Description Logics: a Nice Family of Logics — Modularity —

Uli Sattler[1] *Thomas Schneider*[2]

[1]School of Computer Science, University of Manchester, UK

[2]Department of Computer Science, University of Bremen, Germany

ESSLLI, 10 August 2012

Plan for today

1. What is modularity good for?
2. Modules for reuse
3. Summary and Outlook

And now . . .

1. What is modularity good for?
2. Modules for reuse
3. Summary and Outlook

What can I do with my ontology?

Ontology users and engineers want to use ontologies to

- represent and archive knowledge **(M)**
  in a structured way
- compute inferences from archived knowledge **(M)**
  e.g., classification, query answering
- explain inferences **(M)**
  justifications = pinpointing, abduction
- reuse (parts of) other ontologies to build their ontology **(M)**
  import
- expose the logical structure of the represented knowledge **(M)**
  comprehension

(M) = modularity helps

What can I do with my ontology?

Building and using an ontology often requires

- fast reasoning **(M)**
  expressivity $\leftrightarrow$ complexity; optimisations, incremental reasoning
- collaborative development **(M)**
- version control **(M)**
- *efficient* reuse **(M)**
- an understanding of the ontology's content and structure **(M)**
  comprehension

(M) = modularity helps

A priori vs. a posteriori modularisation

A priori (not covered today)

- At first, a modular structure is decided on.
- Then, the ontology is developed and used according to that structure.

A posteriori

- The ontology is regarded as a monolithic entity.
- At some point, a module is extracted
  or the ontology is decomposed into several modules.

And now . . .

Comparing two ontologies

Assume that . . .

- you want to buy a medical ontology from me
- I offer two medical ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$

Q: which one do you choose?

Possible A: the one that contains more knowledge.

Q: how do you measure the amount of knowledge in $\mathcal{O}_i$?

Possible A: ~~Number of axioms?~~

- Well, compare $\{A \sqsubseteq B,\ B \sqsubseteq A\}$ vs. $\{A \equiv B\}$
- or $\{A \sqsubseteq B,\ \ B \sqsubseteq A \sqcup \neg A,\ \ A \sqcap \neg A \sqsubseteq B\}$ vs. $\{A \equiv B\}$

Possible A: Number of entailments? Number of models?

Ontologies and their entailments

Think of axioms as **generating entailments** – e.g.:

$$\left.\begin{array}{l} A \sqsubseteq \exists r.B \\ \exists r.\top \sqsubseteq C \sqcap D \end{array}\right\} \models \quad A \sqsubseteq D$$

Q: how many entailments can a TBox have?

A: 0? 1? 2? ... n? ... 2^n? ... ∞?

Ontologies and their entailments

Think of axioms as **generating entailments** – e.g.:

$$\left.\begin{array}{l} A \sqsubseteq \exists r.B \\ \exists r.\top \sqsubseteq C \sqcap D \end{array}\right\} \models \quad A \sqsubseteq D$$

Q: how many entailments can a TBox have?

A: ∞

$$A \sqsubseteq D \quad A \sqsubseteq D \sqcup A \quad A \sqsubseteq D \sqcup (A \sqcap D), \quad \ldots$$

Ontologies and their models

Think of axioms as **restricting possible models**

Axioms "filter out" unwanted models – e.g.:

- $\mathtt{Hand} \sqsubseteq \exists\, \mathtt{hasPart.Finger}$

  $\rightsquigarrow$ models cannot have instances of Hand with no hasPart-edge to an instance of Finger

- $\mathtt{Hand} \sqsubseteq\; =5\, \mathtt{hasPart.Finger}$

  $\rightsquigarrow$ models cannot have instances of Hand with $\neq 5$ hasPart-edges to instances of Finger

Q: how many models can a TBox have?

A: 0? 1? 2? ... n? ... 2^n? ... ∞?

Ontologies and their models

Think of axioms as **restricting possible models**

Axioms "filter out" unwanted models – e.g.:

- $\mathtt{Hand} \sqsubseteq \exists\,\mathtt{hasPart.Finger}$

  $\leadsto$ models cannot have instances of Hand with no hasPart-edge to an instance of Finger

- $\mathtt{Hand} \sqsubseteq {=}\,5\,\mathtt{hasPart.Finger}$

  $\leadsto$ models cannot have instances of Hand with $\neq 5$ hasPart-edges to instances of Finger

Q: how many models can a TBox have?

A: 0 ∞

Next attempt at "more" entailments/models

We cannot compare *numbers* of entailments or models

But we can use set inclusion:

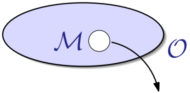
"$\mathcal{O}$ knows at most as much as $\mathcal{O}'$" if

- every entailment of $\mathcal{O}$ is one of $\mathcal{O}'$:
  $\{\eta \mid \mathcal{O} \models \eta\} \subseteq \{\eta \mid \mathcal{O}' \models \eta\}$ or
- every model of $\mathcal{O}'$ is one of $\mathcal{O}$:
  $\{\mathcal{I} \mid \mathcal{I} \models \mathcal{O}'\} \subseteq \{\mathcal{I} \mid \mathcal{I} \models \mathcal{O}\}$

Problem:

How do we test these conditions?

Knowledge w.r.t. a signature



Let's reformulate the initial dialogue.

Assume that ...

- you want to buy **a subset of** a medical ontology $\mathcal{O}$ from me that covers the subdomain of, say, diseases
- I offer two subsets $\mathcal{M}_1$ and $\mathcal{M}_2$

Q: which one do you choose?

Possible A: the one that "knows more" about diseases!

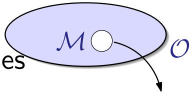
Q: which is the **best subset** I can offer?

Possible A: a **module for diseases**

- $\mathcal{M} \subseteq \mathcal{O}$ that knows as much as $\mathcal{O}$ about diseases:
  $\mathcal{M}$ **indistinguishable** from $\mathcal{O}$ w.r.t. all terms relevant for diseases
- $\mathcal{M}$ as small as possible

Inseparability w.r.t. a signature

Definition

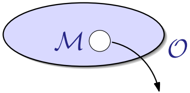


- **Signature** Σ = a set of concept/role names
- The signature of axiom (ontology) X
  = all concept/role names in X
- $\mathcal{O}_1$ and $\mathcal{O}_2$ are **Σ-inseparable w.r.t. a logic $\mathcal{L}$**,
  written $\mathcal{O}_1 \equiv_\Sigma^\mathcal{L} \mathcal{O}_2$, if: [Konev et al. 2009]
  for all $\eta \in \mathcal{L}$ with $\mathrm{sig}(\eta) \subseteq \Sigma$,
  $$\mathcal{O}_1 \models \eta \qquad \text{iff} \qquad \mathcal{O}_2 \models \eta$$
- $\mathcal{O}$ is a **Σ-conservative extension (Σ-dCE)** of $\mathcal{M}$ w.r.t. $\mathcal{L}$
  if $\mathcal{M} \subseteq \mathcal{O}$ and $\mathcal{M} \equiv_\Sigma^\mathcal{L} \mathcal{O}$ [Ghilardi et al. 2006]

  Alternative names:
  - $\mathcal{M}$ **covers** $\mathcal{O}$ for Σ w.r.t. $\mathcal{L}$
  - $\mathcal{M}$ **is a module of** $\mathcal{O}$ for Σ w.r.t. $\mathcal{L}$

Choosing the signature Σ

Definition (repeated from previous slide)



$\mathcal{O}$ is a Σ-module of $\mathcal{M}$ w.r.t. $\mathcal{L}$
if $\mathcal{M} \subseteq \mathcal{O}$ and $\mathcal{M} \equiv_{\Sigma}^{\mathcal{L}} \mathcal{O}$

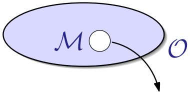
The signature Σ ...

- can be seen as a "topic"
- that the module is required to cover
- is difficult to formulate:

  Q: how many interesting entailments in $\Sigma = \{\texttt{Disease}\}$ can $\mathcal{O}$ possibly have?

Choosing the logic $\mathcal{L}$

Definition (repeated from previous slide)



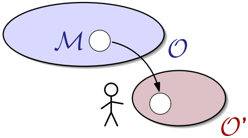
$\mathcal{O}$ is a Σ-module of $\mathcal{M}$ w.r.t. $\mathcal{L}$
if $\mathcal{M} \subseteq \mathcal{O}$ and $\mathcal{M} \equiv^{\mathcal{L}}_{\Sigma} \mathcal{O}$

Choice of $\mathcal{L}$ depends on your usage of the module:

- for ontology design: subsumptions betw. (complex?) concepts
- for ontology usage: your favourite query language

Modules for reuse

If we want to reuse module $\mathcal{M}$,
we need a stronger guarantee:



$$\mathcal{M} \cup \mathcal{O}' \equiv^{\mathcal{L}}_{\Sigma} \mathcal{O} \cup \mathcal{O}' \quad \text{for all } \mathcal{O}'$$

Q: is this reasonable to expect? **A:** no! Consider

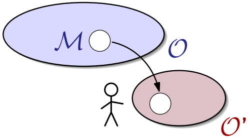
$$\mathcal{O} = \{A \sqsubseteq B,\ A \sqsubseteq \exists r.C\} \quad \Sigma = \{A, r, C\} \quad \mathcal{O}' = \{B \sqsubseteq C\}$$

Then $\mathcal{M} = \{A \sqsubseteq \exists r.C\} \equiv^{\mathcal{ALC}}_{\Sigma} \mathcal{O}$,
but $\mathcal{M} \cup \mathcal{O}' \not\equiv^{\mathcal{ALC}}_{\Sigma} \mathcal{O} \cup \mathcal{O}'$,
because $\mathcal{O} \cup \mathcal{O}' \models A \sqsubseteq C$

Modules for reuse

If we want to reuse module $\mathcal{M}$,
we need a stronger guarantee:

$$\mathcal{M} \cup \mathcal{O}' \equiv^{\mathcal{L}}_{\Sigma} \mathcal{O} \cup \mathcal{O}' \quad \text{for all } \mathcal{O}'$$



Q: is this reasonable to expect? **A:** no!

Solution:

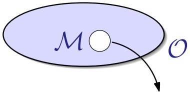
Lemma [Konev et al. 2009]

If $\mathcal{M} \equiv^{\mathcal{L}}_{\Sigma} \mathcal{O}$, then $\mathcal{M} \cup \mathcal{O}' \equiv^{\mathcal{L}}_{\Sigma} \mathcal{O} \cup \mathcal{O}'$, for

- every $\mathcal{O}'$ with $\mathsf{sig}(\mathcal{O}) \cap \mathsf{sig}(\mathcal{O}') \subseteq \Sigma$,
- expressive enough $\mathcal{L}$, e.g. $\mathcal{SROIQ}$ (OWL).

Consequence:
we can safely import $\mathcal{M}$ into any $\mathcal{O}'$ that reuses only terms from Σ

How is a minimal Σ-module extracted?



Simple module extraction algorithm:

- $\mathcal{M} \leftarrow \mathcal{O}$
- While $\mathcal{M} \setminus \{\alpha\} \equiv^{\mathcal{L}}_{\Sigma} \mathcal{O}$, for some $\alpha \in \mathcal{M}$,
  do $\mathcal{M} \leftarrow \mathcal{M} \setminus \{\alpha\}$
- Output $\mathcal{M}$

Observation:

Different orders of choosing α
can lead to different minimal modules

Example

Let $\Sigma = \{\mathsf{Knee}, \mathsf{HingeJoint}\}$. Suppose *Galen* contains:

$$
\begin{aligned}
\mathsf{Knee} &\equiv \mathsf{Joint} \sqcap \exists \mathsf{hasPart}.\mathsf{Patella} \sqcap && (1)\\
&\quad \exists \mathsf{hasFunct}.\mathsf{Hinge} && \\
\mathsf{Patella} &\sqsubseteq \mathsf{Bone} \sqcap \mathsf{Sesamoid} && (2)\\
\mathsf{Ginglymus} &\equiv \mathsf{Joint} \sqcap \exists \mathsf{hasFunct}.\mathsf{Hinge} && (3)\\
\mathsf{Joint} \sqcap \exists \mathsf{hasPart}.(\mathsf{Bone} \sqcap \mathsf{Sesamoid}) &\sqsubseteq \mathsf{Ginglymus} && (4)\\
\mathsf{Ginglymus} &\equiv \mathsf{HingeJoint} && (5)\\
\mathsf{Meniscus} &\equiv \mathsf{FibroCartilage} \sqcap \exists \mathsf{locatedIn}.\mathsf{Knee} && (6)
\end{aligned}
$$

$\subseteq$-Minimal module for Σ ?

Example

Let $\Sigma = \{\text{Knee}, \text{HingeJoint}\}$. Suppose *Galen* contains:

$$\begin{aligned}
\text{Knee} &\equiv \text{Joint} \sqcap \exists\text{hasPart.Patella} \sqcap && (1)\\
&\quad \exists\text{hasFunct.Hinge} && \\
\text{Patella} &\sqsubseteq \text{Bone} \sqcap \text{Sesamoid} && (2)\\
\text{Joint} \sqcap \exists\text{hasPart.}(\text{Bone}\sqcap\text{Sesamoid}) &\sqsubseteq \text{Ginglymus} && (4)\\
\text{Ginglymus} &\equiv \text{HingeJoint} && (5)
\end{aligned}$$

$\subseteq$-Minimal module for Σ? $\{(1), (2), (4), (5)\}$

Example

Let $\Sigma = \{\mathsf{Knee}, \mathsf{HingeJoint}\}$. Suppose *Galen* contains:

$$\mathsf{Knee} \equiv \mathsf{Joint} \sqcap \exists \mathsf{hasPart}.\mathsf{Patella} \sqcap \exists \mathsf{hasFunct}.\mathsf{Hinge} \tag{1}$$

$$\mathsf{Ginglymus} \equiv \mathsf{Joint} \sqcap \exists \mathsf{hasFunct}.\mathsf{Hinge} \tag{3}$$

$$\mathsf{Ginglymus} \equiv \mathsf{HingeJoint} \tag{5}$$

$\subseteq$-Minimal module for Σ ? $\{(1), (3), (5)\}$

Example

Let $\Sigma = \{\mathsf{Knee}, \mathsf{HingeJoint}\}$. Suppose *Galen* contains:

$$
\begin{aligned}
\mathsf{Knee} &\equiv \mathsf{Joint} \sqcap \exists \mathsf{hasPart}.\mathsf{Patella} \sqcap && (1)\\
&\quad \exists \mathsf{hasFunct}.\mathsf{Hinge} && \\
\mathsf{Patella} &\sqsubseteq \mathsf{Bone} \sqcap \mathsf{Sesamoid} && (2)\\
\mathsf{Ginglymus} &\equiv \mathsf{Joint} \sqcap \exists \mathsf{hasFunct}.\mathsf{Hinge} && (3)\\
\mathsf{Joint} \sqcap \exists \mathsf{hasPart}.(\mathsf{Bone} \sqcap \mathsf{Sesamoid}) &\sqsubseteq \mathsf{Ginglymus} && (4)\\
\mathsf{Ginglymus} &\equiv \mathsf{HingeJoint} && (5)\\
\mathsf{Meniscus} &\equiv \mathsf{FibroCartilage} \sqcap \exists \mathsf{locatedIn}.\mathsf{Knee} && (6)
\end{aligned}
$$

$\subseteq$-Minimal module for Σ? $\{(1), (2), (4), (5)\}$ and $\{(1), (3), (5)\}$

Note that a module for Σ does not necessarily contain

- all axioms that use terms from Σ
- only axioms that only use terms from Σ

Bad news for expressive ontology languages?

Big, sad theorem [Ghilardi et al. 2006]

Let $\mathcal{O}_1, \mathcal{O}_2$ be ontologies in $\mathcal{L}$ and Σ a signature.

Determining whether $\mathcal{O}_1 \equiv_\Sigma^\mathcal{L} \mathcal{O}_2$ is

ExpTime-complete	for	$\mathcal{L} = \mathcal{EL}$
2ExpTime-complete	for	$\mathcal{ALC} \leqslant \mathcal{L} \leqslant \mathcal{ALCQI}$, and
undecidable	for	$\mathcal{L} \geqslant \mathcal{ALCQO}$, including OWL

(even if $\mathcal{O}_1, \mathcal{O}_2$ are in $\mathcal{ALC}$).

Consequences for modules of expressive DLs

Extracting modules is highly complex for expressive DLs.

What to do?

1. Give up? No: modules clearly too important
2. Reduce expressivity of logic? Yes! (Not covered here.)
3. Approximate for expressive logics? Yes – but from the *right* direction!

Next: 2 approximations, i.e., sufficient conditions for inseparability

1. based on semantic locality
2. based on syntactic locality

[Cuenca Grau et al. 2009]

Model-theoretic inseparability

Remember: $\mathcal{O}_1 \equiv^{\mathcal{L}}_{\Sigma} \mathcal{O}_2$ if:

for all $\eta \in \mathcal{L}$ with $\mathsf{sig}(\eta) \subseteq \Sigma$,

$$\mathcal{O}_1 \models \eta \qquad \text{iff} \qquad \mathcal{O}_2 \models \eta$$

Good news: $\Uparrow$

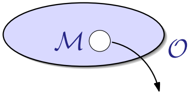
$$\{\mathcal{I}|_\Sigma \mid \mathcal{I} \models \mathcal{O}_1\} = \{\mathcal{I}|_\Sigma \mid \mathcal{I} \models \mathcal{O}_2\}$$

- i.e., $\mathcal{O}_1$ and $\mathcal{O}_2$ have the same models modulo Σ ($\mathcal{I}|_\Sigma$ is the restriction of $\mathcal{I}$ to Σ)
- shorthand: $\mathcal{O}_1 \equiv^{\mathsf{sem}}_{\Sigma} \mathcal{O}_2$ (model-inseparable)
- this notion does not depend on $\mathcal{L}$

Bad news: $\mathcal{O}_1 \equiv^{\mathsf{sem}}_{\Sigma} \mathcal{O}_2$ is undecidable already for $\mathcal{ALC}$!

Semantic locality

We can approximate model-inseparability, exploiting that $\mathcal{M}$ is a subset of $\mathcal{O}$



$$\mathcal{M} \equiv_{\Sigma}^{\mathsf{sem}} \mathcal{O}$$

$$\Updownarrow$$

every $\mathcal{I} \models \mathcal{M}$ can be extended to $\mathcal{J} \models \mathcal{O}$ with $\mathcal{I}|_{\Sigma} = \mathcal{J}|_{\Sigma}$

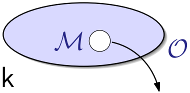
$$\Uparrow$$

every $\mathcal{I} \models \mathcal{M}$ can be extended to $\mathcal{J} \models \mathcal{O}$ with $\mathcal{I}|_{\Sigma} = \mathcal{J}|_{\Sigma}$
and $\forall X \notin \Sigma : X^{\mathcal{J}} = \emptyset$

$$\Updownarrow$$

every $\alpha \in \mathcal{O} \setminus \mathcal{M}$ is **semantically local w.r.t. $\Sigma \cup \mathsf{sig}(\mathcal{M})$:**
α, with all terms not in $\Sigma \cup \mathsf{sig}(\mathcal{M})$ replaced by $\bot$, is a tautology

From semantic to syntactic locality



- Semantic locality involves tautology check

  $\rightsquigarrow$ can be tested using a reasoner

  $\rightsquigarrow$ has the same complexity as standard reasoning

- A syntactic approximation that can be tested in poly-time:

  syntactic locality

  (describes “obviously” sem. local axioms via a grammar)

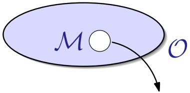
- Both notions lead to modules that are
  - $(\Sigma \cup \mathrm{sig}(\mathcal{M}))$-inseparable from $\mathcal{O}$
  - not necessarily minimal

Examples of syntactically (non)-local axioms

$\overline{B} \sqsubseteq A$	form $C \sqsubseteq C^{\emptyset} \leadsto$ not $\{\overline{B}, \ldots\}$-local
$A \sqsubseteq \overline{B} \sqcap \exists r.\overline{C}$	form $C^{\emptyset} \sqsubseteq C \leadsto \{\overline{B}, \overline{C}\}$-local
$X \sqcap A \sqsubseteq Y$	is Σ-local if, e.g., $A \notin \Sigma$
$\overline{B} \sqcap \exists r.\overline{C} \sqsubseteq A$	is $\{\overline{B}, \overline{C}\}$-local
$\overline{A} \sqsubseteq \overline{A} \sqcup \overline{B}$	is not $\{\overline{A}, \overline{B}\}$-local, yet a tautology!

Module extraction

Module extraction algorithm:



- $\mathcal{M} \leftarrow \emptyset$
- While α not local w.r.t. $\Sigma \cup \mathrm{sig}(\mathcal{M})$, for some $\alpha \in \mathcal{O} \setminus \mathcal{M}$,
  do $\mathcal{M} \leftarrow \mathcal{M} \cup \{\alpha\}$
- Output $\mathcal{M}$

Variations:

- this notion: (semantic/syntactic) $\bot$-module
- dual notion: (semantic/syntactic) $\top$-module
- smaller modules by nesting $\top$- and $\bot$-module extraction: $\top\bot^*$-modules

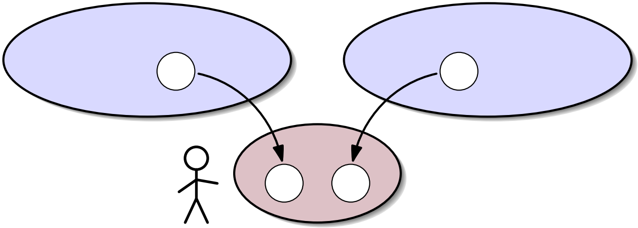
And now . . .

Summary

- Inseparability/coverage is a guarantee relevant (not only) for reuse
- Approximation of minimal covering modules via locality
- Modules based on syntactic locality can be extracted efficiently in logics up to $\mathcal{SROIQ}$ (OWL 2)
- Tool support for extracting modules:
  http://owl.cs.manchester.ac.uk/modularity
  http://owlapi.sourceforge.net/
- This line of research is rather new for DLs and ontology languages, and many questions are (half)open.

An import/reuse scenario

“Borrow” knowledge from external ontologies

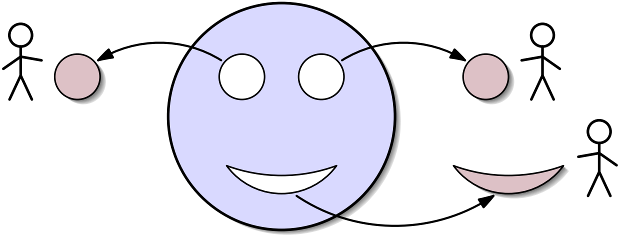


- Provides access to well-established knowledge
- Doesn’t require expertise in external disciplines

This scenario is well-understood and implemented.

A collaboration scenario

Collaborative ontology development

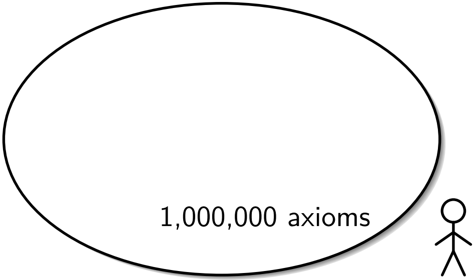


- Developers work (edit, classify) locally
- Extra care at re-combination
- Prescriptive/analytic behaviour

This approach is mostly understood, but not implemented yet.

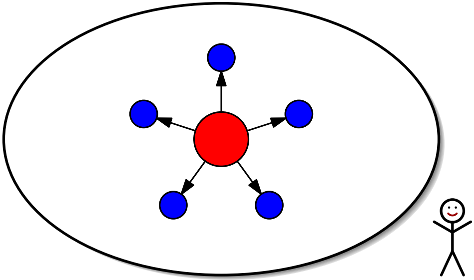
Understanding and/or structuring an ontology

Compute the modular structure of an ontology



Understanding and/or structuring an ontology

Compute the modular structure of an ontology



This is work in progress.

See also . . .

. . . slides from ESSLLI 2011 course “Modularity in Ontologies”

http://www.informatik.uni-bremen.de/~ts/teaching

. . . the references at the end of this presentation

Over to Uli for Justifications!

References

B. Cuenca Grau, I. Horrocks, Y. Kazakov, U. Sattler.
Extracting Modules from Ontologies: a Logic-Based Approach.
In H. Stuckenschmidt et al., eds: *Modular Ontologies*, pages 159–186, vol. 5445 of LNCS, Springer, 2009.
http://www.springerlink.com/content/qq732374182825q0/
http://web.comlab.ox.ac.uk/oucl/work/bernardo.cuenca.grau/publications/paperJAIR.pdf (previous version)

S. Ghilardi, C. Lutz, F. Wolter.
Did I Damage My Ontology? A Case for Conservative Extensions in Description Logics.
In *Proc.* KR, pages 187–197, 2006.
http://www.csc.liv.ac.uk/~frank/publ/GLWZshort.pdf
http://www.csc.liv.ac.uk/~frank/publ/GLWZlong.ps

B. Konev, C. Lutz, D. Walther, and F. Wolter.
Formal Properties of Modularisation.
In H. Stuckenschmidt et al., eds: *Modular Ontologies*, pages 25–66, vol. 5445 of LNCS, Springer, 2009.
http://www.csc.liv.ac.uk/~frank/publ/modulebook.pdf