Module Extraction and Importing

The problem of module extraction can be phrased as follows: given a subset of Σ of the vocabulary of an ontology, find a (minimal) subset of that ontology that is "relevant" for the terms in Σ. There are different approaches to "relevance" for Σ; they can be grouped into structural ones, and logic ones.

We are interested in *logic-based modules*, for which relevance amounts to entailment (or model) preservation over a signature Σ. That is, when we say that a subset $\mathcal{M}$ of an ontology $\mathcal{O}$ "is relevant for" the terms in Σ, we mean that all consequences of $\mathcal{O}$ that can be expressed over Σ are also consequences of $\mathcal{M}$. Then $\mathcal{O}$ is said to be a conservative extension (CE) of $\mathcal{M}$.

Observe that the logical view appears to be theoretically sound and elegant and guarantees that by reusing only terms from Σ one is not able to distinguish between importing $\mathcal{M}$ and importing $\mathcal{O}$ into some ontology $\mathcal{P}'$. However, the decision problem associated to conservative extension is difficult in general: deciding whether $\mathcal{O}$ is a Σ-conservative extension of $\mathcal{O}'$ is 2NEXPTIME-complete for $\mathcal{ALCQI}$ (roughly OWL-Lite) and undecidable for OWL-DL. For model conservative extensions the problem is highly undecidable (non recursively enumerable), even for $\mathcal{ALC}$. These computational obstacles have lead, e.g., to approximations via locality. Locality-based modules provide all logical guaranties of CE-based modules, and although they are not minimal, they have been shown to be useful for economically reusing ontologies.

In the remaining of this paper we will survey locality-based modules as well as other notions and approaches for module extraction.

Comparison of the various module notions can be carried out examining properties relevant for ontology reuse. The relevant properties have been identified in the literature and are described below:

Robustness under vocabulary restrictions. This property implies that a module of an ontology w.r.t. a signature Σ is also a module of this ontology w.r.t. any subset of Σ. This property is important because it means that we do not need to import a different module when we restrict the set of terms that we are interested in.

Robustness under vocabulary extensions. This implies that a module of an ontology $\mathcal{O}$ w.r.t. a signature Σ is also a module of $\mathcal{O}$ w.r.t. any $\Sigma' \supseteq \Sigma$ as long as $\Sigma' \setminus \Sigma$ does not share terms with $\mathcal{O}$. This means that we do not need to import a different module when extending the set of relevant terms with terms not from $\mathcal{O}$.

Robustness under replacement for a logic L. This property implies that if $\mathcal{M}$ is a module of $\mathcal{O}$ w.r.t. Σ, then the result of importing $\mathcal{M}$ into an L-ontology $\mathcal{O}'$ is a module of the result of importing $\mathcal{O}$ into $\mathcal{O}'$. This is called *module coverage* in the literature: importing a module does not affect its property of being a module.

Robustness under joins. If two ontologies are indistinguishable w.r.t. Σ and they share only terms from Σ, then each of them is indistinguishable from their union w.r.t. Σ. This property together with robustness under replacement, implies that it is not necessary to import two indistinguishable versions of the same ontology.

Conservative Extensions and Inseparability

We call the set of *terms* (class, property and individual names) that occur in an ontology $\mathcal{O}$ the *signature* of $\mathcal{O}$, and denote it by $\mathsf{Sig}(\mathcal{O})$. There are at least two different notions of conservative extension in the context of ontologies: Let $\mathcal{O}' \subseteq \mathcal{O}$ be two ontologies, Σ a signature and L a logic.

1. $\mathcal{O}$ is a *deductive Σ-conservative extension (Σ-dCE)* of $\mathcal{O}'$ w.r.t. L, if for every axiom α over L with $\mathsf{Sig}(\alpha) \subseteq \Sigma$, we have $\mathcal{O} \models \alpha$ iff $\mathcal{O}' \models \alpha$.

2. $\mathcal{O}$ is a *model Σ-conservative extension (Σ-mCE)* of $\mathcal{O}'$, if for every model $\mathcal{I}'$ of $\mathcal{O}'$, there exists a model $\mathcal{I}$ of $\mathcal{O}$ such that $\mathcal{I}'|_\Sigma = \mathcal{I}|_\Sigma$.

Note that the notion of model Σ-conservative extension is strictly stronger than the deductive one and it does not depend on the expressiveness of the language L. That is $\mathcal{O}$ being a Σ-mCE of $\mathcal{M}$ implies that $\mathcal{O}$ is a Σ-dCE of $\mathcal{M}$.

A more general notion to characterize modules is that of *(model-theoretic) inseparability.* Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be ontologies and Σ a signature. Then $\mathcal{O}_1$ and $\mathcal{O}_2$ are *Σ-model inseparable*, written $\mathcal{O}_1 \equiv^m_\Sigma \mathcal{O}_2$ if, $\{\mathcal{I}|_\Sigma : \mathcal{I} \models \mathcal{O}_1\} = \{\mathcal{I}|_\Sigma : \mathcal{I} \models \mathcal{O}_2\}$. Σ-model inseparability provides a very strong form of equivalence as $\mathcal{O}_1 \equiv^m_\Sigma \mathcal{O}_2$ guarantees that $\mathcal{O}_1$ can be replaced with $\mathcal{O}_2$ in any application that refers only to symbols from Σ. Indeed, $\mathcal{O}_1 \equiv^m_\Sigma \mathcal{O}_2$ if and only if

$$\mathcal{O}_1 \models \varphi \text{ iff } \mathcal{O}_2 \models \varphi$$

holds for any second order sentence φ using only symbols from Σ. Weaker versions of inseparability relations can be defined. For a given logic L, an *inseparability relation* is a family $S = \{\equiv^S_\Sigma |\ \Sigma \text{ is a signature}\}$ of equivalence relations on the set of L theories. The intuition behind this notion is as follows: $\mathcal{O}_1 \equiv^S_\Sigma \mathcal{O}_2$ means that $\mathcal{O}_1$ and $\mathcal{O}_2$ are indistinguishable w.r.t. Σ, i.e., they represent the same knowledge about the topic represented by Σ. The precise definition of the inseparability relation determines the exact meaning of the terms "indistinguishable" and "the same knowledge". $\mathcal{M}$ being a module for Σ of $\mathcal{O}$ should be equivalent to $\mathcal{M} \subseteq \mathcal{O}$ and $\mathcal{M}$ being inseparable w.r.t. Σ from $\mathcal{O}$. Under the requirement to preserve entailments the following inseparability relations can be defined (in the context of description logics):

- $\mathcal{O}_1$ and $\mathcal{O}_2$ are *Σ-concept inseparable*, in symbols $\mathcal{O}_1 \equiv^c_\Sigma \mathcal{O}_2$, if for all concept names A, B in Σ, it holds that $\mathcal{O}_1 \models A \sqsubseteq B$ if and only if $\mathcal{O}_2 \models A \sqsubseteq B$.

- $\mathcal{O}_1$ and $\mathcal{O}_2$ are *Σ-subsumption inseparable* w.r.t. a logic L, written $\mathcal{O}_1 \equiv^s_\Sigma \mathcal{O}_2$, if for all concept descriptions C, D in L over Σ, it holds that $\mathcal{O}_1 \models C \sqsubseteq D$ if and only if $\mathcal{O}_2 \models C \sqsubseteq D$.

It is easy to see that, for each signature Σ, it holds that $\equiv^m_\Sigma \subseteq \equiv^s_\Sigma \subseteq \equiv^c_\Sigma$.

Module Notions

Conservative extension and inseparability relations induce modules as follows. Let $\mathcal{O}$ be an ontology, $\mathcal{M} \subseteq \mathcal{O}$ and Σ a signature, and S[1] an inseparability relation. We call $\mathcal{M}$

1. a *Σ-module* of $\mathcal{O}$ induced by S if $\mathcal{M} \equiv^S_\Sigma \mathcal{O}$;

2. a *self-contained* *Σ-module* of $\mathcal{O}$ induced by S if $\mathcal{M} \equiv_{\Sigma \cup \mathsf{Sig}(\mathcal{M})} \mathcal{O}$;

3. a *depleting* *Σ-module* of $\mathcal{O}$ induced by S if $\mathcal{O} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{Sig}(\mathcal{M})} \emptyset$

Observe that modules induced by model-theoretic inseparability are mCE-based modules: for a given signature Σ, if $\mathcal{M} \subseteq \mathcal{O}$ and $\mathcal{M} \equiv^m_\Sigma \mathcal{O}$, then $\mathcal{O}$ is a Σ mCE of $\mathcal{M}$. In the remaining of this paper we will focus on modules induced by model-theoretic inseparability.

[1] e.g., $S \in \{m, s, c\}$

We say that a subset $\mathcal{M} \subseteq \mathcal{O}$ *covers* all the knowledge that $\mathcal{O}$ has about Σ, if for every axiom α with, $\mathsf{Sig}(\alpha) \subseteq \Sigma$, we have that $\mathcal{O} \models \alpha$ if and only if $\mathcal{M} \models \alpha$. Note that in that case $\mathcal{O}$ is a Σ dCE of $\mathcal{M}$.

We say that $\mathcal{O}$ is *safe* for Σ if, for every ontology $\mathcal{O}'$ with $\mathsf{Sig}(\mathcal{O}) \cap \mathsf{Sig}(\mathcal{O}') \subseteq \Sigma$, we have that $\mathcal{O} \cup \mathcal{O}'$ is a conservative extension of $\mathcal{O}'$; alternatively, this notion can be formulated in terms of inseparability: an ontology $\mathcal{O}$ is safe for a signature Σ if and only if $\mathcal{T} \equiv_{\Sigma}^{m} \emptyset$.

Observe that if $\mathcal{M}$ is a depleting Σ-module of $\mathcal{O}$, then for every ontology $\mathcal{O}'$ and every axiom α with $\mathsf{Sig}(\mathcal{O}' \cup \{\alpha\}) \cap \mathsf{Sig}(\mathcal{O}) \subseteq \Sigma$, it holds $\mathcal{O}' \cup \mathcal{O} \models \alpha$ if and only if $\mathcal{O}' \cup \mathcal{M} \models \alpha$.

In the module importing scenario, the signature Σ acts as the *interface* signature between $\mathcal{O}'$ and $\mathcal{O}$ in the sense that it contains symbols that $\mathcal{O}'$ and α may share with $\mathcal{O}$.

A "plain" Σ-module $\mathcal{M}$ of $\mathcal{O}$ covers all knowledge that $\mathcal{O}$ has about $\Sigma \cup \mathsf{Sig}(\mathcal{M})$. The notion of self-contained module is stronger than the plain Σ-module notion in that it requires the module to preserve entailments that can be formulated in the interface signature *plus* the signature of the module. That is, it covers all the knowledge that $\mathcal{O}$ has about $\Sigma \cup \mathsf{Sig}(\mathcal{M})$. On the other hand, for a *depleting* Σ-module $\mathcal{M}$ of $\mathcal{O}$, the difference $\mathcal{O} \setminus \mathcal{M}$ has no knowledge about $\Sigma \cup \mathsf{Sig}(\mathcal{M})$. This means that the difference of $\mathcal{O}$ and its module $\mathcal{M}$ does not entail any axioms in terms of $\Sigma \cup \mathsf{Sig}(\mathcal{M})$ other than tautologies. The relationship between the different kinds of modules induces by separability relations is as follows.

- If $\mathcal{M}$ is a self-contained Σ-module of $\mathcal{O}$, then it is a (plain) Σ-module.
- If $\mathcal{M}$ is a depleting Σ-module of $\mathcal{T}$, then it is a self-contained Σ-module.

$\mathcal{O}'$ is a *minimal* (self-contained, depleting) Σ-module in $\mathcal{O}$ if there is no $\mathcal{O}_1 \subsetneq \mathcal{O}'$ that is also a (self-contained, depleting) Σ-module in $\mathcal{O}$. With that notion in place, the following module extraction tasks can be defined:

T1. compute *all* minimal Σ-modules in $\mathcal{O}$.

T2. compute some minimal Σ-module

This follows from undecidability results about conservative extension and Σ-modules

(undec.) There is no algorithm for performing any of the task **T**1 and **T**2 for $\mathcal{ALC}$-ontologies.

Note however, that it is always possible to extract a Σ-module in $\mathcal{O}$. In particular, $\mathcal{O}$ is always a Σ module in $\mathcal{O}$. Finally, while there can be exponentially many minimal Σ-modules, minimal depleting modules are uniquely determined–under mild conditions involving inseparability relations.

Locality

The logic-based approach for defining modules in terms of model conservative extensions seems theoretically sound and appropriate to guarantee that reusing only terms from Σ one will not be able to distinguish between importing a ΣmCE (or dCE) module $\mathcal{M}$ of $\mathcal{O}$ and importing the whole $\mathcal{O}$. However, deciding CEs is computationally expensive in general: deciding whether two ontologies entail the same concept inclusions over a given signature is usually harder than standard reasoning. Due to the computationally difficulty to decide both kinds of CEs, approximations have been introduced based on the notion of *locality* of a single axioms. Intuitively, given Σ, a local axiom can always be satisfied independently of the interpretation of the Σ-terms, but in a restricted way: by interpreting all non-Σ terms as either the empty set ($\emptyset$-locality) or as the full domain (Δ-locality). In other words, *an axiom*

```
Input: TBox $\mathcal{T}$, signature $\Sigma$, $x \in \{\emptyset, \Delta, \bot, \top\}$
Output: $x$-module $\mathcal{M}$ of $\mathcal{O}$ w.r.t. $\Sigma$

$M \leftarrow \emptyset$; $\mathcal{T}' \leftarrow \mathcal{T}$
repeat
  changed $\leftarrow$ false
  for all $\alpha \in \mathcal{T}'$ do
    if $\alpha$ not $x$-local w.r.t. $\Sigma \cup \widetilde{\mathcal{M}}$ then
      $\mathcal{M} \leftarrow \mathcal{M} \cup \{\alpha\}$
      $\mathcal{T}' \leftarrow \mathcal{T}' \setminus \{\alpha\}$
      changed $\leftarrow$ true
    end if
  end for
until changed = false
return $\mathcal{M}$
```

Figure 1: Locality-based Module Extraction

α is local w.r.t. Σ if every trivial expansion of any Σ-interpretation to $\Sigma \cup \mathsf{Sig}(\alpha)$ is a model of α.

It has been shown that for a subset $\mathcal{M} \subseteq \mathcal{O}$ all axioms in $\mathcal{O} \setminus \mathcal{M}$ being $\emptyset$-local (or all axioms are Δ-local) w.r.t. $\Sigma \cup \mathsf{sig}(\mathcal{M})$ it is sufficient for $\mathcal{O}$ to be a Σ-mCE of $\mathcal{M}$. However, the converse does not hold. The previous result can be also formulated in a more general way as follows. Let $\mathcal{O}_1, \mathcal{O}_2$ be two ontologies and Σ a signature such that $\mathcal{O}_2$ is local w.r.t. $\Sigma \cup \mathsf{Sig}(\mathcal{O}_1)$, then $\mathcal{O}_1 \cup \mathcal{O}_2$ is a Σ-model conservative extension of $\mathcal{O}_1$.

One can use the standard capabilities of available DL-reasoners for testing ($\emptyset$ (Δ)-locality. Note, however, that reasoning on expressive DLs (underlying OWL-DL) is not tractable. In order to achieve *tractable* module extraction, syntactic approximations of locality have been introduced. These approximations come in two variants: $\bot$*-locality* and $\top$*-locality.* It has been shown that $\bot$-locality ($\top$-locality) of an axiom α w.r.t. Σ implies $\emptyset$-locality (Δ-locality) of α w.r.t. Σ. Therefore, all axioms being $\bot$-local (or all axioms being $\top$-local) w.r.t. $\Sigma \cup \mathsf{sig}(\mathcal{M})$ is sufficient for $\mathcal{O}$ to be a Σ-mCE of $\mathcal{M}$, but the converse does not hold.

Locality-based Module. Locality notions can be used to define modules as follows. Let $x \in \{\emptyset, \Delta, \top, \bot\}$ Let $\mathcal{M} \subseteq \mathcal{O}$ be ontologies and Σ a signature. $\mathcal{M}$ is an *x-locality-based* Σ-*module in* $\mathcal{O}$ if $\mathcal{O} \setminus \mathcal{M}$ is x-local w.r.t. $\Sigma \cup \mathsf{Sig}(\mathcal{M})$.

Modules of $\mathcal{O}$ for each of the four locality notions are obtained by starting with an empty set of axioms and subsequently adding axioms from $\mathcal{O}$ that are non-local. In order for this procedure to be correct, the signature against with locality is checked has to be extended with the terms in the axiom that is added in each step. x-locality based as define above, are always mCE-based (and therefore dCE-based).

In general, locality based modules are not minimal but provide the preservation of certain entailments. Modules based on syntactic locality can be made small by nesting $\top$-extraction into $\bot$-extraction and vice vera, and the result is still an mCE-based module.

A locality-based module $\mathcal{M}$ is a self-contained and depleting Σ-module. $\top$-locality and $\bot$-locality are sufficient for safety: If an ontology $\mathcal{O}$ is $\bot$-local or $\top$-local w.r.t. Σ, then $\mathcal{O}$ is safe for Σ.

Other Module Extraction Techniques

For rather unexpressive ontology languages ad-hoc (minimal) module extraction procedures have been developed. For the DL $\mathcal{EL}^+$ connected reachability-based modules correspond to modules based on syntactic locality.

Connected reachability and modules. Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology, $\Sigma \subseteq \mathsf{Sig}(\mathcal{O})$ a signature, and $x, y \in \mathsf{Sig}(\mathcal{O})$ concept or role names. x is said to be connectedly reachable from Σ w.r.t. $\mathcal{O}$ (for short, Σ-reachable) iff $x \in \Sigma$ or there is an axiom (either GCI or RI) $\alpha_L \sqsubseteq \alpha_R \in \mathcal{O}$ s.t. $x \in \mathsf{Sig}(\alpha_R)$ and, for all $y \in \mathsf{Sig}(\alpha_L)$, y is reachable from Σ. An axiom $\beta_L \sqsubseteq \beta_R$ is said to be connected reachable from Σ w.r.t. $\mathcal{O}$ (for short, Σ-reachable) if, for all $x \in \mathsf{Sig}(\beta_L)$, x is Σ-reachable. The *reachability- based module for* Σ *in* $\mathcal{O}$ is the set of all Σ-reachable axioms.

Intuitively, x is connectedly reachable from $\{y\}$ w.r.t. $\mathcal{O}$ means that y syntactically refers to x, either directly or indirectly via axioms in $\mathcal{O}$. If x, y are concept names, then the reachability suggests a potential subsumption relationship $y \sqsubseteq_{\mathcal{O}} x$.

It has been shown that for an $\mathcal{EL}^+$ ontology and signature Σ, the subset $\mathcal{O}_\Sigma \subseteq$ of all Σ-reachable axioms can be computed in linear time and that it is the minimal $\bot$ locality-based module w.r.t. Σ of $\mathcal{O}$.

Minimal Module Extraction (MEX and AMEX algorithms). Based on the notion of *depleting* Σ*-module*. Recall that a depleting Σ-module can be defined as a subset of an ontology that is self-contained in the sense that it is Σ-inseparable from the ontology, where Σ is the signature of the module.

One of the main features of Σ-inseparability is that it is language independent. Indeed, it implies inseparability w.r.t. any standard DL and even SO-logic. However, deciding Σ-inseparability of ontologies in fairly restricted scenarios is undecidable:

- Deciding Σ-inseparability is undecidable for $\mathcal{ALC}$-ontologies even if one ontology is acyclic an the other is empty;
- For the case of $\mathcal{EL}$ undecidability is attained when one ontology is empty and the other is a general ontology.

There are some natural conditions that lead to decidability though: e.g., to restrict the signature to a concept signature (i.e., only concept names/classes, but no role names/attributes):

1. Σ-inseparability of from the empty TBox, when Σ is a concept signature is in Π_2^p for $\mathcal{ALC}$ and PTime for $\mathcal{EL}$;
2. Σ-inseparability of acyclic $\mathcal{EL}$-TBoxes form the empty TBox is also PTime

Note that if $\mathcal{M}$ is a depleting Σ-module, then $\mathcal{O} \setminus \mathcal{M}$ is safe for $\mathsf{Sig}(\mathcal{M})$ and so the module can be maintained separately outside of $\mathcal{O}$ without the risk of unintended interaction with the rest of $\mathcal{O}$. Also note that checking depleting Σ modules is exactly the same problem as deciding Σ-inseparability form the empty ontology. Every depleting module $\mathcal{M}$ of $\mathcal{O}$ is inseparable from $\mathcal{O}$ for its signature. Thus, an ontology $\mathcal{O}$ and its depleting Σ-module $\mathcal{M}$ can be equivalently replaced by each other in applications concerning only Σ.

Based on the observations above, algorithms for module extraction from acyclic $\mathcal{ELI}$-TBoxes have been developed.

From deciding inseparability to module extraction: For acyclic $\mathcal{ELI}$ TBoxes, the *unique minimal* depleting Σ-module can be computed as follows.

- Given an acyclic $\mathcal{EL}$ ontology $\mathcal{O}$ and signature Σ, the decision procedure extracts from $\mathcal{O}$ the smallest $\mathcal{M} \subseteq \mathcal{O}$ such that $\mathcal{O} \setminus \mathcal{M} \equiv^{m}_{\Sigma \cup \mathsf{Sig}(\mathcal{M})} \emptyset$

  then $\mathcal{O} \setminus \mathcal{M}$ is safe for $\Sigma \cup \mathsf{Sig}(\mathcal{M})$ w.r.t. $\mathcal{EL}$
- Equivalently, $\mathcal{M} \equiv^{m}_{\Sigma \cup \mathsf{Sig}(\mathcal{M})} \mathcal{O}$.

then $\mathcal{M}$ is a depleting Σ-module in $\mathcal{O}$ w.r.t. $\mathcal{EL}$.

The previous approach for module extraction has been extended for acyclic $\mathcal{ALCQI}$-TBoxes. Although the problem for checking depleting modules for $\mathcal{ALCI}$ is undecidable, it has been proven that decidability can be regained when the remaining ontology $\mathcal{O} \setminus \mathcal{M}$ does not contain a direct Σ-dependency. An acyclic TBox $\mathcal{O}$ has a *direct* Σ*-dependency*, for some signature Σ, if there exits $\{A, X\} \subseteq \Sigma$ with $A \prec^{+}_{\mathcal{O}} X$.

Based on the previous result, a variation of the algorithm for $\mathcal{ELI}$ can be obtained that extracts a depleting Σ-module $\mathcal{M} \subseteq \mathcal{O}$ from an acyclic $\mathcal{ALCI}$ ontology that is not minimal in general, but it is minimal with the property that $\mathcal{O} \setminus \mathcal{M}$ does not contain a direct $\Sigma \cup \mathsf{Sig}(\mathcal{M})$-dependency.

Modules for ontology decomposition These are a special case of Σ-modules. The definition of this kind of modules is intended to capture the relevant information about a single concept.

A subset $\mathcal{M}_A$ of $\mathcal{O}$ is a *module for a concept name* $A \in \mathsf{Sig}(\mathcal{O})$ if

1. $\mathcal{M}_A$ is a Σ-module in $\mathcal{O}$, with $\Sigma = \{A\}$;
2. for every concept $B \in \mathsf{Sig}(\mathcal{O})$,
   - $\mathcal{M}_A \models A \sqsubseteq B$ iff $\mathcal{O} \models A \sqsubseteq B$.
   - $\mathcal{M}_A \models B \sqsubseteq A$ iff $\mathcal{O} \models B \sqsubseteq A$.
3. There are no concept names $D, E \in \mathsf{Sig}(\mathcal{O})$ such that $D \in \mathsf{Sig}(\mathcal{M}_A)$, $E \notin \mathsf{Sig}(\mathcal{M}_A)$ and either $\mathcal{O} \models D \sqsubseteq E$, or $\mathcal{O} \models E \sqsubseteq D$.

Observe that by definition, modules for a concept name are self-contained. Intuitively, extracting a module $\mathcal{M}_A$ of (an OWL ontology) $\mathcal{O}$ should guarantee that A, as well as all its sub-concepts, super-concepts and instances are included in $\mathcal{M}_A$.

Implementations

Table below summarizes some of the module extraction algorithms implementations for OWL ontologies discussed.

Graph theory-based approaches

Beside the logic-based module extraction approaches, numerous techniques for extracting fragments of ontologies for knowledge reused purposed have been developed that rely on syntactically traversing the axioms in the ontology and employ various heuristics for determining which axioms are relevant and which not.

An example of such a procedure is the algorithm implemented in the PROMPT-FACTOR tool. Given a signature Σ and an ontology $\mathcal{O}$, the algorithm retrieves a fragment $\mathcal{M}$ of $\mathcal{O}$ A common, and arguably, simple approach to modularize an ontology is to traverse the ontology hierarchy and apply heuristics to identify a *sub-graph* corresponding to a module. However, such approaches do no take into consideration the underlying semantics of the ontology, and hence *do not* generate modules that are *complete.* in the sense that those modules might not contains all the information *relevant* to the elements of Σ. Nonetheless, graph-based algorithms for module extraction are tractable, intuitive to an user, and somewhat widely used.

Doran, Tamma and Iannone focus on extracting an ontology module that describes a single, user supplied, concept for the purpose of ontology reuse. Their approach is agnostic with respect to the ontology language, as long as the ontology can be transformed into their *Abstract Graph Model for ontology Module Extraction.* The traversal carried out for extraction is done conditionally, with the conditions changing to suit the ontology language. For example, if the "seed" concept is involved in a disjoint relation then this will not be traversed.

d'Aquin, Sabou and Motta present an extraction process which forms part of a knowledge selection process.

Seidenberg and Rector developed a technique specifically for the Galen ontology, but it is possible to take the generic core and apply it to any ontology. It takes one or more classes of the ontology as an input, and anything that participates (even indirectly) to the description of an included class is included as well. To reduce the size of the obtained module without losing relevant information, Galen properties are filtered on the base of the property hierarchy.

Noy and Musen approach is based on the notion of *traversal view extraction.* It is made available via the PROMPT plugin for Protégé. Starting from *one class* of the considered ontology, this approach traverses the relations of this class recursively to include related entities. It can be distinguished rom other approaches in that it is intended as an interactive tool. This allows the used to incrementally build ontology modules by extending the currently extracted one

Properties	**Module Notions**				
	plain	self-contained	depleting	$\Delta, \emptyset, \top, \bot, (\top\bot)^*$ and $(\bot\top)^*$ locality-based	$\top\bot$, and $\bot\top$ locality-based
mCE (dCE)	✓	✓	✓	✓	
inseparability	$\mathcal{O} \equiv^m_\Sigma \mathcal{M}$	$\mathcal{O} \equiv^m_{\Sigma\cup\mathsf{Sig}(\mathcal{M})} \mathcal{M}$	$\mathcal{O} \setminus \mathcal{M} \equiv^m_{\Sigma\cup\mathsf{Sig}(\mathcal{M})} \emptyset$		
self-contained	✗	✓	✓	✓	✓
depleting	✗	✗	✓	✓	✓
safety					
robustness voc. restr.	✓	✓	✓	✓	✗
robustness voc. ext.	✓	✓	✓	✓	✗
robustness replacement	✓	✓	✓	✓	✓
robustness joins	✓	✓	✓	✓	✓

Table 1: Properties of Σ-modules

Module Extraction	**Language**	**Implementations:**
Locality-based	OWL–DL	OWL API (`http://owlapi.sourceforge.net`). For a complete overview of the available methods, see the Javadoc (`http://owlapi.sourceforge.net/javadoc/index.html`), in particular the class `SyntacticLocalityModuleExtractor`. The relevant classes and interfaces are in the following packages: • com.clarkparsia.owlapi.modularity.locality • org.semanticweb.owlapi.modularity • uk.ac.manchester.cs.owlapi.modularity Web Module extractor (`http://mowl-power.cs.man.ac.uk:8080/modularity/`)
MEX	OWL-EL acyclic ontologies	Implemented in OCaml as the MEX command line software (`http://cgi.csc.liv.ac.uk/~konev/software/mex.bin`)
AMEX	acyclic $\mathcal{ALCQI}$ Ontologies roughly acyclic OWL-lite ontologies	AMEX (`http://cgi.csc.liv.ac.uk/~wgatens/software/amex.html`): command line module extractor written in Java, built with the OWL-API and distributed as a self-contained java jar archive. It requires the 3rd party QBF solver called sKizzo (`http://www.skizzo.info/`) for certain parts of the extraction algorithm (see the AMEX web page for more details).

Table 2: Module Extraction Implementations