \\
\bot\text{-mod}(\{\mathtt{B}\}, \mathcal{O}) &= \bot\text{-mod}(\{\mathtt{D}\}, \mathcal{O}) &= \bot\text{-mod}(\{\mathtt{E}\}, \mathcal{O}) &= \{\beta\} \\
\bot\text{-mod}(\{\mathtt{C}\}, \mathcal{O}) &= \bot\text{-mod}(\{\mathtt{F}\}, \mathcal{O}) &= \bot\text{-mod}(\{\mathtt{G}\}, \mathcal{O}) &= \{\gamma\}
\end{aligned}
$$

Therefore, there are three atoms $\mathfrak{a} = \{\alpha\}$, $\mathfrak{b} = \{\beta\}$, $\mathfrak{c} = \{\gamma\}$ with the dependencies $\mathfrak{b} \preceq \mathfrak{a}$ and $\mathfrak{c} \preceq \mathfrak{a}$. Now take the MSS $\{\mathtt{B}, \mathtt{C}\}$ for $\mathfrak{a}$ and replace $\mathtt{B}$ and $\mathtt{C}$, which occur in $\mathfrak{b}$ and $\mathfrak{c}$, with the MSSs $\{\mathtt{D}\}$ and $\{\mathtt{F}\}$ for $\mathfrak{b}$ and $\mathfrak{c}$. Then $\{\mathtt{D}, \mathtt{F}\}$ is an MSS for $(\mathfrak{a}] = \mathcal{O}$ although obviously $\{\mathtt{D}, \mathtt{F}\}$ is disjoint with $\widetilde{\mathfrak{a}}$ and with any member of $\mathsf{MGS}(\mathfrak{a})$.

Despite these complications, axioms of $\mathfrak{a}$ can only be pulled into the module once the extended seed signature includes at least one of the members of $\mathsf{MGS}(\mathfrak{a})$. The algorithm next recursively computes MSS for all direct children of $\mathfrak{a}$ (Line 10) and then proceeds to discover other MSSs of $(\mathfrak{a}]$ by combining the sets MSS for direct children of $\mathfrak{a}$ with the set $\mathsf{MGS}(\mathfrak{a})$ (Lines 12–20).

It does so by "elaborating" each $\Sigma \in \mathsf{MGS}(\mathfrak{a})$. It selects those atoms $\mathfrak{b} \preceq \mathfrak{a}$ which behave as described above, i.e., $\widetilde{(\mathfrak{b}]}$ overlaps with Σ. The set of all such direct children of $\mathfrak{a}$ w.r.t. Σ is stored as $\mathsf{RC}_{\Sigma}(\mathfrak{a})$ (Line 13). Then the algorithm removes from Σ (the signature being "elaborated") the terms in the "lower" atoms ($\bigcup_{i=1,\dots,n} \widetilde{(\mathfrak{b}_i]}$) and stores the result in $\Sigma_{\mathfrak{a}}$ (Line 15). Lines 16–18 go through all seed signatures X which are guaranteed to pull every atom in $\mathsf{RC}_{\Sigma}(\mathfrak{a})$. Then, $X \cup \Sigma_{\mathfrak{a}}$ is a seed signature (not necessarily minimal) for $(\mathfrak{a}]$, as explained above. All such $X \cup \Sigma_{\mathfrak{a}}$ are collected in $\mathsf{PreMSS}(\mathfrak{a})$.

The members $\Sigma \in \mathsf{PreMSS}(\mathfrak{a})$ are not guaranteed to be a *minimal* seed signature for $(\mathfrak{a}]$ because of possible weak dependencies between direct children of $\mathfrak{a}$. Informally, there could be a subset of Σ which first pulls some $\mathfrak{b}_i$, then some child of $\mathfrak{b}_i$ and only then $\mathfrak{b}_j$. Therefore, the algorithm has to "minimize" every $\Sigma \in \mathsf{PreMSS}(\mathfrak{a})$ by checking whether any of its subsets are, by themselves, already seed signatures of $(\mathfrak{a}]$ (Lines 21–23). However, entries of $\mathsf{PreMSS}(\mathfrak{a})$ are usually good approximations of truly minimal seed signatures; in particular, they are much better approximations than just the signature of $(\mathfrak{a}]$.

4.2 Properties of the Labeling Algorithm

The correctness of Algorithm 1 is established in [3]. It requires time exponential in the size of the ontology in the worst case, see the discussion in [3]. Despite the worst-case intractability the algorithm has the *anytime* property: the loops for elaborating (Lines 14–21) and minimizing (Line 21–23) a seed signature could be interrupted upon time-out, which will result in computing some subset of the MSS set for an atom.[10] This allows for practical approximations in the case when computing all MSS takes too long. We call atoms whose labels do not contain all MSS *dirty* (other atoms are called *clean*).

Dirty atoms require special handling during module extraction because their relevance may not be determinable due to missing of some MSS. In other words, if a dirty atom $\mathfrak{a}$ is not relevant to a signature, it could mean two things: first, the atom is not a part of the module or, second, the atom *is* part of the module but a seed signature, which would indicate the relevance of $\mathfrak{a}$, has not been computed due to the time-out. Therefore, in order for the FME algorithm to remain correct it is forced to include dirty atoms into the module even though they may be irrelevant. This means, in particular, that performance of the FME algorithm directly depends on whether the MSS algorithm has been able to compute all MSS for every atom. This is subject of the evaluation which we discuss next. The open-source Java implementation used for our experiments is available at `http://tinyurl.com/bioportalFME`.

4.3 Evaluation of the Labeling Algorithm

We evaluated the labeling algorithm on the same BioPortal ontologies as used in Sect. 3. The main goal of the evaluation is to assess the practical feasibility of

[10] Minimization has to be interrupted carefully to make sure that all produced signatures are minimal w.r.t. inclusion even though some signatures could be missing.

computing all MSS for atoms in the BioPortal ontologies. We set the time-out for computing labels for every atom to be 5 seconds, so the algorithm is guaranteed to finish in 5 times the number of atoms in seconds. The results are presented in the following table.

Total no. of ont.s	Avg. size of $\mathsf{MSS}(\mathfrak{a})$	Avg. number of terms in all $\mathsf{MSS}(\mathfrak{a})$	Max. size of $\mathsf{MSS}(\mathfrak{a})$	Number of ont. with dirty atoms	Max. number of dirty atoms
181	1.4	2.1	$4,252$	5	554

For the vast majority of ontologies (176 out of 181) the algorithm was able to compute all MSS for all atoms. Also, the average label size (that is, the number of MSSes per atom) and the average number of terms in all MSSes per atom are small: 1.4 and 2.1, respectively (when averaged first within an ontology then over all ontologies). This is yet another consequence of the simplicity of the BioPortal ontologies: their atoms are relevant to only a small number of terms which implies a small average number of atoms (and consequently, axioms) per module, see the next subsection. This observation might suggest that the BioPortal ontologies, in contrast to those examined by Del Vescovo et al. in [5], do not have exponential numbers of modules, but it is no firm evidence because it does not tell us about the asymptotic growth of their module numbers relative to their sizes.

Regarding the few ontologies with dirty atoms, they either do not decompose well or have an interesting property of the AD graph: certain atoms *non-transitively* depend on a high number of other atoms. Both reasons are true, e.g., for the Nanoparticle ontology, for which the MSS algorithm left 554 atoms dirty and managed to compute $1,019$ MSS sets for one atom, and the International Classification for Nursing Practice ontology (72 dirty atoms and $4,252$ MSS sets, respectively). We leave it for future research to investigate such cases, where a subset of an ontology turns out to be relevant for such a high number of distinct, but overlapping, seed signatures.

4.4 Fast Module Extraction Algorithm and Evaluation

Finally, we present a LAD-based FME algorithm, which extracts modules based on Prop. 3 (i.e., by examining MSS sets in labels), and its evaluation. Similarly to the labeling algorithm, the current version of the FME algorithm is restricted to $\top$- or $\bot$-locality.

The relevance check at Line 6 takes into account the possible dirtiness of an atom. More formally, the atom is *possibly relevant* to Σ if it is clean and there exists $\Sigma' \in \mathsf{MSS}(\mathfrak{a})$ such that $\Sigma' \subseteq \Sigma$ or it is dirty and $\widetilde{(\mathfrak{a}]} \cap \Sigma \neq \emptyset$ and there is no $\Sigma' \in \mathsf{MSS}(\mathfrak{a})$ such that $\Sigma \subset \Sigma'$.[11]

The FME algorithm has two important advantages over the standard ME algorithm. First, it should be faster for most of ontologies because it benefits from

[11] Observe that if a subset of $\mathsf{MSS}(\mathfrak{a})$ contains a proper superset of Σ, then, since all seed signatures are minimal, the full set $\mathsf{MSS}(\mathfrak{a})$ cannot contain a subset of Σ.

Algorithm 2 Atomic decomposition-based module extraction algorithm (FME)

```
Input: LAD for FME of an ontology $\mathcal{O}$, a seed signature $\Sigma$
Output: The module $x$-mod$(\Sigma, \mathcal{O})$, where $x \in \{\top, \bot\}$
$\mathcal{M} \leftarrow \emptyset$
repeat
    enlarged $\leftarrow$ false
    $\mathcal{M} \leftarrow \mathcal{M} \cup$ "all atoms that are possibly relevant to $\Sigma$"
    if $\widetilde{\mathcal{M}} \setminus \Sigma \neq \emptyset$ then
        enlarged $\leftarrow$ true
    end if
    $\Sigma \leftarrow \Sigma \cup \widetilde{\mathcal{M}}$
until enlarged = false
return $\mathcal{M}$
```

the labeled AD in two ways: i) it exploits labels to quickly detect relevant atoms, ii) once an atom $\mathfrak{a}$ is established to be relevant the corresponding module $(\mathfrak{a}]$ is added to the module without further checks. Second, it consumes substantially less memory since only relevant atoms (and their principal ideals) need to be loaded. The second advantage is especially important when modules are small comparing to the size of the ontology. This is the case with most of the BioPortal ontologies where the median module's size for small seed signatures is under 1%, as illustrated by the FME evaluation results, which we show next.

We ran the FME algorithm on the same set of BioPortal ontologies, which were used for decomposition and the labeling evaluation. Seed signatures are generated by a random selection of class names. For each size both FME and ME algorithms were run 100 times on different seed signatures and the results are averaged over all runs. The results are averaged over all 181 ontologies and presented in the following table. Correctness of the FME algorithm was also verified empirically by checking that the resulting modules contain all axioms extracted by the standard ME algorithm.[12]

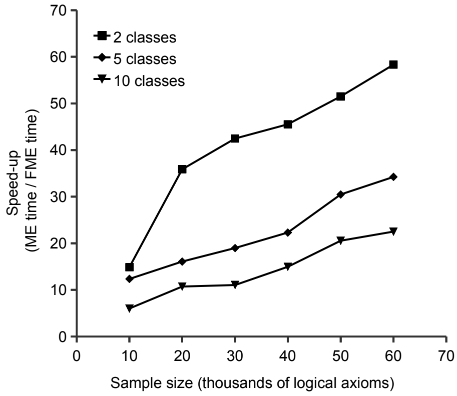
Size of seed sig.	Avg. (median) rel. module size (%)	Number of positive cases	Avg. ME runtime (ms)	Avg. FME speed-up	Max. FME speed-up
2	0.77 (0.04)	173	1.09	7.33	37.28
5	0.91 (0.08)	169	1.15	3.86	27.12
10	0.99 (0.13)	150	1.18	2.48	8.34

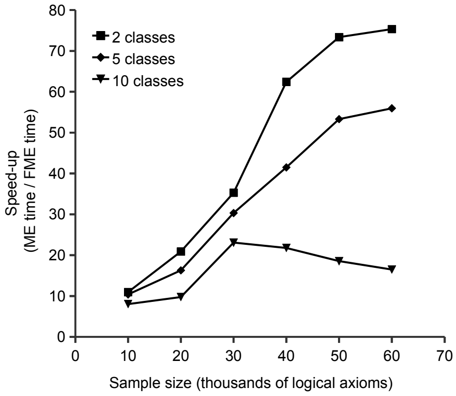
"Relative module size" = size of the module divided by the size of the ontology
"Positive cases" = ontologies for which FME is faster than ME
"Avg. (max.) speed-up" = average (max.) value of ME time divided by FME time

[12] The converse is only guaranteed to be true when there is no dirty atoms. Otherwise an FME module could be a superset (i.e., an approximation) of the ME module for the same seed signature. Of course, the irrelevant atoms can easily be removed by running the ME algorithm on the FME module, i.e., by refining the approximation.

Several conclusions can be drawn from the results. First, good decomposability of BioPortal ontologies indeed implies small modules on average (column 2). Second, even the standard ME is very fast (around 1 millisecond). Third, the FME algorithm is typically faster than the standard ME algorithm, however, this depends on several factors: i) decomposability of the ontology, ii) average number of atoms' labels, and iii) size of the seed signature. The first factor is important for both FME and ME algorithms as it effects the size of the module. The second factor determines how quickly the FME algorithm can perform the relevance check on an atom. In the worst case, the algorithm has to examine each MSS for an atom to decide if it is relevant.[13] The seed signature's size determines the number of relevant atoms. When the seed signature gets larger, the algorithm has to examine more atoms for relevancy. Finally, note that the results include ontologies with dirty atoms on which the FME algorithm could be up to 5 times slower than the ME algorithm because of considering possibly irrelevant atoms (this illuminates the importance of efficient labeling).

We also investigated the cases in which the FME algorithm runs an order of magnitude faster than the standard ME algorithm. This seems to be the case with ontologies which decompose into small atoms with a low number of MSS set, and small seed signatures. This is fairly typical for BioPortal ontologies, including some well-known ones. We illustrate this by comparing the running time of FME and ME on randomly generated samples of size between 10K and 60K axioms of GO (the Gene Ontology) and ChEBI (Chemical Entities of Biological Interest Ontology).[14] Seed signatures of size 2, 5 and 10 are generated as in the previous experiment. The results are shown in the two figures below (GO on the left, ChEBI on the right).





The graphs show that FME time tends to grow more slowly with the size of the ontology than ME time. This is unsurprising because the ratio of the module's size to the ontology size is decreasing (provided the seed signature's

[13] In fact, this depends on the data structure used to store sets of MSS. We use simple hash sets, so the check takes $O(|\mathsf{MSS}(\mathfrak{a})| \times |\Sigma|)$, where Σ is the seed signature.

[14] Both ontologies are slightly over 60K logical axioms.

size remains constant) and the FME is usually able to quickly locate relevant atoms while the ME algorithm has to examine each axiom.[15] As in the previous experiments, the speed-up is greater for smaller seed signatures. Note, however, that for seed signatures of 10 terms the relative speed-up of FME decreases after $30K$ axioms for ChEBI. Although it is still an order of magnitude, the behavior suggests that additional optimizations might be necessary for FME. For example, the relevance check could be made much quicker if MSS sets are stored in a data structure tuned for testing set inclusion.

In addition, the FME algorithm can work when only labels and the graph structure of the AD (but not axioms) are loaded into memory. This could be important for maintaining large ontologies, or even large collections of large ontologies, such as ontology repositories. In that case, contrary to the standard ME, the FME algorithm could still extract modules by loading axioms of only relevant atoms (plus possibly some dirty atoms for which irrelevance cannot be proved). For example, if BioPortal ontologies were maintained in the decomposed form, it would be possible to provide clients, such as SSWAP, with modules for a required seed signature in a scalable (from the memory perspective) way.

5 Summary and Future Directions

In this paper we have presented results of decomposing and extracting modules from most of BioPortal ontologies. We showed that the majority of ontologies decompose well, discussed possible reasons for poor decomposability, and implications of decomposability for possible use cases, in particular, semantic Web service annotation and discovery. In addition, we presented novel AD-based algorithms for computing minimal seed signatures for compact modules and module extraction.

Overall, the reported results show the utility of ontology modularity and decomposition for such tasks as semantic Web service matchmaking. In particular, it is likely that only small portion of a biomedical ontology is relevant for terms used in a Web service description, e.g., on SSWAP or SADI (see the average module size in the table on Page 13). Therefore, reasoning required to discover the service could be (efficiently) performed on a small set of OWL axioms. Furthermore, decomposition helps to get that set (module) faster than the standard module extraction and without the necessity to keep the ontology in memory.

We intend to continue our work on decomposition in several directions. First, we will investigate the possibility of maintaining ontologies in a decomposed form. This is more scalable from the memory perspective, enables faster ME, and is also potentially useful for comprehension and collaborative development of the ontology. However, it will require the possibility of incremental updates to the AD since its computation can be time consuming. Second, we will extend

[15] For space reasons our description of the FME algorithm does not show how labels serve as *indexes* by enabling us to perform the relevance test only on atoms whose MSS sets overlap with the seed signature. We must mention that a syntactic indexing (but coarser and less efficient) could be used for the standard ME as well.

our algorithms to $\top\bot^*$-modules and, possibly, to semantic locality. Third, we will keep on investigating modeling guidelines for developing well decomposable ontologies and will seek to improve understanding of poor decomposability.

Acknowledgements We thank the anonymous reviewers for their comments and Evan Lane for discussions with D.G. and A.W. This material is based upon work 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. Tech. rep. (2011), `http://tinyurl.com/modbioportal`
4. Del Vescovo, C., Parsia, B., Sattler, U.: Topicality in logic-based ontologies. In: Proc. of ICCS-11. pp. 187–200 (2011)
5. 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)
6. 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)
7. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The modular structure of an ontology: atomic decomposition. Tech. rep., University of Manchester (2011), available at `http://bit.ly/i4olY0`
8. Gessler, D., Schiltz, G.S., May, G.D., Avraham, S., Town, C.D., Grant, D.M., Nelson, R.T.: SSWAP: A simple semantic web architecture and protocol for semantic web services. BMC Bioinformatics 10, 309 (2009)
9. 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)
10. Horridge, M., Parsia, B., Sattler, U.: The state of bio-medical ontologies. In: Proc. of 2011 ISMB Bio-Ontologies SIG (2011)
11. 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)
12. 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)
13. Spearman, C.: The proof and measurement of association between two things. Amer. J. Psychol. 15, 72–101 (1904)
14. Wilkinson, M.D., Vandervalk, B., McCarthy, L.: SADI Semantic Web services – 'cause you can't always GET what you want! In: Proc. of APSCC. pp. 13–18 (2009)