Which Kind of Module Should I Extract?

Uli Sattler\textsuperscript{1} \hspace{1cm} Thomas Schneider\textsuperscript{1} \hspace{1cm} Michael Zakharyaschev\textsuperscript{2}

\textsuperscript{1}School of Computer Science, University of Manchester
\textsuperscript{2}Birkbeck College, London

DL, 28 July 2009
And now . . .

1. Motivation
2. Inseparability relations
3. Robustness properties
4. Conclusions
Why module extraction?

Reuse external ontologies: borrow knowledge about certain terms
Why module extraction?

Reuse external ontologies: borrow knowledge about certain terms

- Provides access to well-established knowledge
- Doesn’t require expertise in external disciplines
Why module extraction?

Reuse external ontologies: borrow knowledge about certain terms
Why module extraction?

Reuse external ontologies: borrow knowledge about certain terms

Animals

knowledge about “Bird” and “feedsOn”

Farm
Why module extraction?

Reuse external ontologies: borrow knowledge about certain terms

Animals

knowledge about “Bird” and “feedsOn”

Farm

How much of Animals do we need?
Why module extraction?

Reuse external ontologies: borrow knowledge about certain terms

{Bird, feedsOn}  

knowledge about “Bird” and “feedsOn”

Animals

Farm

How much of Animals do we need?
Why module extraction?

Reuse external ontologies: borrow knowledge about certain terms

\{\text{Bird, feedsOn}\} \rightarrow \text{knowledge about “Bird” and “feedsOn”}

**Coverage** Import *everything* relevant for the chosen terms.

**Economy** Import *only* what’s relevant for them. Compute that module quickly.
Why module extraction?

Reuse external ontologies: borrow knowledge about certain terms

{Bird, feedsOn}  

knowledge about “Bird” and “feedsOn”

Animals  

Farm

Coverage  Import *everything* relevant for the chosen terms.

Economy  Import *only* what’s relevant for them.
Compute that module quickly.
Modules that provide coverage

Input
Ontology $\mathcal{O} —$ a set of axioms
Signature $\Sigma$ (set of concept and role names from $\mathcal{O}$)

Output
a $\Sigma$-module $\mathcal{M}$ of $\mathcal{O}$:

- $\mathcal{M} \subseteq \mathcal{O}$
- $\mathcal{M}$ and $\mathcal{O}$ have the same $\Sigma$-entailments:
  For all axioms $\alpha$ using only terms from $\Sigma$,
  $\mathcal{O} \models \alpha$ iff $\mathcal{M} \models \alpha$
Modules that provide coverage

Input
Ontology $\mathcal{O}$ — a set of axioms
Signature $\Sigma$ (set of concept and role names from $\mathcal{O}$)

Output
a $\Sigma$-module $\mathcal{M}$ of $\mathcal{O}$:

- $\mathcal{M} \subseteq \mathcal{O}$
- $\mathcal{M}$ and $\mathcal{O}$ have the same $\Sigma$-entailments:
  For all axioms $\alpha$ using only terms from $\Sigma$,
  $\mathcal{O} \models \alpha$ iff $\mathcal{M} \models \alpha$

Coverage ✔
Motivation

Inseparability relations

Robustness properties

Conclusions

Modules that provide coverage

Input  Ontology \( \mathcal{O} \) — a set of axioms
Signature \( \Sigma \) (set of concept and role names from \( \mathcal{O} \))

Output  a \( \Sigma \)-module \( \mathcal{M} \) of \( \mathcal{O} \):

- \( \mathcal{M} \subseteq \mathcal{O} \)
- \( \mathcal{M} \) and \( \mathcal{O} \) have the same \( \Sigma \)-entailments:
  For all axioms \( \alpha \) using only terms from \( \Sigma \),
  \( \mathcal{O} \models \alpha \) iff \( \mathcal{M} \models \alpha \)

Coverage  ✔

Economy  Minimality  \( \iff \) efficient computability
### Modules that provide coverage

**Input**
- Ontology $\mathcal{O}$ — a set of axioms
- Signature $\Sigma$ (set of concept and role names from $\mathcal{O}$)

**Output**
- A $\Sigma$-module $\mathcal{M}$ of $\mathcal{O}$:
  - $\mathcal{M} \subseteq \mathcal{O}$
  - $\mathcal{M}$ and $\mathcal{O}$ have the same $\Sigma$-entailments:
    - For all axioms $\alpha$ using only terms from $\Sigma$, $\mathcal{O} \models \alpha$ iff $\mathcal{M} \models \alpha$

#### Coverage
- ✔️

#### Economy
- Minimality: conservativity-based modules
- Efficient computability: locality-based modules
Relevant module types

- dCE
- deductive conservativity
- intractable...undecidable
Relevant module types

\[ x \text{-module}(\mathcal{O}, \Sigma) \subseteq y \text{-module}(\mathcal{O}, \Sigma) \]

- intractable...undecidable
Relevant module types

- dCE: deductive conservativity
- mCE: model cons.
- $\Delta$: semantic locality

$x$-module($\mathcal{O}, \Sigma$) $\subseteq$ $y$-module($\mathcal{O}, \Sigma$)

- intractable...undecidable
- as difficult as reasoning
Relevant module types

- dCE: deductive conservativity
- mCE: model cons.
- ∅: semantic locality
- ⊤: syntactic locality

\[ x - \delta - y \quad x\text{-module}(O, \Sigma) \subseteq y\text{-module}(O, \Sigma) \]

- Intractable...Undecidable
- As difficult as reasoning
- Tractable
Relevant module types

- dCE: deductive conservativity
- mCE: model conservativity
- ∅: semantic locality
- ⊤: syntactic locality

\( x \rightarrow y \) \( x\text{-module}(\mathcal{O}, \Sigma) \subseteq y\text{-module}(\mathcal{O}, \Sigma) \)

- intractable . . . undecidable
- as difficult as reasoning
- tractable
Relevant module types

- dCE: deductive conservativity
- mCE: model cons.
- ∅: semantic locality
- ⊤*: syntactic locality

\[ x \sim y \Rightarrow x\text{-module}(\mathcal{O}, \Sigma) \subseteq y\text{-module}(\mathcal{O}, \Sigma) \]

- Pink: intractable...undecidable
- Yellow: as difficult as reasoning
- Green: tractable
Goals

- General framework for comparing module notions that provide coverage
- Identify relevant properties
- Application to conservativity-based and locality-based modules
And now . . .

1. Motivation

2. Inseparability relations

3. Robustness properties

4. Conclusions
Intuitions

- $\mathcal{O}_1$ and $\mathcal{O}_2$ are inseparable w.r.t. $\Sigma$:
  The knowledge about $\Sigma$ in $\mathcal{O}_1$ and $\mathcal{O}_2$ can’t be distinguished

- Different degrees of distinguishability
Intuitions

- $\mathcal{O}_1$ and $\mathcal{O}_2$ are inseparable w.r.t. $\Sigma$:
  The knowledge about $\Sigma$ in $\mathcal{O}_1$ and $\mathcal{O}_2$ can’t be distinguished

- Different degrees of distinguishability

- Notation: $\mathcal{O}_1 \equiv^S_{\Sigma} \mathcal{O}_2$

- $\equiv^S_{\Sigma}$ is an equivalence relation
Intuitions

- \( \mathcal{O}_1 \) and \( \mathcal{O}_2 \) are inseparable w.r.t. \( \Sigma \):
  The knowledge about \( \Sigma \) in \( \mathcal{O}_1 \) and \( \mathcal{O}_2 \) can’t be distinguished
- Different degrees of distinguishability
- Notation: \( \mathcal{O}_1 \equiv^S \Sigma \mathcal{O}_2 \)
- \( \equiv^S \Sigma \) is an equivalence relation
- Inseparability relation: \( S = \{ \equiv^S \Sigma \mid \Sigma \text{ is a signature} \} \)
Different inseparability relations

- $\mathcal{O}_1 \equiv_{\Sigma} \mathcal{O}_2$ if:
  $\mathcal{O}_1$ and $\mathcal{O}_2$ entail the same $\Sigma$-concept subsumptions
Different inseparability relations

- $\mathcal{O}_1 \equiv^{\text{dCE}}_{\Sigma} \mathcal{O}_2$ if:
  $\mathcal{O}_1$ and $\mathcal{O}_2$ entail the same $\Sigma$-concept subsumptions

- $\mathcal{O}_1 \equiv^{\text{mCE}}_{\Sigma} \mathcal{O}_2$ if:
  $\mathcal{O}_1$ and $\mathcal{O}_2$ have the same models w.r.t. $\Sigma$
Different inseparability relations

- $O_1 \equiv^{dCE}_\Sigma O_2$ if:
  $O_1$ and $O_2$ entail the same $\Sigma$-concept subsumptions

- $O_1 \equiv^{mCE}_\Sigma O_2$ if:
  $O_1$ and $O_2$ have the same models w.r.t. $\Sigma$

- $O_1 \equiv^{\perp}_\Sigma O_2$ if:
  $O_1$ and $O_2$ have the same $\perp$-module w.r.t. $\Sigma$
Different inseparability relations

- $O_1 \equiv_{dCE}^\Sigma O_2$ if:
  - $O_1$ and $O_2$ entail the same $\Sigma$-concept subsumptions

- $O_1 \equiv_{mCE}^\Sigma O_2$ if:
  - $O_1$ and $O_2$ have the same models w.r.t. $\Sigma$

- $O_1 \equiv_{\bot}^\Sigma O_2$ if:
  - $O_1$ and $O_2$ have the same $\bot$-module w.r.t. $\Sigma$

Analogous definition for

- $\equiv_0^\Sigma$
- $\equiv_\Delta^\Sigma$
- $\equiv_T^\Sigma$
- $\equiv_{\bot T}^\Sigma$
- $\equiv_{\bot T}^\Sigma$
- $\equiv_{\bot T}^\Sigma$
- $\equiv_{\bot T}^\Sigma$
- $\equiv_{\bot T}^\Sigma$
- $\equiv_{\bot T}^\Sigma$
Inseparability relations induce modules

Let $S$ be an inseparability relation, $\Sigma$ a signature and $M \subseteq O$.

$M$ is called if

\[
\begin{array}{c|c|c}
\text{an } S_{\Sigma}-\text{module of } O & \text{if} & \text{see} \\
M \equiv_{S_{\Sigma}} O & 1 \\
\end{array}
\]

Example: $S = dCE$, $\Sigma = \{\text{Bird, feedsOn}\}$, $M$ contains Grass.

\[O \models \text{Bird} \sqsubseteq \exists \text{feedsOn}.T \quad \text{iff} \quad M \models \text{Bird} \sqsubseteq \exists \text{feedsOn}.T\]
Inseparability relations induce modules

Let $S$ be an inseparability relation, $\Sigma$ a signature and $\mathcal{M} \subseteq \mathcal{O}$.

<table>
<thead>
<tr>
<th>$\mathcal{M}$ is called</th>
<th>if</th>
<th>see</th>
</tr>
</thead>
<tbody>
<tr>
<td>an $S_\Sigma$-module of $\mathcal{O}$</td>
<td>$\mathcal{M} \equiv S_\Sigma \mathcal{O}$</td>
<td>1</td>
</tr>
<tr>
<td>a self-contained $S_\Sigma$-module of $\mathcal{O}$</td>
<td>$\mathcal{M} \equiv S_\Sigma \cup \text{sig}(\mathcal{M}) \mathcal{O}$</td>
<td>2</td>
</tr>
</tbody>
</table>

**Example:** $S = \text{dCE}$, $\Sigma = \{\text{Bird}, \text{feedsOn}\}$, $\mathcal{M}$ contains Grass.

1 $\mathcal{O} \models \text{Bird} \sqsubseteq \exists \text{feedsOn}.T$ \iff $\mathcal{M} \models \text{Bird} \sqsubseteq \exists \text{feedsOn}.T$

2 $\mathcal{O} \models \text{Bird} \sqsubseteq \exists \text{feedsOn}.\text{Grass}$ \iff $\mathcal{M} \models \text{Bird} \sqsubseteq \exists \text{feedsOn}.\text{Grass}$
Inseparability relations induce modules

Let $S$ be an inseparability relation, $\Sigma$ a signature and $\mathcal{M} \subseteq \mathcal{O}$.

<table>
<thead>
<tr>
<th>$\mathcal{M}$ is called</th>
<th>if $\mathcal{M}$ is called</th>
<th>see</th>
</tr>
</thead>
<tbody>
<tr>
<td>an $S_\Sigma$-module of $\mathcal{O}$</td>
<td>$\mathcal{M} \equiv S_\Sigma \mathcal{O}$</td>
<td>1</td>
</tr>
<tr>
<td>a self-contained $S_\Sigma$-module of $\mathcal{O}$</td>
<td>$\mathcal{M} \equiv S_{\Sigma \cup \text{sig}(\mathcal{M})} \mathcal{O}$</td>
<td>2</td>
</tr>
<tr>
<td>a depleting $S_\Sigma$-module of $\mathcal{O}$</td>
<td>$\emptyset \equiv S_{\Sigma \cup \text{sig}(\mathcal{M})} \mathcal{O} \setminus \mathcal{M}$</td>
<td>3</td>
</tr>
</tbody>
</table>

Example: $S = \text{dCE}, \Sigma = \{\text{Bird, feedsOn}\}$, $\mathcal{M}$ contains Grass.

1. $\mathcal{O} \models \text{Bird} \sqsubseteq \exists \text{feedsOn.T}$ iff $\mathcal{M} \models \text{Bird} \sqsubseteq \exists \text{feedsOn.T}$
2. $\mathcal{O} \models \text{Bird} \sqsubseteq \exists \text{feedsOn.Grass}$ iff $\mathcal{M} \models \text{Bird} \sqsubseteq \exists \text{feedsOn.Grass}$
3. $\mathcal{O} \setminus \mathcal{M}$ entails only tautologies w.r.t. $\{\text{Bird, feedsOn, Grass}\}$. 
And now . . .

1 Motivation
2 Inseparability relations
3 Robustness properties
4 Conclusions
Robustness properties (1)

$S$ is robust under vocabulary restrictions:

If $O_1 \equiv^S \Sigma O_2$ and $\Sigma' \subseteq \Sigma$, then $O_1 \equiv^S_{\Sigma'} O_2$. 

Consequences:

If $M$ is a $\Sigma$-module of $O$ and $\Sigma' \subseteq \Sigma$, then $M$ is a $\Sigma'$-module of $O$. 

On restricting the signature, no new import is necessary.
Robustness properties (1)

**S** is robust under vocabulary restrictions:

If $O_1 \equiv^S_{\Sigma} O_2$ and $\Sigma' \subseteq \Sigma$, then $O_1 \equiv^S_{\Sigma'} O_2$.

Consequences:

If $M$ is a $\Sigma$-module of $O$ and $\Sigma' \subseteq \Sigma$, then $M$ is a $\Sigma'$-module of $O$.

$\sim$ On restricting the signature, no new import is necessary.
Robustness properties (2)

- Vocabulary extensions

  If $\mathcal{M}$ is a $\Sigma$-module of $\mathcal{O}$ and $(\Sigma' \setminus \Sigma) \cap \text{sig}(\mathcal{O}) = \emptyset$, then $\mathcal{M}$ is a $\Sigma'$-module of $\mathcal{O}$.

  → On extending the signature with terms outside $\mathcal{O}$, no new import is necessary.
Robustness properties (2)

- **Vocabulary extensions**
  
  If $\mathcal{M}$ is a $\Sigma$-module of $\mathcal{O}$ and $(\Sigma' \setminus \Sigma) \cap \text{sig}(\mathcal{O}) = \emptyset$, then $\mathcal{M}$ is a $\Sigma'$-module of $\mathcal{O}$.

  $\sim$ On extending the signature with terms outside $\mathcal{O}$, no new import is necessary.

- **Replacement**
  
  If $\mathcal{M}$ is a $\Sigma$-module of $\mathcal{O}$ and $(\text{sig}(\mathcal{O'}) \setminus \Sigma) \cap \text{sig}(\mathcal{O}) = \emptyset$, then $\mathcal{M} \cup \mathcal{O'}$ is a $\Sigma$-module of $\mathcal{O} \cup \mathcal{O'}$.

  $\sim$ The module relation is compatible with imports.
Robustness properties (2)

- **Vocabulary extensions**

  If \( M \) is a \( \Sigma \)-module of \( O \) and \((\Sigma' \setminus \Sigma) \cap \text{sig}(O) = \emptyset\), then \( M \) is a \( \Sigma' \)-module of \( O \).

  \( \leadsto \) On extending the signature with terms outside \( O \), no new import is necessary.

- **Replacement**

  If \( M \) is a \( \Sigma \)-module of \( O \) and \((\text{sig}(O') \setminus \Sigma) \cap \text{sig}(O) = \emptyset\), then \( M \cup O' \) is a \( \Sigma \)-module of \( O \cup O' \).

  \( \leadsto \) The module relation is compatible with imports.

- **Joins**

  If we have two indistinguishable ontologies, it suffices to import one of them.
### Overview of properties

#### Inseparability rel. (IR)

<table>
<thead>
<tr>
<th>Property</th>
<th>$\equiv_{\Sigma}$</th>
<th>$\equiv_{\Sigma}$</th>
<th>$\equiv_{\Sigma}$</th>
<th>$\equiv_{\Sigma}$</th>
<th>$\equiv_{\Sigma}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modules are induced ...</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>modules</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>self-contained modules</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>depleting modules</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>IR is robust under ...</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>vocab. restrictions</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>vocab. extensions</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>replacement</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>joins</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
</tbody>
</table>
And now . . .

1. Motivation
2. Inseparability relations
3. Robustness properties
4. Conclusions
mCE-based and (most) locality-based modules are very robust.

dCE-based modules are not robust.

Locality-based modules can be extracted efficiently.

\[ \sim \text{Intermediate step for extracting mCE-based modules} \]
Conclusions

- mCE-based and (most) locality-based modules are very robust.
- dCE-based modules are not robust.
- Locality-based modules can be extracted efficiently.
  $\leadsto$ Intermediate step for extracting mCE-based modules

Thank you.