The modular structure of an ontology: an empirical study*

Chiara Del Vescovo, Bijan Parsia, Uli Sattler, and Thomas Schneider

School of Computer Science, University of Manchester, UK
{delvescc,bparsia,sattler,schneider}@cs.man.ac.uk

1 Introduction

Why modularize an ontology? In software engineering, modularly structured systems are desirable, all other things being equal. Given a well-designed modular program, it is generally easier to process, modify, and analyze it and to reuse parts by exploiting the modular structure. As a result, support for modules (or components, classes, objects, packages, aspects) is a commonplace feature in programming languages.

Ontologies are computational artefacts akin to programs and, in notable examples, can get quite large as well as complex, which suggests that exploiting modularity might be fruitful, and research into modularity for ontologies has been an active area for ontology engineering. Recently, a lot of effort has gone into developing *logically sensible* modules, that is, modules which offer strong logical guarantees for intuitive modular properties. One such guarantee is called *coverage* and means that the module captures all the ontology's knowledge about a given set of terms (signature)—a kind of dependancy isolation. This guarantee is provided by modules based on conservative extensions, but also by efficiently computable approximations, such as locality-based modules.

The task of extracting one such module given a signature, which we call GetOne in this section, is well understood and starting to be deployed in standard ontology development environments (e.g., Protégé 4[1]). The extraction of locality-based modules has already been effectively used in the field for ontology reuse [1] as well as a subservice for incremental reasoning [2].

While GetOne is an important and useful service, it, by itself, tells us nothing about the modular structure of the ontology as a whole. The modular structure is determined by the set of *all* modules, or at least a subset thereof. We call the task of a-posteriori determining the modular structure of an ontology GetAll. While GetOne is well-understood and often computationally cheap, GetAll has hardly been examined for module notions with strong logical guarantees, with the work described in [3] being a promising exception. GetOne also requires the user to know in advance the right set of terms to input to the extractor: we call this a *seed* signature for the module and note that one module can have several such seed

* This work has been supported by the UK EPSRC grant no. EP/E065155/1.

[1] http://owl.cs.manchester.ac.uk/modularity/

signatures. Since there are non-obvious relations between the final signature of a module and its seed signature, users are often unsure how to generate a proper request and confused by the results. If they had access to the overall modular structure of the ontology determined by GetAll, they could use it to guide their extraction choices. In general, supported by the experience described in [3], we believe that, by revealing the modular structure of an ontology, we can obtain information about its topicality, connectedness, structure, superfluous parts, or agreement between actual and intended modeling.

In the worst case, the number of *all* modules of an ontology is exponential in number of terms or axioms in the ontology, in fact in the minimum of these numbers. Thus, it is possible that all realontologies have too many modules to extract all of them, even if an optimized extraction methodology were at hand. Even with only polynomially many modules, there may be too many for direct user inspection. Then, some other form of analysis would have to be designed.

In this paper, we report on experiments to obtain or estimate this number and to evaluate the modular structure of an ontology where we succeeded to compute it.

Related work. One solution to GetAll is obtained in [3,4] via partitions related to $\mathcal{E}$-connections. The resulting modules are disjoint, and this technique is of limited applicability—when it suceeds, it divides an ontology into three kinds of modules: (A) those which import vocabulary from others, (B) those whose vocabulary is imported, and (C) isolated parts. In experiments and user experience, the numbers of parts extracted were quite low and often corresponded usefully to user understanding. For instance, the tutorial ontology Koala, consisting of 42 logical axioms, is partitioned into one A-module about animals and three B-modules about genders, degrees and habitats.

It has also been shown in [3] that certain combinations of these parts provide coverage. For Koala, such a combination would still be the whole ontology. In general, partitions were observed to be too coarse grained; sometimes extraction resulted in a single partition even though the ontology seemed well structured. Furthermore, the robustness properties of the parts (e.g., under vocabulary extension) are not as well-understood as those of locality-based modules. However, partitions share efficient computability with locality-based modules.

Another approach to GetAll is described in [5]. It underlies the tool ModOnto, which aims at providing support for working with ontology modules that is similar to, and borrows intuitions from, software modules. This approach is logic-based and a-posteriori but, to the best of our knowledge, it has not been examined whether such modules provide coverage in the above sense. Furthermore, ModOnto does not aim at obtaining *all* modules from an ontology.

Another procedure for partitioning an ontology is described in [6]. However, this method only takes the concept hierarchy of the ontology into account and can therefore not provide the strong logical guarantee of coverage.

Among the a-posteriori approaches to GetOne, some provide logical guarantees such as coverage, and others do not. The latter are not of interest for this paper. The former are usually restricted to DLs of low expressivity, where decid-

ing conservative extensions—which underly coverage—is tractable. Prominent examples are the module extraction feature of CEL [7] and the system MEX [8]. However, we aim at an approach that covers DLs up to OWL 2.

There are a number of logic-based approaches to modularity that function a-priori, i.e., the modules of an ontology have to be specified in advance by features that are added to the underlying (description) logic and whose semantics is well-defined. These approaches often support distributed reasoning; they include C-OWL [9], $\mathcal{E}$-connections [10], Distributed Description Logics [11], and Package-Based Description Logics [12]. Even in these cases, however, we may want to understand the modular structure of the syntactically delineated parts. Furthermore, with imposed structure, it is not always clear that that structure is correct. Decisions about modular structure have to be taken early in the modeling which may enshrine misunderstandings. Examples were reported in [3], where user attempts to capture the modular structure of their ontology by separating the axioms into separate files were totally at odds with the analyzed structure.

Overview of the experiments and results. In the following, we will report on experiments performed to extract *all* modules from several ontologies as a first solution candidate for GetAll. We have considered three notions of modules based on syntactic locality—they all provide coverage, but differ in the size of the modules and in other useful properties of modules, see [13]—and extracted such modules for all subsets of the terms in the respective ontology. At this stage, we are mainly interested in module *numbers* rather than sizes or interrelations: the main concern is whether the suspected combinatorial explosion occurs. In order to test the latter, we have sampled subsets of each ontology and performed a full modularization on each subontology, measuring the relation between module number and subontology size for each ontology. We have also tried different approaches to reduce the number of modules to the most "interesting" ones.

An extended version of this paper and additional material for the evaluation of the experiments, such as spreadsheets and charts, are available online [14,15].

2 Preliminaries

Underlying description logics. We assume the reader to be familiar with OWL and the underlying description logics (DLs) [16,17]. We consider an ontology to be a finite set of axioms, which are of the form $C \sqsubseteq D$ or $C \equiv D$, where C, D are (possibly complex) concepts, or $R \sqsubseteq S$, where R, S are (possibly inverse) roles. Since we are interested in the logical part of an ontology, we disregard non-logical axioms. However, it is easy to add the corresponding annotation and declaration axioms in retrospect once the logical part of a module has been extracted. This is included in the publicly available implementation of locality-based module extraction in the OWL API.[2]

Let $\mathsf{N_C}$ be a set of concept names, and $\mathsf{N_R}$ a set of role names. A *signature* Σ is a set of terms, i.e., $\Sigma \subseteq \mathsf{N_C} \cup \mathsf{N_R}$. We can think of a signature

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

as specifying a topic of interest. Axioms that only use terms from Σ can be thought of as "on-topic", and all other axioms as "off-topic". For instance, if $\Sigma = \{\mathsf{Animal}, \mathsf{Duck}, \mathsf{Grass}, \mathsf{eats}\}$, then $\mathsf{Duck} \sqsubseteq \exists \mathsf{eats}.\mathsf{Grass}$ is on-topic, while $\mathsf{Duck} \sqsubseteq \mathsf{Bird}$ is off-topic.

Any concept or role name, ontology, or axiom that uses only terms from Σ is called a *Σ-concept*, *Σ-role*, *Σ-ontology*, or *Σ-axiom*. Given any such object X, we call the set of terms in X the *signature of X* and denote it with $\widetilde{X}$.

Conservative extensions and locality. Conservative extensions (CEs) capture the above described encapsulation of knowledge. A CE-based module for a signature Σ of an ontology $\mathcal{O}$ preserves all entailments η in $\mathcal{O}$ that can be formulated using symbols Σ only. For more precise definitions, see e.g., [18,19,14].

Unfortunately, CEs are hard or even impossible to decide for many DLs, see [20,18,13]. Therefore, approximations have been devised. We focus on *syntactic locality* (here for short: locality). Locality-based modules can be efficiently computed and provide coverage, that is, they capture *all* the relevant entailments, but not necessarily *only* those [21,22]. Although locality is defined for the DL $\mathcal{SHIQ}$, it is straightforward to extend it to $\mathcal{SHOIQ}(D)$ (see [21,22]), and the implementation of locality-based module extraction in the OWL API. We are using the notion of locality from [13, Def. 3].

It has been shown in [21] that $\mathcal{M} \subseteq \mathcal{O}$ and all axioms in $\mathcal{O} \setminus \mathcal{M}$ being $\bot$-local (or all axioms being $\top$-local) w.r.t. $\Sigma \cup \widetilde{M}$ is sufficient for $\mathcal{M}$ to be a CE-based module for Σ of $\mathcal{O}$. The converse does not hold in general.

It is described in [21] how to obtain modules of $\mathcal{O}$ for $\top$- and $\bot$-locality. We are using the notions of $\top$-, $\bot$-, $\top\bot^*$- and $\bot\top^*$-modules from [13, Def. 4]. That is, given an ontology $\mathcal{O}$, a *seed signature* Σ and a module notion $x \in \{\top, \bot, \top\bot^*, \bot\top^*\}$, we denote the x-module of $\mathcal{O}$ w.r.t. Σ by $x\text{-mod}(\Sigma, \mathcal{O})$. If we do not specify x, we generally speak of a *locality-based* module. Please note that Σ may contain terms outside $\widetilde{\mathcal{M}}$ and vice versa.

It is straightforward to show that $\top\bot^*\text{-mod}(\Sigma, \mathcal{O}) = \bot\top^*\text{-mod}(\Sigma, \mathcal{O})$ for each $\mathcal{O}$ and Σ. In contrast, $\top$- and $\bot$-modules do not have to be equal—in fact, the former are usually larger than the latter. Through the nesting, $\top\bot^*\text{-mod}(\Sigma, \mathcal{O})$ is always contained in $\top\text{-mod}(\Sigma, \mathcal{O})$ and $\bot\text{-mod}(\Sigma, \mathcal{O})$. Finally, we want to point out that, for $\mathcal{M} = x\text{-mod}(\Sigma, \mathcal{O})$, neither $\Sigma \subseteq \widetilde{\mathcal{M}}$ nor $\widetilde{\mathcal{M}} \subseteq \Sigma$ needs to hold.

The following property of locality-based modules will be of interest for our modularization. For $x \in \{\bot, \top\}$, Proposition 1 has been shown in [21]. The transfer to nested modules is straightforward.

Proposition 1. *Let $\mathcal{O}$ be an ontology, Σ, Σ' be a signatures, $x \in \{\bot, \top, \top\bot^*\}$; let $\mathcal{M} = x\text{-mod}(\Sigma, \mathcal{O})$ and $\Sigma \subseteq \Sigma' \subseteq \Sigma \cup \widetilde{\mathcal{M}}$. Then $x\text{-mod}(\Sigma', \mathcal{O}) = \mathcal{M}$.*

Genuine modules. In order to limit the overall number of modules, we introduce the notion of a *genuine module*. Intuitively, a given module $\mathcal{M}$ of an ontology is *fake* if it can be partitioned into a set $\{\mathcal{M}_1, \ldots, \mathcal{M}_n\}$ of smaller modules such that each relevant entailment of $\mathcal{M}$ is an entailment of some $\mathcal{M}_i$. All other modules are called *genuine*. See [14] for a precise definition and sufficient condition. In particular, if the whole ontology has a *partition* into modules,

then every entailment can be obtained from some of those modules, i.e., reasoning can be restricted to modules. Fake modules are uninteresting because $\mathcal{M}$ being fake means that different seed signatures of the $\mathcal{M}_i$ do not interact with each other. Given that often the overall number of modules appears to grow exponentially with the size of the subontology, a natural question arising is whether this is only caused by the fact that there are exponentially many fake modules.

3 Description of the experiments

Ontologies. We performed the experiments on several existing ontologies that we consider to be well designed and sufficiently diverse. We used Koala, Mereology, University, People, miniTambis, OWL-S, Tambis and Galen, whose sizes (axioms + terms) range from $42 + 25$ to $4{,}528 + 3{,}161$. See [14] for an overview of sizes and expressivities as well as an explanation of the choice criteria.

Full modularization. Let $\mathcal{O}$ be the ontology to be modularized. Our goal is to find all modules of $\mathcal{O}$, i.e., to compute $\{x\text{-mod}(\Sigma, \mathcal{O}) \mid \Sigma \in \widetilde{\mathcal{O}}\}$. In order to keep track of the seed signatures, we seek an algorithm which, given $\mathcal{O}$ as input, returns a representation of all pairs $(\Sigma, \mathcal{M})$ with $\Sigma \subseteq \widetilde{\mathcal{O}}$ and $\mathcal{M} = x\text{-mod}(\Sigma, \mathcal{O})$.

The most naïve procedure is to simply traverse through all seed signatures Σ, extract the corresponding module and add it to the output. Since there are exponentially many seed signatures, this is not feasible—even for Koala, 2^{25} runs of even the easiest test is unrealistic. Fortunately, we have good reasons to believe that there are significantly fewer modules than seed signatures in realistic ontologies: first, Proposition 1 says that, given the locality-based module $\mathcal{M} = x\text{-mod}(\Sigma, \mathcal{O})$, every seed signature Σ' that extends Σ and is a subset of $\Sigma \cup \widetilde{\mathcal{M}}$ yields the same module $\mathcal{M}$. Second, even if two seed signatures Σ and Σ' are not in such a relationship, the modules for Σ and Σ' can still coincide.

It should be noted, however, that there are very simple families of ontologies that already have exponentially many genuine modules, for instance the taxonomies $\mathcal{T}_n = \{B \sqsubseteq A\} \cup \{C_i \sqsubseteq B \mid i = 1, \ldots, n\}$, or the ontologies $\mathcal{O}_n = \{B_i \sqsubseteq A,\ C_i \sqsubseteq B_i \mid i = 1, \ldots, n\} \cup \{B_i \sqsubseteq \neg B_j \mid 1 \leqslant i < j \leqslant n\}$. More examples are given in [14]. However, we have not been able to construct any example with exponentially many genuine modules for inferred concept hierarchies of bounded width. In contrast, there are ontologies of arbitrary size that have exactly one module or at most quadratically many modules. Thus, while the worst case number of modules is high, it is not analytically impossible that real ontologies would have a reasonable number of modules. Unfortunately, empirically, as discussed in Section 4, this does not seem to be the case.

Since a module can have several seed signatures, we represent a module as a pair consisting of $\mathcal{M}$ and the set $\mathcal{S}$ of all *minimal* seed signatures Σ for which $\mathcal{M}$ is a module. Whenever a module for a new seed signature Σ' is to be computed, we first check whether Σ' satisfies $\Sigma \subseteq \Sigma' \subseteq \Sigma \cup \widetilde{\mathcal{M}}$ for some already extracted module $\mathcal{M}$ and some associated minimal seed signature Σ. Only if this is not the case, the module $\mathcal{M}' = x\text{-mod}(\Sigma', \mathcal{O})$ is computed. If $\mathcal{M}'$ coincides with some already extracted module $\mathcal{M}$, then Σ' is added to the set of minimal seed

Algorithm 1 Extract all x-modules

Input: an ontology $\mathcal{O}$ with signature $\widetilde{\mathcal{O}}$
Output: a set $\mathfrak{M} = \{(\mathcal{S}_1, \mathcal{M}_1), \ldots, (\mathcal{S}_n, \mathcal{M}_n)\}$ of all x-modules of $\mathcal{O}$, associated with their sets of minimal seed signatures (SSigs)

{*Start*: extract x-modules for all singleton SSigs}
$\mathfrak{M} \leftarrow \emptyset$
for all $t \in \widetilde{\mathcal{O}}$ **do**
  $\mathcal{M} \leftarrow$ extract x-module of $\mathcal{O}$ w.r.t. $\{t\}$
  call integrate$(\mathfrak{M}, \{t\}, \mathcal{M})$
end for

{*Extension*: iteratively add single terms to SSigs}
while $\mathfrak{M}$ contains $(\mathcal{S}, \mathcal{M})$ with marked $\Sigma \in \mathcal{S}$ **do**
  $(\mathcal{S}, \mathcal{M}) \leftarrow$ some elem. of $\mathfrak{M}$ with marked $\Sigma \in \mathcal{S}$
  $\Sigma \leftarrow$ some marked element of $\mathcal{S}$
  for all $t \in \widetilde{\mathcal{O}} \setminus (\Sigma \cup \widetilde{\mathcal{M}})$ **do**
    $\mathcal{M}' \leftarrow$ extract x-module of $\mathcal{O}$ w.r.t. $\Sigma \cup \{t\}$
    call integrate$(\mathfrak{M}, \Sigma \cup \{t\}, \mathcal{M}')$
  end for
  unmark Σ in $(\mathcal{S}, \mathcal{M})$
end while
return $\mathfrak{M}$

Algorithm 2 integrate$(\mathfrak{M}, \Sigma, \mathcal{M})$

for all $(\mathcal{S}', \mathcal{M}') \in \mathfrak{M}'$ **do**
  if $\mathcal{M} = \mathcal{M}'$ **then**
    $\mathcal{S}' \leftarrow \mathcal{S}' \cup \Sigma$
    mark Σ in $(\mathcal{S}', \mathcal{M}')$
    return
  end if
end for
$\mathfrak{M} \leftarrow \mathfrak{M} \cup (\{\Sigma\}, \mathcal{M})$
mark Σ in $(\{\Sigma\}, \mathcal{M}')$
return

signatures associated with $\mathcal{M}$; otherwise the pair $(\{\Sigma'\}, \mathcal{M}')$ is added to the set of extracted modules. This is performed by Algorithm 1, which calls Alg. 2. For soundness and completeness of Algorithm 1 and optimizations, see [14].

Sampling via subsets. In preliminary testing it soon became apparent that even our optimized algorithm would not reasonably terminate on even fairly small ontologies. Since we have a search space exponential in the size of the ontology and potentially exponentially many modules, it was not clear whether the problem was that our algorithm was not sufficiently optimized (so that the search space dominated) or that the output was impossible to generate. Since it is pointless to try to optimize an algorithm for a function whose output is exponentially large in the size of the typical input, it is imperative to determine whether real-world ontologies do have an exponential number of modules. This last question is one goal of the experiments described in this paper.

In order to test the hypothesis that real-life ontologies have an exponential number of modules, we have sampled subsets of different sizes from the ontologies considered. By fully modularizing each of these subsets, we can draw conclusions about the asymptotic relation between its size and the number of modules obtained. Randomly generated subsets would tend to contain unrelated

axioms, taken out of the context in which they have been included by the ontology developers. Since unrelated axioms, or ontologies with many unrelated terms, generally yield many modules, it would be harder to justify the hypothesis that real-world ontologies tend to have significantly less than exponentially many modules if we used arbitrary, less coherent subsets.

We have therefore chosen to let each subset be a module for a randomly generated signature—although we are aware that such subsets are more modular than necessary because ontologies are not normally developed modularly. But this is not a problem: it can only cause us to *understate* the number of modules.

We have sampled 10 signatures of each size between 0 and a threshold of 50 (or ontology's signature size if that was smaller). In some cases where the subset sizes were not optimally distributed (e.g., when small subsets were missing), we sampled 30 signatures of each size. For these signatures, we have extracted the $\top\bot^*$-modules, excluding duplicates, and ordered them by size. Then we have fully modularized all subsets in descending order, aborting when a single modularization took longer than a preset timeout of 20, 60 or 600 minutes, see Section 4 for an explanation of that choice. For each subset, we counted the number of all modules and of its genuine modules. See [14] for computer specifications.

4 Results

Module numbers for full modularization. Figure 1 shows the full modularization of Koala and Mereology for the module types $\top$, $\bot$ and $\top\bot^*$. In the case of $\top\bot^*$, we also determined genuine modules, denoted by $\top\bot^*_g$. In addition to the number of modules, we have listed the runtime and four aggregations of module sizes, where "size" refers to the number of logical axioms. Since the number of axioms is a syntax-dependent measure, we plan to include other measures, such as the number of terms and the sum of the sizes of all axioms, in future work.

	Koala				Mereology			
	$\top$	$\bot$	$\top\bot^*$	$\top\bot^*_g$	$\top$	$\bot$	$\top\bot^*$	$\top\bot^*_g$
#Modules	**12**	**520**	**3,660**	**2,143**	**40**	**552**	**1952**	**272**
Time [s]	*0*	*1*	*9*	*34*	*0*	*6*	*158*	*158*
Min size	29	6	0	0	18	0	0	0
Avg size	35	27	23	23	26	25	20	22
Max size	42	42	42	42	40	40	40	38
Std. dev.	4	6	6	6	6	7	8	8

$\top\bot^*_g$ = *genuine* $\top\bot^*$ modules. "Size" = number of logical axioms.

Figure 1. Full modularization of Koala and Mereology

For both ontologies, the number of modules increases from $\top$- via $\bot$- to $\top\bot^*$ modules as expected: as mentioned before, $\top$-modules tend to be bigger,

and therefore more modules coincide in this case. However, $\top$-modules are too coarse-grained: most of them comprise almost the whole ontology, and all have a size of at least 29 (69% of Koala) or 18 (41% of Mereology).

The extracted $\bot$-modules yield a more fine-grained picture, although all their sizes for Koala are still above 6 (14%). We already pay for this with an increase in the number of modules by a factor of more than 43 (Koala) and 14 (Mereology). With $\top\bot^*$, smaller modules are included, but for the price of another increase in module numbers by a factor of 7 (Koala) and 3.5 (Mereology). For a more fine-grained modularization, we also pay with an increased extraction time. See [14] for comments on the observed differences between Koala and Mereology.

Attempts to fully modularize ontologies larger than Koala and Mereology with the described algorithm did not succeed. We cancelled each such computation after several hours, when thousands of modules have been extracted.

Although Koala and Mereology have much fewer modules than the theoretical upper bound of 2^{25}, we still get too many for (regular) inspection by ontology users. We have therefore tried two more ways to reduce their modules to fewer "interesting" ones. Both approaches showed no significant impact, see [14].

Module numbers for subset sampling. After carrying out the subset sampling technique described in Section 3, we are strongly convinced that most of the ontologies examined exhibit the feared exponential behavior. Figure 2 shows scatterplots of the number of $\top\bot^*$ modules (genuine $\top\bot^*$ modules) versus the size of the subset for People and Koala. Each chart shows an exponential trendline, which is the least-squares fit through the data points by using the equation $m = ce^{bn}$, where n is the size of the subset, m is the number of modules, e is the base of the natural logarithm, and c, b are constants. This equation and the corresponding determination coefficient (R^2 value) are given beneath each chart. Spreadsheets with the underlying data, as well as spreadsheets and charts for the other ontologies, can be found at [15]. The R^2 values and trendline equations for the examined ontologies are summarized in Figure 3, where we also included the estimated number of modules for the full ontology as per the equation, the timeout used and the overall runtime.

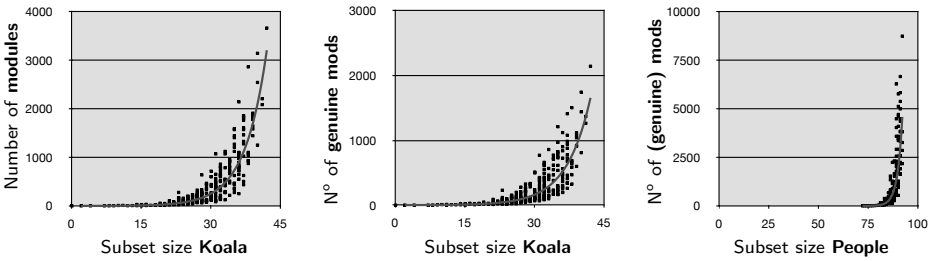

Figure 2. Numbers of modules versus subset sizes for Koala and People

The scatterplots and determination coefficients for the first six ontologies in Figure 3 provide strong evidence that the number of modules depends expo-

Ontology	Confidence R^2_{m}	R^2_{g}	Trendline equation m	g	Estimate m	g	Timeout [min]	Runtime [min]
People	.95	.95	$2 \cdot 10^{-13} e^{.41n}$		10^6	10^6	20	148
Mereology	.87	.94	$1.2e^{.16n}$	$1.1e^{.13n}$	10^3	10^2	—	4
Koala	.90	.88	$.45e^{.21n}$	$.50e^{.19n}$	10^3	10^3	—	4
Galen	.94	.86	$1.2e^{.24n}$	$1.6e^{.16n}$	NaN	NaN	60	288
University	.84	.83	$1.7e^{.19n}$	$1.6e^{.14n}$	10^4	10^3	20	354
OWL-S	.82	.84	$.0027e^{.17n}$	$.0032e^{.16n}$	10^{17}	10^{17}	60	73
Tambis	.75	.70	$1.1e^{.22n}$	$1.4e^{.13n}$	10^{58}	10^{33}	600	681
miniTambis	.47	.52	$2.6e^{.18n}$	$2.5e^{.14n}$	10^{14}	10^{10}	600	963

m, g	$\top\bot^*$ modules, genuine $\top\bot^*$ modules
R^2_{m}, R^2_{g}	Determination coefficient of fitted trendlines
Estimate	Module numbers for full ontology as per trendline
NaN	Estimate is larger than 10^{142}

Figure 3. Witnesses for exponential behavior

nentially on the size of the subset. In most cases, the exponential behavior was observable with at most a 60-minute timeout.

Weight analysis for Koala. Even if we consider only genuine modules, there are ontologies that have exponentially many of them. In order to focus on even fewer, "interesting" modules, we have devised the measures *cohesion* and *pulling power*. Thy are based on the number of seed signatures (SSigs) of a module $\mathcal{M}$ and the number of terms in $\widetilde{\mathcal{M}}$. An SSig Σ of $\mathcal{M}$ is called *minimal* (MSSig) if there is no signature $\Sigma' \subset \Sigma$ that is an SSig of $\mathcal{M}$. If we ignore terms not present in the module, we speak of a *real MSSig* for $\mathcal{M}$: this is a signature $\Sigma' = \Sigma \cap \widetilde{\mathcal{M}}$ where Σ is an MSSig for $\mathcal{M}$. Let r, s, m be the number of real MSSigs for $\mathcal{M}$, the size of the smallest MSSig for $\mathcal{M}$, and the size of $\widetilde{\mathcal{M}}$.

The *cohesion* of $\mathcal{M}$ measures how strongly the terms in $\mathcal{M}$ are held together, as indicated by the number of seed signatures for $\mathcal{M}$. More precisely, the cohesion of $\mathcal{M}$ is defined to be the ratio r/s. The *pulling power* of $\mathcal{M}$ measures how many terms are needed in an MSSig to "pull" all terms into $\mathcal{M}$ that we find there. We define the *pulling power* of $\mathcal{M}$ to be the ratio m/s.

As a first draft, we define the *weight* of a module $\mathcal{M}$ to be the product of its cohesion and pulling power: $w = \frac{r \cdot m}{s^2}$. We computed the weight of all 3660 modules of Koala. The 11 heaviest modules and their set differences yield a partition of almost the whole ontology into 10 parts, each of which consists of terms that intutively form a topic (subconcepts included): Animal; Person and isHardWorking; Student; Parent; Student; Koala and Marsupial; TasmanianDevil; Quokka; Habitat; Degree; Gender. These topics reflect the core parts of the ontology. Those axioms that do not occur among the heaviest modules tend to be those that we intuitively would call less important for the ontology, for instance RainForest $\sqsubseteq$ Forest.The first 11 modules cover almost all of Koala's logical axioms (39 out of 42), and all axioms are covered from the 34th heaviest module on. The first 19 heaviest modules are also genuine.

The next step will be to refine this measure and apply it to more ontologies. Since we cannot expect to fully modularise ontologies bigger then Koala, we will need to find ways to extract heavy-weight modules separately.

5 Discussion and outlook

The fundamental conclusion is clear: the number of modules, even when we restrict our attention to genuine modules, is exponential in the size of the ontology for real ontologies. Our most reasonable estimates of the total number of modules in small to midsize ontologies (i.e., anything over 100 axioms) show that full modularization is practically impossible. As we are computing locality based modules, which tend to be larger than conservative extension based modules, our results give us a lower bound on the number of modules.

It is, of course, possible that there are principled ways to reduce the target number of modules. We could use a coarser approximation, though that would be hard to justify on logical grounds. Attempts to use "less minimal" modules or to heuristically merge modules have exhibited bad behavior, with a strong tendency to collapse to very few modules that comprise most of the ontology.

We believe that this conclusion is robust, even with the failure of our experiments on Tambis and miniTambis to uncover exponential behavior. As we said in Section 4, our expectation is that a longer timeout will reveal the problematic behavior. We also suspect a connection between the relatively low number of modules for these two ontologies and the fact that they have a large number of unsatisfiable concepts. For details, see [14]. The ratio between genuine and fake modules can be seen as a measure of axiomatic richness, at least indicating how strongly the axioms in the ontology connect its terms: the fewer of its modules are fake, the more "mutually touching" its terms are.

Attempts at estimating the module number statistically were unhelpful too. We could randomly draw a small number of seed signatures, compute their modules and use that number to estimate the number of all modules. We convinced ourselves using elementary stochastics that we cannot get a confident estimate by sampling only a small proportion of all seed signatures. See [14] for details.

While the outcome of the experiments is discouraging from the point of view of using the complete modularization in order to analyze the ontology, it does suggest several interesting lines of future work. First, we have already seen several features of ontologies that correlate well with a large or small number of modules. However, except for the phenomenon seen in Mereology, we do not have a verified explanation. Thus, for example, we need to get a precise picture of the relationship between justificatory and modular structure. Second, even if we cannot compute all modules, we may be able to compute a better approximation of their number. Given that signature sampling did not seem to help, we intend to explore sources of module number increase or reduction, such as the shape of the inferred concept hierarchy and patterns of axioms. Methodologically, it seems that artificial ontologies should be used, e.g., for confirmation of the relationship between justificatory structure and module number.

References

1. Jimeno, A., Jiménez-Ruiz, E., Berlanga, R., Rebholz-Schuhmann, D.: Use of shared lexical resources for efficient ontological engineering. In: SWAT4LS-08. Volume 435 of CEUR. (2008)
2. Cuenca Grau, B., Halaschek-Wiener, C., Kazakov, Y.: History matters: Incremental ontology reasoning using modules. In: Proc. of ISWC/ASWC-07. Volume 4825 of LNCS. (2007) 183–196
3. Cuenca Grau, B., Parsia, B., Sirin, E., Kalyanpur, A.: Modularity and web ontologies. In: Proc. of KR-06. (2006) 198–209
4. Cuenca Grau, B., Parsia, B., Sirin, E.: Combining OWL ontologies using $\mathcal{E}$-connections. JWebSem **4**(1) (2006) 40–59
5. Bezerra, C., de Freitas, F.L.G., Zimmermann, A., Euzenat, J.: ModOnto: A tool for modularizing ontologies. In: Proc. WONTO-08. Volume 427 of CEUR. (2008)
6. Stuckenschmidt, H., Klein, M.: Structure-based partitioning of large concept hierarchies. In: Proc. of ISWC-04. Volume 3298 of LNCS. (2004) 289–303
7. Suntisrivaraporn, B.: Module extraction and incremental classification: A pragmatic approach for $\mathcal{EL}^+$ ontologies. In: Proc. of ESWC-08. Volume 5021 of LNCS. (2008) 230–244
8. Konev, B., Lutz, C., Walther, D., Wolter, F.: Logical difference and module extraction with CEX and MEX. In: Proc. of DL 2008. Volume 353 of CEUR. (2008)
9. Stuckenschmidt, H., van Harmelen, F., Bouquet, P., Giunchiglia, F., Serafini, L.: Using C-OWL for the alignment and merging of medical ontologies. In: Proc. KR-MED. Volume 102 of CEUR. (2004) 88–101
10. Kutz, O., Lutz, C., Wolter, F., Zakharyaschev, M.: E-connections of abstract description systems. Artificial Intelligence **156**(1) (2004) 1–73
11. Borgida, A., Serafini, L.: Distributed description logics: Assimilating information from peer sources. J. Data Semantics **1** (2003) 153–184
12. Bao, J., Voutsadakis, G., Slutzki, G., Honavar, V.: Package-based description logics. [23] 349–371
13. Sattler, U., Schneider, T., Zakharyaschev, M.: Which kind of module should I extract? In: DL 2009. Volume 477 of CEUR. (2009)
14. Parsia, B., Sattler, U., Schneider, T.: The modular structure of an ontology: an empirical study. Technical report, University of Manchester `http://www.cs.man.ac.uk/%7Eschneidt/publ/modstrucreport.pdf` .
15. Materials: `http://owl.cs.manchester.ac.uk/modproj/meat-experiment` .
16. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. JWebSem **1**(1) (2003) 7–26
17. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Proc. of KR-06. (2006) 57–67
18. Konev, B., Lutz, C., Walther, D., Wolter, F.: Formal properties of modularization. [23] 25–66
19. Lutz, C., Walther, D., Wolter, F.: Conservative extensions in expressive description logics. In: Proc. of IJCAI-07. (2007) 453–458
20. Ghilardi, S., Lutz, C., Wolter, F.: Did I damage my ontology? A case for conservative extensions in description logics. In: Proc. of KR-06. (2006) 187–197
21. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. Artif. Intell. Res. **31** (2008) 273–318
22. 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. Volume 5021 of LNCS. (2008) 185–199
23. Stuckenschmidt, H., Parent, C., Spaccapietra, S., eds.: Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization. Volume 5445 of LNCS. Springer (2009)