Modularity in Ontologies: Recent Advances

Thomas Schneider[1] Dirk Walther[2]

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

[2]Faculty of Informatics, Technical University of Madrid, Spain

ESSLLI, 5 August 2011

Plan for today

1. Tipps, tricks and pitfalls using locality
2. Extending coverage
3. Atomic decomposition
4. Collaborative ontology development
5. Wrap-up

And now . . .

1. Tipps, tricks and pitfalls using locality
2. Extending coverage
3. Atomic decomposition
4. Collaborative ontology development
5. Wrap-up

Remember . . .

- Safety and economy/coverage are important guarantees (not only) for reuse.
- They can be defined using inseparability.
- They can be approximated using locality.
- Modules based on syntactic locality can be extracted efficiently in logics up to OWL.
- There is tool support for extracting modules.
  `http://owl.cs.manchester.ac.uk/modularity`
  `http://owlapi.sourceforge.net/`

Are locality-based modules economic?

Locality-based modules are not minimal in general:
they include

- many `EquivalentClass` axioms
- tautologies
- axioms about individuals

Research goal: smaller (but still efficiently computable) modules

---

(Is small really beautiful?)

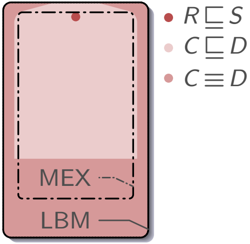
Yes, they are!

- Experiments with SNOMED
  (health care; restricted language; 350,000 axioms)
- Compared modules for 24,000 terms from intensive care unit
- Locality-based modules (LBM) $\Leftrightarrow$ minimal modules (MEX)

Yes, they are!

- Experiments with SNOMED
  (health care; restricted language; 350,000 axioms)
- Compared modules for 24,000 terms from intensive care unit
- Locality-based modules (LBM) ⇔ minimal modules (MEX)
- Results:

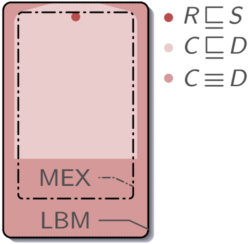
# axioms MEX	# axioms LBM
10%	15%
4–5 s	4–7 s



Yes, they are!

- Experiments with SNOMED
  (health care; restricted language; 350,000 axioms)
- Compared modules for 24,000 terms from intensive care unit
- Locality-based modules (LBM) ⇔ minimal modules (MEX)
- Results:

# axioms	
MEX	LBM
10%	15%
4–5 s	4–7 s



"I want a bit more."—Then extend your topic! ⇝ Small *is* beautiful.

What to do if safety is violated?

Q: Help, my tool found a non-local axiom! What shall I do?

A: There are several possibilities:

What to do if safety is violated?

Q: Help, my tool found a non-local axiom! What shall I do?

A: There are several possibilities:

(1) Your axiom might violate locality, but not safety.
(Remember: locality *approximates* safety.)

What to do if safety is violated?

Q: Help, my tool found a non-local axiom! What shall I do?

A: There are several possibilities:

(1) Your axiom might violate locality, but not safety.
(Remember: locality *approximates* safety.)

$\leadsto$ Call 0800-inseparability,
ask your favourite logician to decide whether the axiom is safe.

What to do if safety is violated?

Q: Help, my tool found a non-local axiom! What shall I do?

A: There are several possibilities:

(2) Your axiom violates safety?
Do you have a good reason to write it?
If yes, keep it, but be aware that you've amended the topic!

What to do if safety is violated?

Q: Help, my tool found a non-local axiom! What shall I do?

A: There are several possibilities:

(3) Want to repair a non-local axiom?

- Delete it.
- Modify it:

$$\text{Bird} \sqsubseteq \text{Flies} \quad \rightsquigarrow \quad \text{Bird} \sqcap \neg\text{Penguin} \sqsubseteq \text{Flies}$$

$$\text{Bird} \sqsubseteq \text{Flies} \quad \rightsquigarrow \quad \text{Bird} \sqsubseteq \text{Flies} \sqcup \text{Penguin}$$

- Explanations . . .

What to do if safety is violated?

Q: Help, my tool found a non-local axiom! What shall I do?

A: There are several possibilities:

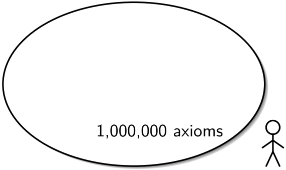
(4) Prescriptive/analytic safety checking . . .

Independence

- Required property: If $\mathcal{O}_1$ is safe for Σ_2 and Σ_3, then $\mathcal{O}_1 \cup \mathcal{O}_2$ should be safe for Σ_3.

- Difficult to achieve prescriptively: only holds under restrictive preconditions
- Advice: treat independence analytically.

Specifying the topic



Which terms do I want to import?

- Ask 0800-domainexpert for a list of terms.
- Browse through the class hierarchy and find suitable terms.
- Shopping for symbols:
  - Select terms.
  - Get a preview of the module.
  - If you're satisfied, check out the module.

And now . . .

1. Tipps, tricks and pitfalls using locality
2. Extending coverage
3. Atomic decomposition
4. Collaborative ontology development
5. Wrap-up

Remember: module for a signature

Definition

$\mathcal{M} \subseteq \mathcal{O}_2$ is **a module for Σ in $\mathcal{O}_2$** w.r.t. $\mathcal{L}$ if,

for every $\mathcal{L}$-ontology $\mathcal{O}_1$ with $\mathrm{sig}(\mathcal{O}_1) \cap \mathrm{sig}(\mathcal{O}_2) \subseteq \Sigma$,
$\mathcal{O}_1 \cup \mathcal{O}_2 \equiv^{\mathcal{L}}_{\mathrm{sig}(\mathcal{O}_1)} \mathcal{O}_1 \cup \mathcal{M}$.

Observation

If $\mathcal{M} \subseteq \mathcal{O}_2$ and $\mathcal{O}_2$ is a model Σ-c.e. of $\mathcal{M}$ $(\mathcal{O}_2 \equiv^{\mathsf{SO}}_{\Sigma} \mathcal{M})$,
then $\mathcal{M}$ is a module for Σ in $\mathcal{O}_2$ w.r.t. any $\mathcal{L} \leqslant \mathsf{SO}$

$\rightsquigarrow$ Does it suffice to require $\mathcal{O}_2 \equiv^{\mathsf{SO}}_{\Sigma} \mathcal{M}$?

Self-contained modules

- Remember: a module usually contains terms *not* from Σ
- What can happen if $\mathcal{O}_2 \not\equiv^{\mathcal{L}}_{\mathsf{sig}(\mathcal{M}\setminus\Sigma)} \mathcal{M}$?

Definition

$\mathcal{M} \subseteq \mathcal{O}_2$ is **a self-contained Σ-module in $\mathcal{O}_2$** w.r.t. $\mathcal{L}$ if $\mathcal{O}_2 \equiv^{\mathcal{L}}_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \mathcal{M}$.

Self-contained modules

- Remember: a module usually contains terms *not* from Σ
- What can happen if $\mathcal{O}_2 \not\equiv^{\mathcal{L}}_{\mathsf{sig}(\mathcal{M}\setminus\Sigma)} \mathcal{M}$?

Definition

$\mathcal{M} \subseteq \mathcal{O}_2$ is **a self-contained Σ-module in $\mathcal{O}_2$** w.r.t. $\mathcal{L}$ if $\mathcal{O}_2 \equiv^{\mathcal{L}}_{\Sigma\cup\mathsf{sig}(\mathcal{M})} \mathcal{M}$.

- Locality-based modules are self-contained w.r.t. SO.
- Every self-contained Σ-module in $\mathcal{O}_2$ is a Σ-module in $\mathcal{O}_2$ (robustness under vocabulary restriction)

Depleting modules

- Modules preserve entailments: $\mathcal{O}_2 \equiv^{\mathsf{SO}}_{\Sigma} \mathcal{M}$ means for all $\alpha \in \mathsf{SO}$ with $\mathsf{sig}(\alpha) \subseteq \Sigma$, $\mathcal{O}_2 \models \alpha$ iff $\mathcal{M} \models \alpha$.
- What if we want to guarantee that *all* reasons for the preserved entailments are in $\mathcal{M}$?

  Modules for explanation services / maintaining $\mathcal{O}_2$

Definition

$\mathcal{M} \subseteq \mathcal{O}_2$ is **a depleting Σ-module in $\mathcal{O}_2$** w.r.t. $\mathcal{L}$ if

$\mathcal{O}_2 \setminus \mathcal{M} \equiv^{\mathcal{L}}_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$.

Depleting modules

- Modules preserve entailments: $\mathcal{O}_2 \equiv^{\mathsf{SO}}_{\Sigma} \mathcal{M}$ means for all $\alpha \in \mathsf{SO}$ with $\mathsf{sig}(\alpha) \subseteq \Sigma$, $\mathcal{O}_2 \models \alpha$ iff $\mathcal{M} \models \alpha$.
- What if we want to guarantee that *all* reasons for the preserved entailments are in $\mathcal{M}$?

  Modules for explanation services / maintaining $\mathcal{O}_2$

Definition

$\mathcal{M} \subseteq \mathcal{O}_2$ is **a depleting Σ-module in $\mathcal{O}_2$** w.r.t. $\mathcal{L}$ if

$\mathcal{O}_2 \setminus \mathcal{M} \equiv^{\mathcal{L}}_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$.

- Locality-based modules are depleting w.r.t. SO.
- If $(\mathcal{L}, \mathcal{L})$ is robust under replacements, then
  - every depleting Σ-module is a self-contained Σ-module;
  - every depleting Σ-module a Σ-module.

Depleting module notions lead to *unique* minimal modules:

Suppose $\mathcal{M}_1, \mathcal{M}_2$ are depleting Σ-modules of $\mathcal{O}$:

$$\mathcal{O} \setminus \mathcal{M}_i \equiv^{\mathcal{L}}_{\Sigma \cup \mathsf{sig}(\mathcal{M}_i)} \emptyset$$

Via robustness under vocabulary restrictions:

$$\mathcal{O} \setminus \mathcal{M}_i \equiv^{\mathcal{L}}_{\Sigma \cup \mathsf{sig}(\mathcal{M}_1 \cap \mathcal{M}_2)} \emptyset$$

Hence: $$\mathcal{O} \setminus \mathcal{M}_1 \equiv^{\mathcal{L}}_{\Sigma \cup \mathsf{sig}(\mathcal{M}_1 \cap \mathcal{M}_2)} \mathcal{O} \setminus \mathcal{M}_2 \equiv^{\mathcal{L}}_{\Sigma \cup \mathsf{sig}(\mathcal{M}_1 \cap \mathcal{M}_2)} \emptyset$$

Robustness under joins(!):

$$(\mathcal{O} \setminus \mathcal{M}_1) \cup (\mathcal{O} \setminus \mathcal{M}_2) \equiv^{\mathcal{L}}_{\Sigma \cup \mathsf{sig}(\mathcal{M}_1 \cap \mathcal{M}_2)} \emptyset$$

i.e.: $$\mathcal{O} \setminus (\mathcal{M}_1 \cap \mathcal{M}_2) \equiv^{\mathcal{L}}_{\Sigma \cup \mathsf{sig}(\mathcal{M}_1 \cap \mathcal{M}_2)} \emptyset$$

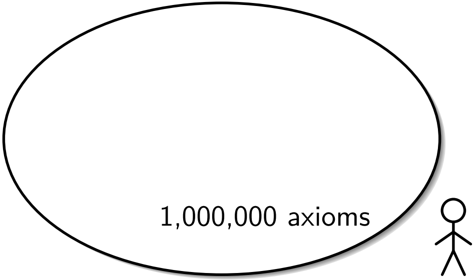
Therefore, $\mathcal{M}_1 \cap \mathcal{M}_2$ is a smaller depleting Σ-module of $\mathcal{O}$.

I.e., depleting modules are closed under intersection
$\Rightarrow$ there exists a unique minimal depleting Σ-module.

And now . . .

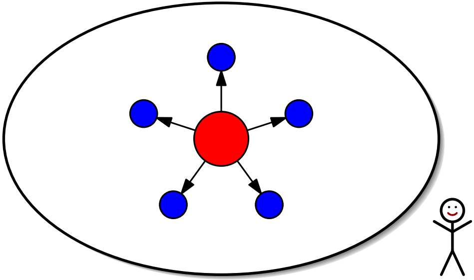
What is my ontology about?

We can't inspect all its axioms.



What is my ontology about?

We can inspect its modular structure, obtained *a posteriori*.



We bet Robert Stevens

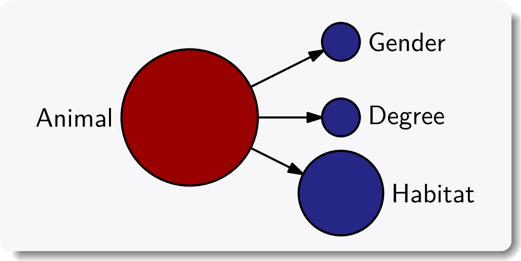
- Ontology about periodic table of the chemical elements
- Logical structure ≈ intended modelling?
  - What is its logical structure?
  - What are its main parts?

We bet Robert Stevens

- Ontology about periodic table of the chemical elements
- Logical structure ≈ intended modelling?
  - What is its logical structure?
  - What are its main parts?
- Challenge: *automatic* partition into meaningful modules

Modular structure with existing tools

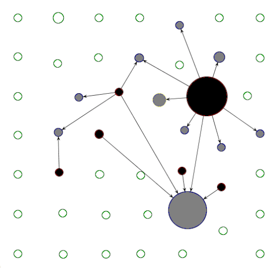
Partition of *Koala* via E-connections in Swoop



- importing part
- imported but non-importing part
- isolated part

→ "imports vocabulary from"

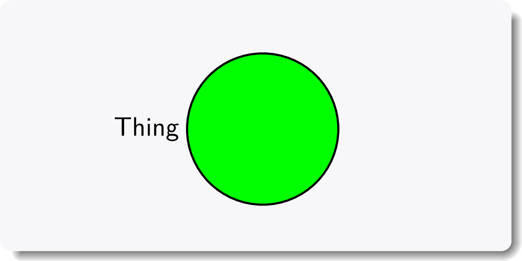
Partition for ontology *SWEET*



- ● importing part
- ● imported but non-importing part
- ○ isolated part

→ "imports vocabulary from"

Partition for ontology *Periodic*



- importing part (red)
- imported but non-importing part (blue)
- isolated part (green)

→ "imports vocabulary from"

Modular structure via LBMs – goals

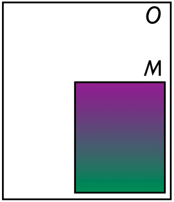
- Draw conclusions on characteristics of an ontology:
  - To which extent does $\mathcal{O}$ cover its topics?
  - How strongly are certain terms connected in $\mathcal{O}$?
  - What is the axiomatic richness of $\mathcal{O}$?
  - Does $\mathcal{O}$ have superfluous parts?
  - Agreement between logical and intended intuitive modelling?
- Guide users in choosing the right topic(s)

Ontologies & Modules

- An **ontology** is a finite set of axioms in a (description) logic
- A **module** $M(\Sigma, O) \subseteq O$ encapsulates knowledge w.r.t a signature Σ:

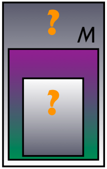
for all α with $\mathrm{sig}(\alpha) \subseteq \Sigma$: $O \models \alpha$ iff $M(\Sigma, O) \models \alpha$

M({part}, Mereology.owl) = {Trans: part,
part InverseOf: PartOf,
Trans: partOf}



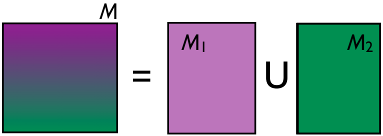
Modular *Structure*

- Modules are great...if you know your (seed) signature...
  - and for "module local" tasks such as reuse
- *Single* module extraction does *not* help if you
  - do *not* know the *right* seed signature
  - want to understand *other* modules
  - want to understand *axiom dependency structure*



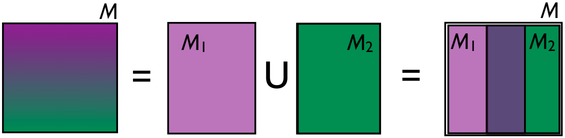
- To analyse the *modular structure* of the ontology:
  - **significant** modules
  - **significant** relations between modules
  - ...which reveals logical dependence between axioms

Are all modules significant?



- To understand M, one must
  - understand the dependancy structure of M_1
  - understand the dependancy structure of M_2
  - **nothing** else: M_1 and M_2 have no further dependancies
- ➡ M is **not** significant: it is a ***fake*** module
  - Thus, M_1 and M_2 may be "significant"
  - knowing that M is "only" a union is important

Are all modules significant?

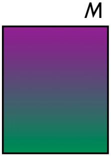


- To understand M, one must
  - understand the dependancy structure of M_1
  - understand the dependancy structure of M_2
  - **nothing** else: M_1 and M_2 have no further dependancies

➡ M is **not** significant: it is a ***fake*** module
  - Thus, M_1 and M_2 may be "significant"
  - knowing that M is "only" a union is important

Are all modules significant?

- Consider a module M that is **not *fake***
- To understand M, one has to understand M **as a whole**
  - all axioms in M logically interact
  - in different ways – but interact
- Not fake implies significant: ***genuine***



Ratio of Fake to Genuine

- Given a set of genuine modules
  - unions lead to fake modules,
  - ➡ the space of fake modules is large (exponential)
  - but not *every* union of genuine modules is a module
- The cardinality of the set of all modules can and does grow exponentially in the size of *O*
  - See Parsia & Schneider, KR 2010 & WoMO 2010
- *Is module growth primarily due to trivial combinations?*
  - are most modules **fake**?

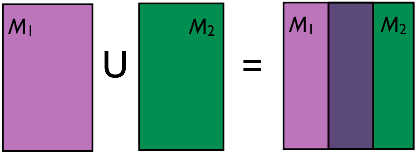
Yes!

Theorem 1: Each genuine module is the smallest module for some axiom $\alpha \in O$.

- The family of genuine modules is linear in $|O|$
  - ★ Most modules are fake!
- Proof exploits properties of modules
  - uniqueness, monotonicity, self-containedness, …
  - which are satisfied by all locality-based modules

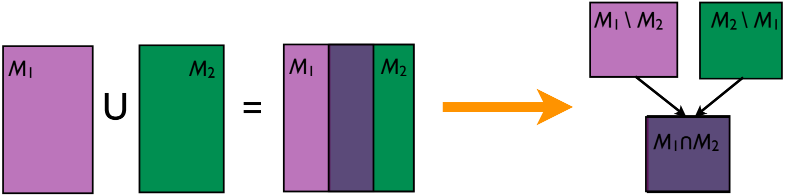
Relations between Modules

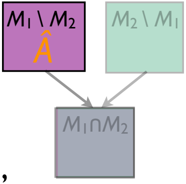
- Genuine modules may overlap
- This exposes significant logical dependence between axioms:
  - axioms in $M_1 \setminus M_2$ depend on axioms in $M_1 \cap M_2$



Relations between Modules

- Genuine modules may overlap
- This exposes significant logical dependence between axioms:
  - axioms in $M_1 \setminus M_2$ depend on axioms in $M_1 \cap M_2$



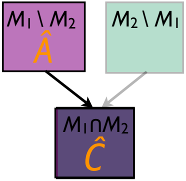


Atoms

- $\hat{A} \subseteq O$ is an **atom** if it is a maximal set s.t., for each module M, either $\hat{A} \subseteq M$ or $\hat{A} \cap M = \emptyset$.
  - ➡ The smallest module for an axiom α contains the whole atom to which α belongs!
  - ➡ Axioms in an atom are logically interdependent
  - ➡ Any two atoms are disjoint
  - ➡ The family of atoms is a partition of the ontology
    - Only linearly many atoms
  - ➡ Each GM is a disjoint union of atoms

Proposition: There is a 1-1 correspondence between genuine modules and atoms.

Atomic Decomposition



- Dependence between atoms:
  - $\hat{A} \succcurlyeq \hat{C}$ if, for each M: $\hat{A} \subseteq M$ implies $\hat{C} \subseteq M$
  - Axioms in $\hat{A}$ logically depend on axioms in $\hat{C}$

Theorem 2: The relation $\succcurlyeq$ is reflexive, antisymmetric, and transitive.

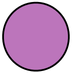
- a Hasse diagram exposes 2 logical dependencies amongst axioms in atoms & between atoms

Mereology Ontology

42 axioms
1952 modules

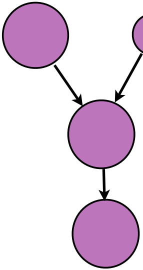
Mereology Ontology

42 axioms
1952 modules

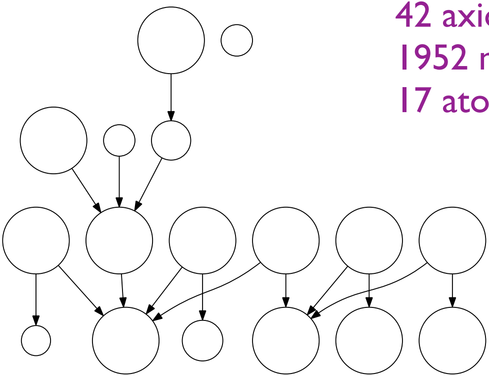


Mereology Ontology

42 axioms
1952 modules



Mereology Ontology



42 axioms
1952 modules
17 atoms/GMs

- Can we compute all genuine modules?
  - and all atoms
  - with their dependencies?
  - ...without computing all modules?!

Yes!

- Remember:

> Theorem 1: Each genuine module is the smallest module for some axiom $\alpha \in O$.

- extract $M(\mathrm{sig}(\alpha), O)$
  - $\leq$ linearly many module extractions
- AD induced by the comparison of GMs
  - quadratic procedure

In Reality?

- We have decomposed 181 OWL ontologies in Bioportal
- Decomposability: average
  - nr. axioms/atom: 1.73
  - max nr. axioms/atom: 86
  - nr. axioms/GM: 66
  - max nr. axioms/GM: 143

Future Work

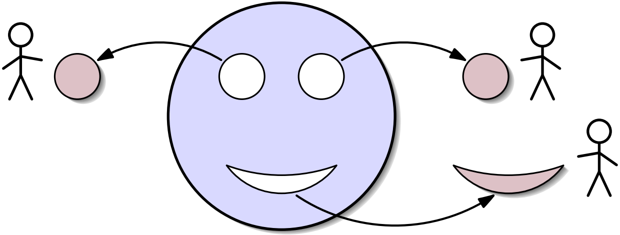
- More on dependency of axioms
  - between atoms and **sets** of atoms
- Labels for atoms
  - different labels for different tasks
- Applications
  - All Module Count: see WoMO 2011
  - Fast Module Extraction
  - Topicality for Ontology Comprehension: see ICCS 2011
  - ...

And now ...

1. Tipps, tricks and pitfalls using locality
2. Extending coverage
3. Atomic decomposition
4. Collaborative ontology development
5. Wrap-up

Aim

- Raise and discuss open questions regarding collaborative ontology development
- Can current notions of safety and coverage capture all requirements?
- Analyse (collaborative) ontology development and relate to modularity



Roles

- roles for interacting with ontology
  - curator (domain expert)
    - check functionality of ontologies
    - propose changes
  - developer
    - implement changes in ontologies
  - user
    - ask queries in some interface $(\mathcal{QL}, \Sigma)$
- participants can assume roles
  - distribution of roles (access rights) among people
    - one curator/developer and many users
    - several curators/developers and many users
    - everyone is curator/developer/user
  - example: SNOMED CT
    - small group of curators/developers
    - large group of users

Interface

Interface $(\mathcal{QL}, \Sigma)$ consists of a query logic $\mathcal{QL}$ and a signature Σ.
Who uses which interface?

- curator/developer
  - $\mathcal{QL}$: largest possible that a user could use
  - Σ: vocabulary from the domain of expertise
- user:
  - $\mathcal{QL}$ and Σ depend on application

Scenario

- single curator (later multi-agent case)

Working on the entire ontology directly or via modules?

- aspects: feasibility, economy, control/access
  - user: comprehension, manageability, navigation
  - tool: efficient processing (loading, reasoning,...)
- use of modules becomes more pressing the larger O

$\rightsquigarrow$ Suppose the ontology to be edited is large and we resort to using modules.

Workflow

Workflow for editing an ontology O via modules:

1. extract a module M from O
2. externally modify M; obtain M'
3. replace M with M' in O

(1) extract M from O

How to determine a module in the ontology?

- extracting a module
  - compute module
  - check out (co)
- curator extracts module M wrt. his interface $(\mathcal{QL}, \Sigma)$
- allowed to arbitrarily change functionality of M wrt. $(\mathcal{QL}, \Sigma)$ (i.e. can change $Th_{\Sigma}^{\mathcal{QL}}(M)$)
- types of modules:
  - covering: $M \equiv_{\Sigma}^{\mathcal{QL}} O$
  - self-contained: $M \equiv_{\Sigma \cup \mathsf{sig}(M)}^{\mathcal{QL}} O$
    - implies all consequences over terms in M
  - depleting: $O \setminus M \equiv_{\Sigma \cup \mathsf{sig}(M)}^{\mathcal{QL}} \emptyset$
    - exhaustive: contains everything about Σ-terms and additional terms in M
    - functionality reduction of $M \rightsquigarrow$ same for O

(2) externally modify M and obtain M'

What can we do with the module M once we have extracted it from O?

- syntactic changes in M:
  - adding/deleting/changing of axioms
  - signature change: within Σ / within $\Sigma \cup \mathrm{sig}(M)$ / addition of new symbols not in M
- syntactic changes imply functional changes
  - addition / deletion of $(\mathcal{QL},\Sigma)$-consequences of M

(3) replace module M with M'

Reintegration of the modified module into the original ontology.

- commit (ci) – obtain O' from O by replacing M with M'
- possible requirements:
  - do not change meaning of terms remaining in O
    - functionality of O over $(\mathcal{QL},\ \mathsf{sig}(O) \setminus (\Sigma \cup \mathsf{sig}(M)))$
      $\rightsquigarrow$ safety? reasonable?
  - want to change meaning of terms remaining in O
    - functionality of O over $(\mathcal{QL},\ \mathsf{sig}(O) \setminus (\Sigma \cup \mathsf{sig}(M)))$
- one curator may not be qualified to judge all effects
  (e.g. change may affect different area of expertise)
  $\rightsquigarrow$ several curators

Multi-agent Case

Scenario for 2 agents

- relationships of modules for agents
  - module: disjointness / intersecting
  - functionality: two interfaces $(\mathcal{QL}_1, \Sigma_1)$ vs. $(\mathcal{QL}_2, \Sigma_2)$
    $\leadsto$ Disjointness required? If so, what does it mean for interfaces?
- interleaving workflows of agents
  (a) overlapping: co1, co2, ci1, ci2
  (b) contained: co1, co2, ci2, ci1
- conflict
  - occurs after ci2 (agent 2 commits his modified module)
  - syntactical vs. functional conflict
    - syntactical conflict: ci1 changes M of second agent
    - functional conflict: changes of agent 1 affect functionality of entire ontology wrt. agent 2's interface

Modifying only the *functionality* of a module

Idea:

- Curator interacts with tool via interface
- Tool replaces developer and implements C's requested changes
- Axioms are internally represented and usually hidden from developer/user

To modify $\mathcal{O}$, Curator asks for functionality (queries in his/her interface), then requests to

- remove unwanted functionality (tool uses justification)
- add missing functionality (tool uses justification)

Questions:

- Fully automatic modifications, or with developer interaction?
- Which interface does developer use?
  ($\geqslant$ user's $\mathcal{QL}$? Separation between developers/users?)

And now . . .

What we have covered

- Typical use cases for logic-based modularity
- Overview of module extraction approaches
- Theory and practice of logic-based a-posteriori modularisation approaches
  - Modules and interfaces
  - Inseparability notions and robustness properties
  - Requirements from ontology engineering
  - Locality-based modules
- Related notions
  - Logical difference
  - Forgetting/uniform interpolants
- Current work: atomic decomposition
- Open questions (see next slide)

Current and Future Tool Support for Expressive DLs

Interesting questions:

- How can we support designer of $\mathcal{O}_1$ to pick $\mathcal{O}_2, \Sigma, \mathcal{L}$ and
  - import a Σ-module in $\mathcal{O}_2$?
  - make sure that $\mathcal{O}_1$ remains Σ-safe?
- How can we **show** $\mathcal{M}$ (Σ-module in $\mathcal{O}_2$) to designer of $\mathcal{O}_1$ to ensure that they really want to import it?
- How can we ensure safety of $\mathcal{O}_1$ for various signatures if "imported" ontologies are unknown?
- How can we use (semantic and syntactic) locality to compute "good" modules?
- How can we **visualise** the modular structure of an ontology?
  ➽ *Friday's lecture*
- How can we **explain** that X is not safe for Y?
- How can we use modules to speed up reasoning?

That's it.

Thank you for coming!

dirk.walther-AT-upm.es
tschneider-AT-informatik.uni-bremen.de

There's the Workshop on Modular Ontologies (WoMO) next week.