Decomposition and Modular Structure of BioPortal Ontologies

Chiara Del Vescovo[1], Damian Gessler[2], *Pavel Klinov*[2], Bijan Parsia[1], Uli Sattler[1], Thomas Schneider[3], and Andrew Winget[4]

[1] The University of Manchester
[2] University of Arizona, BIO5 Institute
[3] Universität Bremen
[4] St. John College

Motivational use case: SSWAP

Simple Semantic Web Architecture and Protocol

http://sswap.info

Framework for discovery and invocation of semantic Web services

- Service descriptions use terms from arbitrary ontologies
- SSWAP finds services based on their semantics
- Web clients invoke services based on their semantics

Motivational use case: SSWAP

Simple Semantic Web Architecture and Protocol

http://sswap.info

Framework for discovery and invocation of semantic Web services

- Service descriptions use terms from arbitrary ontologies
- SSWAP finds services based on their semantics
- Web clients invoke services based on their semantics

Transaction-time reasoning is absolute key

- Looking at each axiom is infeasible
- Traditional single-file ontology representation is unsatisfactory

Modular approach

Key idea:

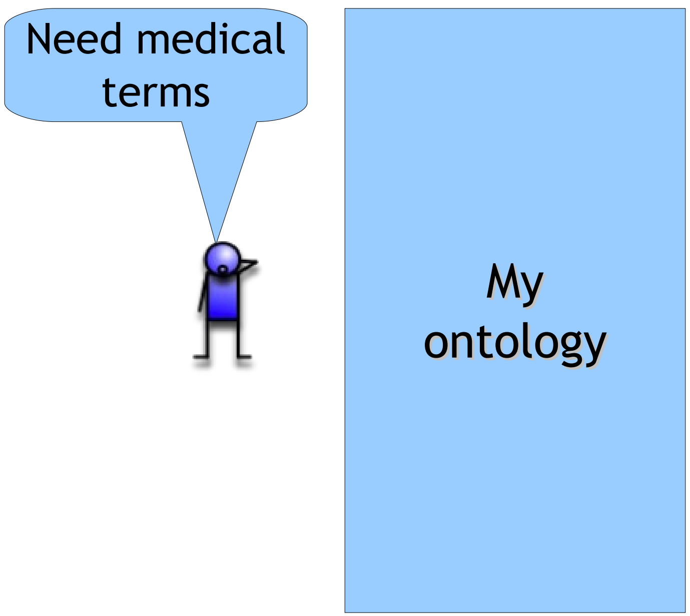
to reason about specific terms
use modules other than entire ontologies

Modularity:

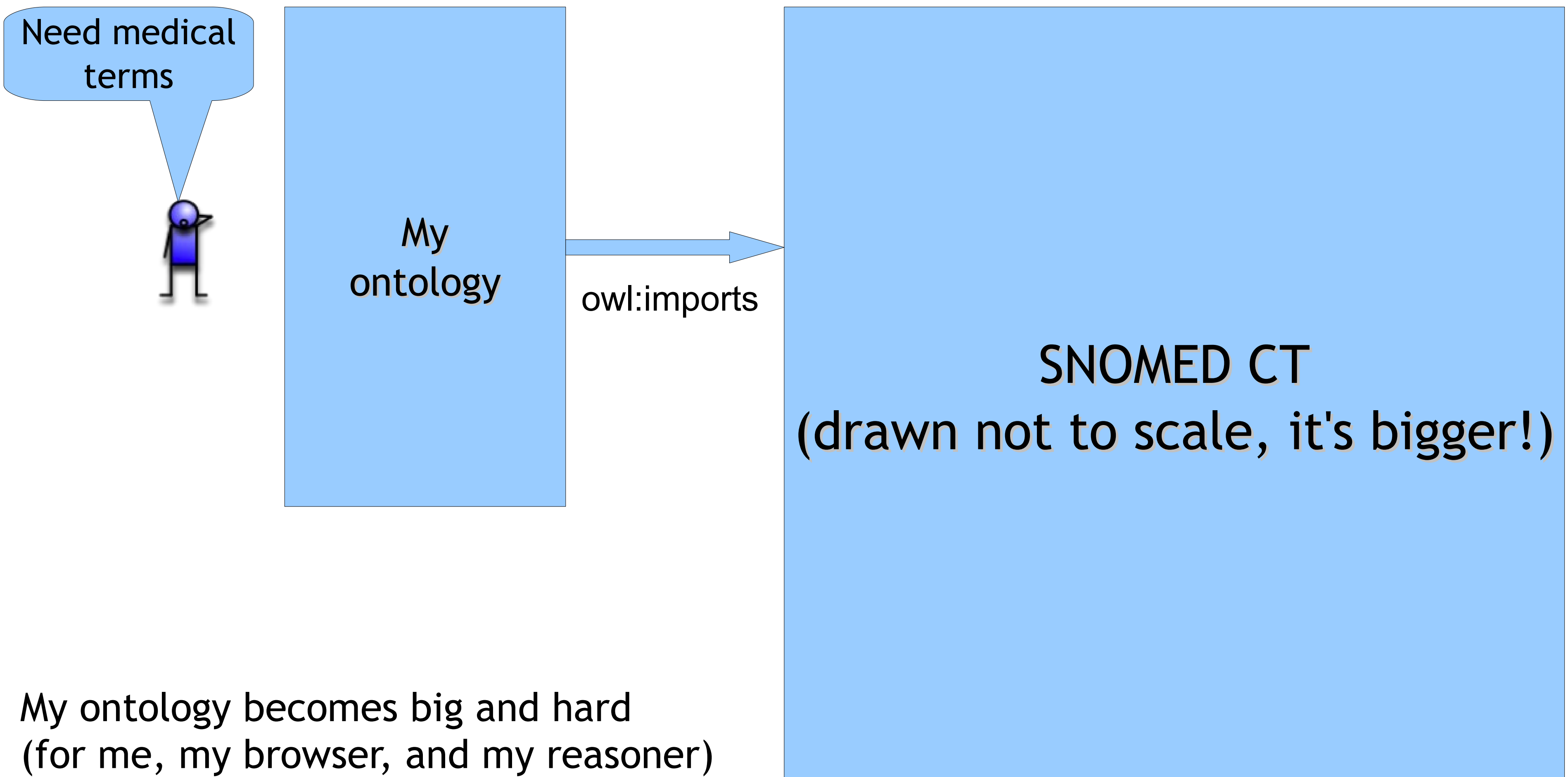
theory/algorithms to define/compute minimal, logically complete, and relevant fragments

But ontologies are still maintained as plain sets of axioms

Reusing ontologies

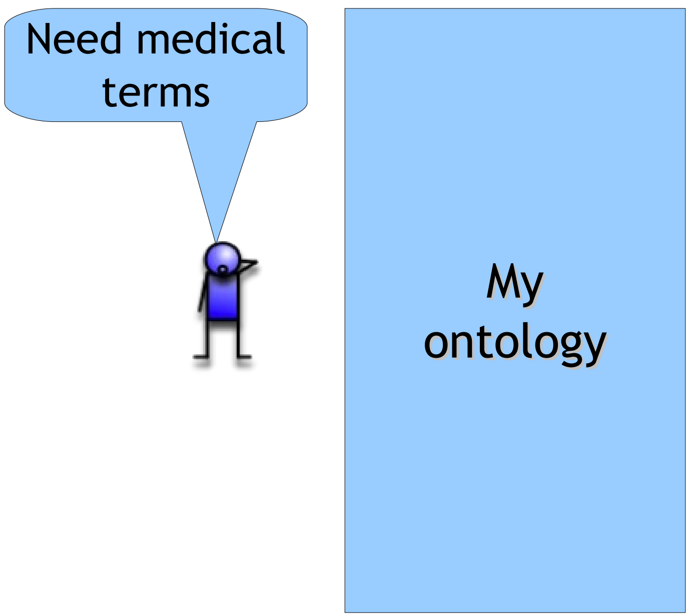


Reusing ontologies

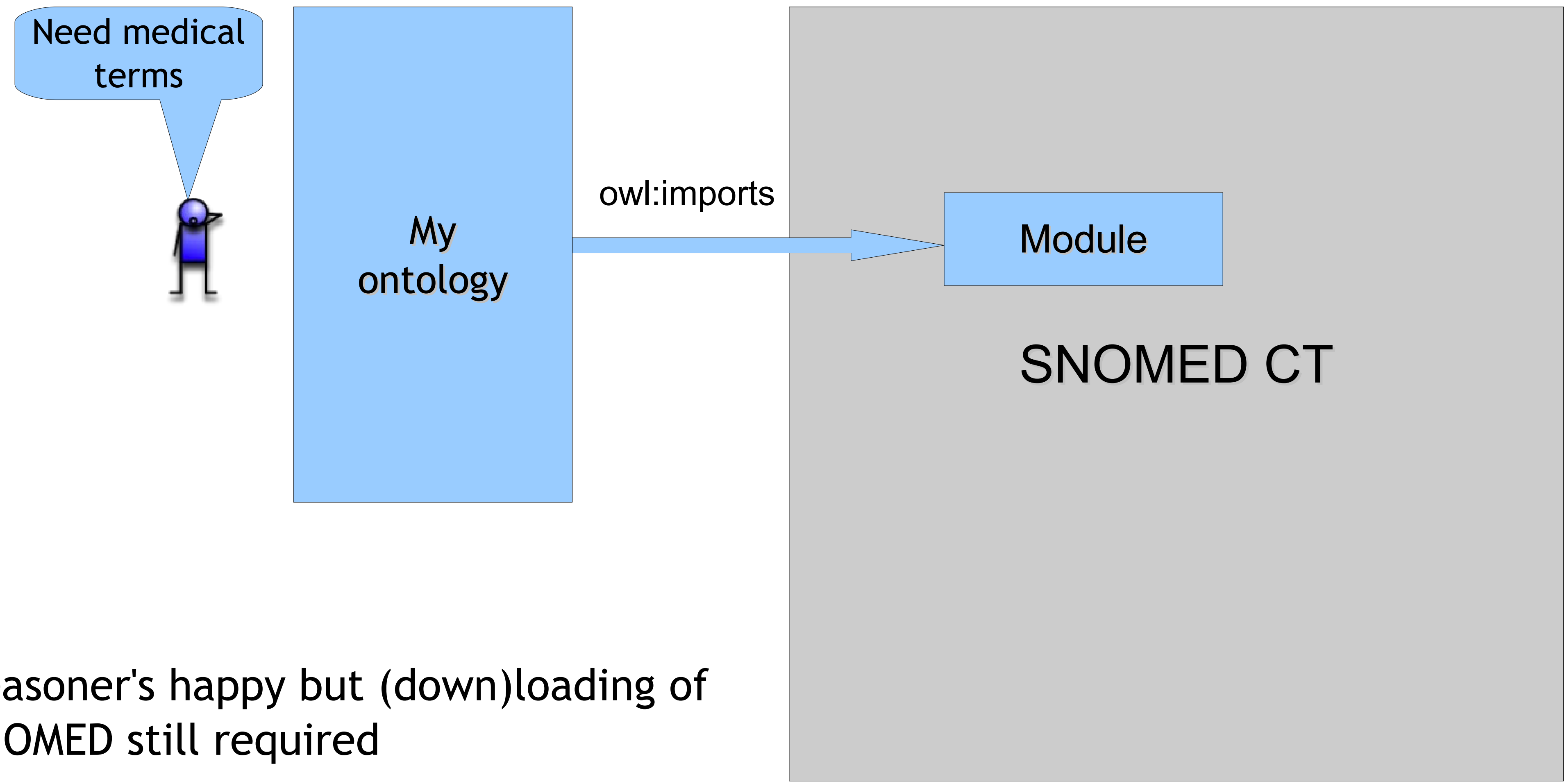


My ontology becomes big and hard
(for me, my browser, and my reasoner)

We can do better with modules

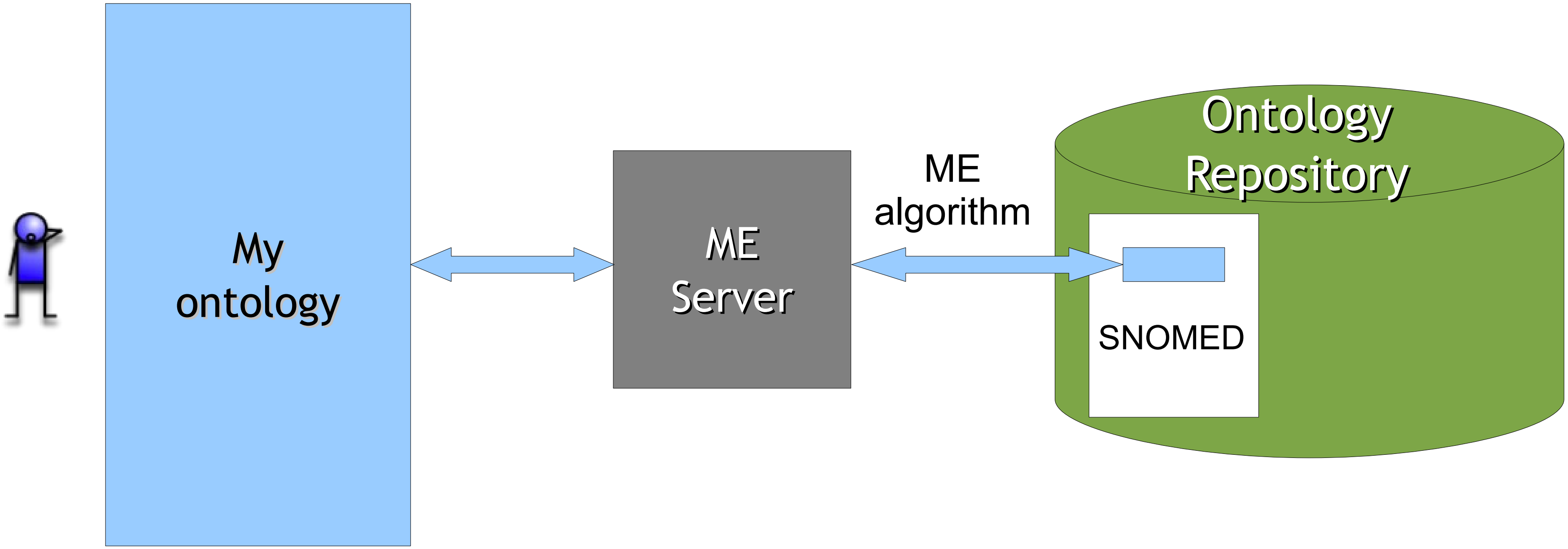


We can do better with modules



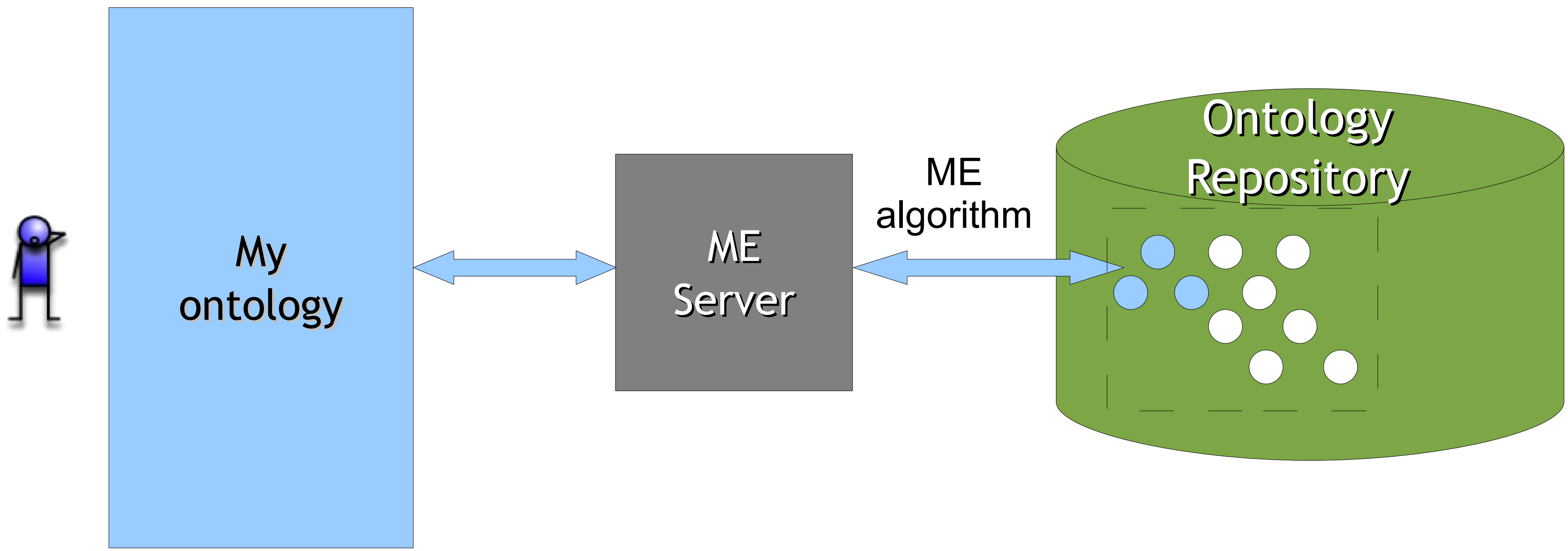
Reasoner's happy but (down)loading of SNOMED still required

Serving modules on Web



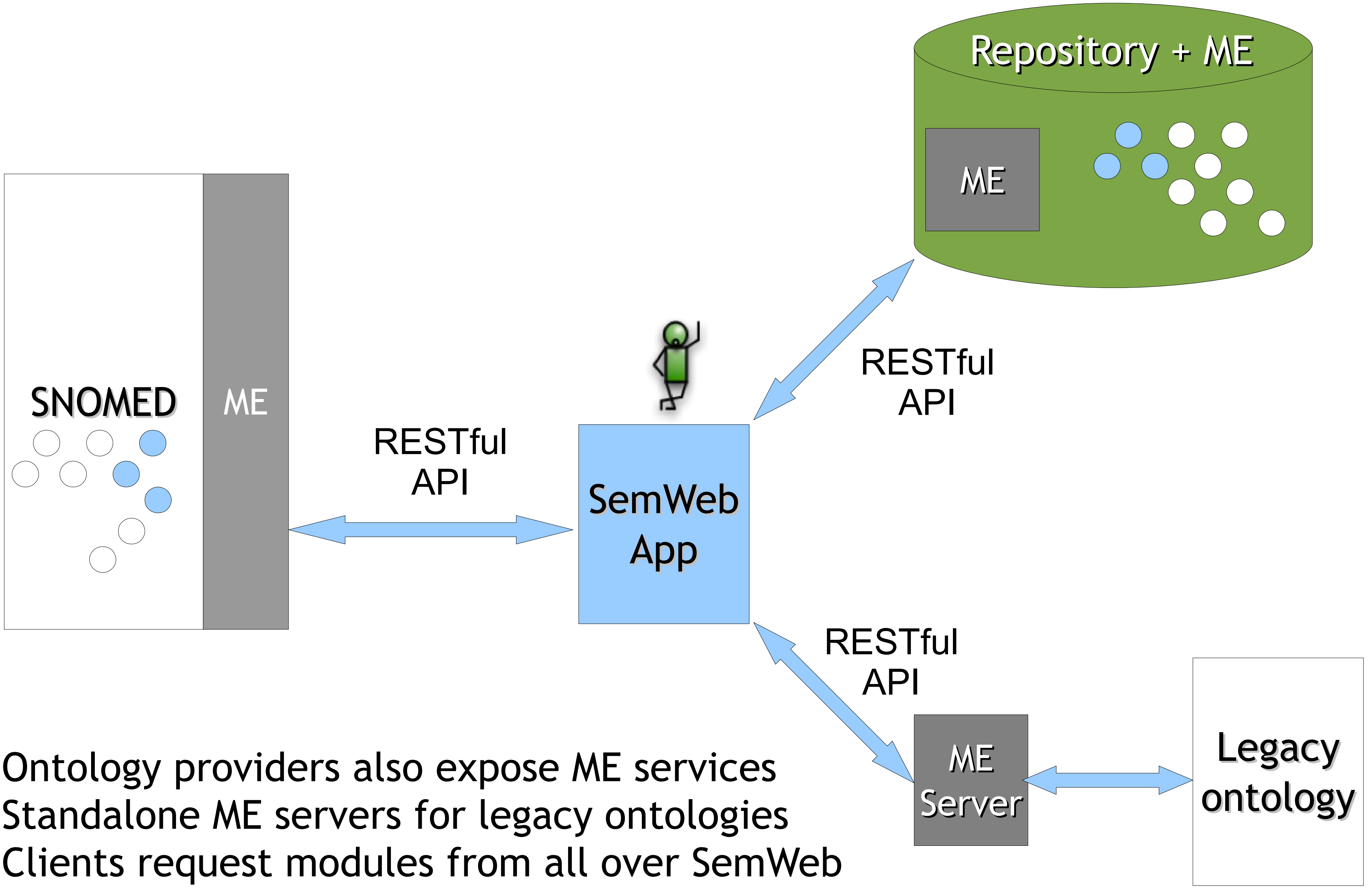
Ontologies are still monolithic
ME services do all the hard job

This talk: decomposition



Separation between logical and physical views

Modular use of ontologies



Ontology providers also expose ME services
Standalone ME servers for legacy ontologies
Clients request modules from all over SemWeb

The University of Manchester

Next:
Modularity and atomic decomposition

Modularity

Module: subset of O that covers terms in signature Σ

Module(Σ,O): a subset of O s.t.

an axiom over Σ follows from O iff

it follows from Module(Σ,O)

Modularity

Module: subset of O that covers terms in signature Σ

Module(Σ,O): a subset of O s.t.

an axiom over Σ follows from O iff

it follows from Module(Σ,O)

Solid logical foundations:

- Goes back to conservative extensions in logic
- Realizable
- Computable in polynomial time
- Implemented in OWL API

But: ontologies have too many (exponentially) modules

Atomic decomposition (AD)

Partitions ontology into atoms:

- atom: set of axioms that isn't split across modules
- dependency structure:

  A ≥ B if any module that contains A also contains B

Atomic decomposition (AD)

Partitions ontology into atoms:

- atom: set of axioms that isn't split across modules
- dependency structure:

  A ≥ B if any module that contains A also contains B

Why is it interesting?

- explicates logical dependencies in O
- succinctly represents all modules
- can be computed efficiently

Let's take an example...

$\alpha_1 = $ '$\texttt{Animal} \sqsubseteq (= 1\texttt{hasGender}.\top)$',

$\alpha_2 = $ '$\texttt{Animal} \sqsubseteq (\geq 1\texttt{hasHabitat}.\top)$',

$\alpha_3 = $ '$\texttt{Person} \sqsubseteq \texttt{Animal}$',

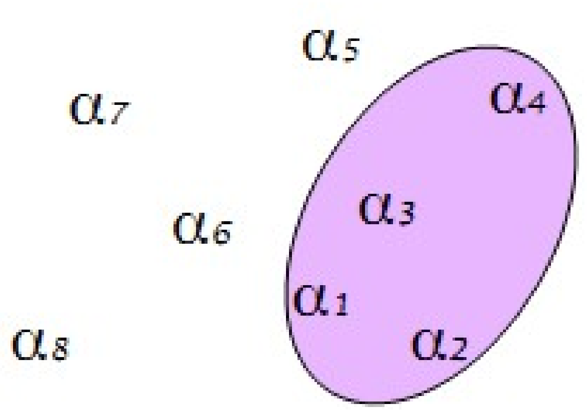
$\alpha_4 = $ '$\texttt{Vegan} \equiv \texttt{Person} \sqcap \forall \texttt{eats}.(\texttt{Vegetable} \sqcup \texttt{Mushroom})$',

$\alpha_5 = $ '$\texttt{TeeTotaller} \equiv \texttt{Person} \sqcap \forall \texttt{drinks}.\texttt{NonAlcoholicThing}$',

$\alpha_6 = $ '$\texttt{Student} \sqsubseteq \texttt{Person} \sqcap \exists \texttt{hasHabitat}.\texttt{University}$',

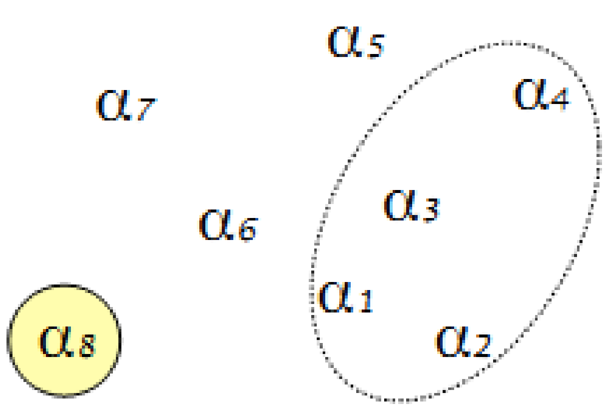
$\alpha_7 = $ '$\texttt{GraduateStudent} \equiv \texttt{Student} \sqcap \exists \texttt{hasDegree}.\{\mathrm{BA}, \mathrm{BS}\}$',

$\alpha_8 = $ '$\texttt{Car} \sqsubseteq \texttt{Vehicle}$'



Example

$$\alpha_1 = \text{`}\mathtt{Animal} \sqsubseteq (= 1\mathtt{hasGender}.\top)\text{'},$$
$$\alpha_2 = \text{`}\mathtt{Animal} \sqsubseteq (\geq 1\mathtt{hasHabitat}.\top)\text{'},$$
$$\alpha_3 = \text{`}\mathtt{Person} \sqsubseteq \mathtt{Animal}\text{'},$$
$$\alpha_4 = \text{`}\mathtt{Vegan} \equiv \mathtt{Person} \sqcap \forall \mathtt{eats}.(\mathtt{Vegetable} \sqcup \mathtt{Mushroom})\text{'},$$
$$\alpha_5 = \text{`}\mathtt{TeeTotaller} \equiv \mathtt{Person} \sqcap \forall \mathtt{drinks}.\mathtt{NonAlcoholicThing}\text{'},$$
$$\alpha_6 = \text{`}\mathtt{Student} \sqsubseteq \mathtt{Person} \sqcap \exists \mathtt{hasHabitat}.\mathtt{University}\text{'},$$
$$\alpha_7 = \text{`}\mathtt{GraduateStudent} \equiv \mathtt{Student} \sqcap \exists \mathtt{hasDegree}.\{\mathrm{BA}, \mathrm{BS}\}\text{'},$$
$$\alpha_8 = \text{`}\mathtt{Car} \sqsubseteq \mathtt{Vehicle}\text{'}$$



Example

α_1 = 'Animal $\sqsubseteq$ (= 1hasGender.$\top$)',

α_2 = 'Animal $\sqsubseteq$ ($\geq$ 1hasHabitat.$\top$)',

α_3 = 'Person $\sqsubseteq$ Animal',

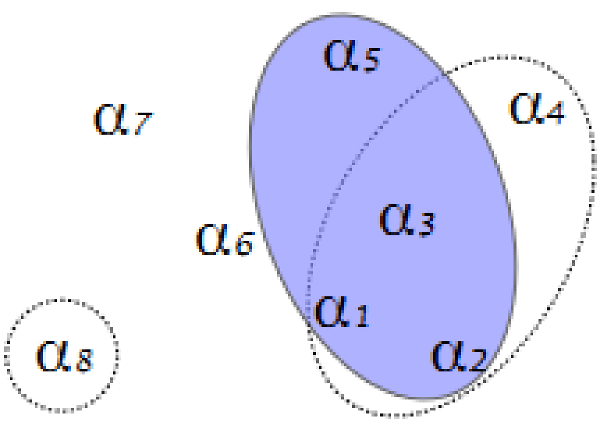
α_4 = 'Vegan $\equiv$ Person $\sqcap$ $\forall$eats.(Vegetable $\sqcup$ Mushroom)',

α_5 = 'TeeTotaller $\equiv$ Person $\sqcap$ $\forall$drinks.NonAlcoholicThing',

α_6 = 'Student $\sqsubseteq$ Person $\sqcap$ $\exists$hasHabitat.University',

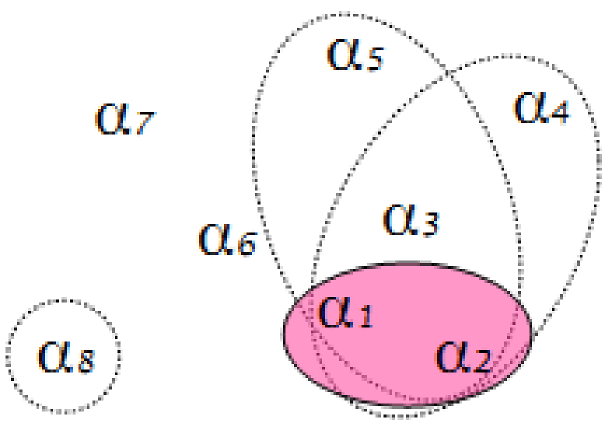
α_7 = 'GraduateStudent $\equiv$ Student $\sqcap$ $\exists$hasDegree.{BA, BS}',

α_8 = 'Car $\sqsubseteq$ Vehicle'



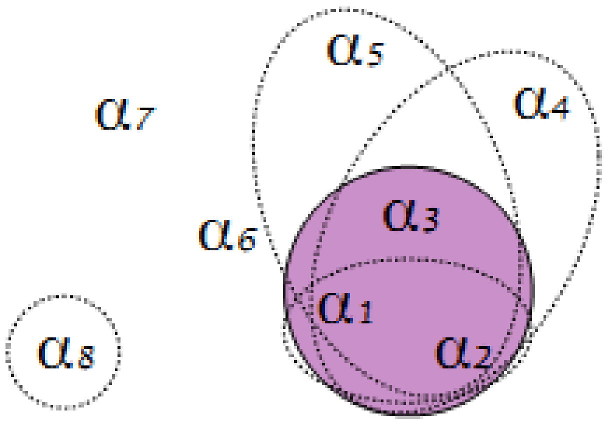
Example

α_1 = 'Animal $\sqsubseteq$ (= 1hasGender.$\top$)',
α_2 = 'Animal $\sqsubseteq$ ($\geq$ 1hasHabitat.$\top$)',
α_3 = 'Person $\sqsubseteq$ Animal',
α_4 = 'Vegan $\equiv$ Person $\sqcap$ $\forall$eats.(Vegetable $\sqcup$ Mushroom)',
α_5 = 'TeeTotaller $\equiv$ Person $\sqcap$ $\forall$drinks.NonAlcoholicThing',
α_6 = 'Student $\sqsubseteq$ Person $\sqcap$ $\exists$hasHabitat.University',
α_7 = 'GraduateStudent $\equiv$ Student $\sqcap$ $\exists$hasDegree.{BA, BS}',
α_8 = 'Car $\sqsubseteq$ Vehicle'



Example

$\alpha_1 =$ 'Animal $\sqsubseteq$ $(= 1 \texttt{hasGender}.\top)$',
$\alpha_2 =$ 'Animal $\sqsubseteq$ $(\geq 1 \texttt{hasHabitat}.\top)$',
$\alpha_3 =$ 'Person $\sqsubseteq$ Animal',
$\alpha_4 =$ 'Vegan $\equiv$ Person $\sqcap \forall$eats.(Vegetable $\sqcup$ Mushroom)',
$\alpha_5 =$ 'TeeTotaller $\equiv$ Person $\sqcap \forall$drinks.NonAlcoholicThing',
$\alpha_6 =$ 'Student $\sqsubseteq$ Person $\sqcap \exists$hasHabitat.University',
$\alpha_7 =$ 'GraduateStudent $\equiv$ Student $\sqcap \exists$hasDegree.$\{BA, BS\}$',
$\alpha_8 =$ 'Car $\sqsubseteq$ Vehicle'



MANCHESTER 1824

The University of Manchester

BIO5

Example

α_1 = 'Animal $\sqsubseteq$ (= 1hasGender.$\top$)',

α_2 = 'Animal $\sqsubseteq$ ($\geq$ 1hasHabitat.$\top$)',

α_3 = 'Person $\sqsubseteq$ Animal',

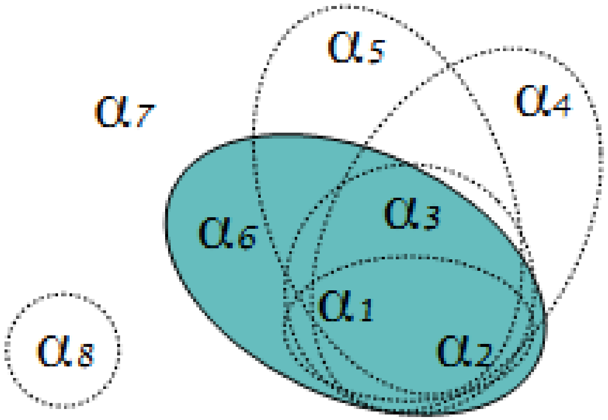
α_4 = 'Vegan $\equiv$ Person $\sqcap$ $\forall$eats.(Vegetable $\sqcup$ Mushroom)',

α_5 = 'TeeTotaller $\equiv$ Person $\sqcap$ $\forall$drinks.NonAlcoholicThing',

α_6 = 'Student $\sqsubseteq$ Person $\sqcap$ $\exists$hasHabitat.University',

α_7 = 'GraduateStudent $\equiv$ Student $\sqcap$ $\exists$hasDegree.{BA, BS}',

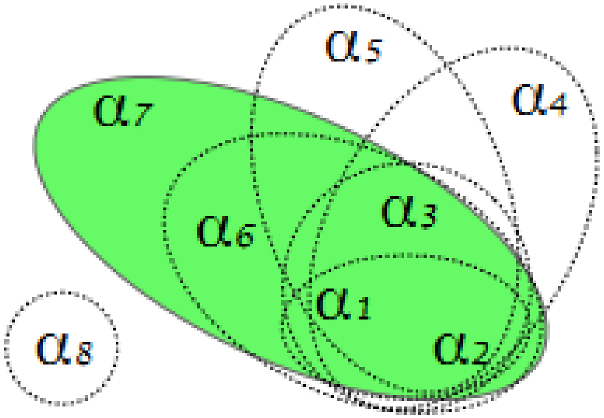
α_8 = 'Car $\sqsubseteq$ Vehicle'



The University of Manchester

Example

α_1 = 'Animal $\sqsubseteq$ (= 1hasGender.$\top$)',
α_2 = 'Animal $\sqsubseteq$ ($\geq$ 1hasHabitat.$\top$)',
α_3 = 'Person $\sqsubseteq$ Animal',
α_4 = 'Vegan $\equiv$ Person $\sqcap$ $\forall$eats.(Vegetable $\sqcup$ Mushroom)',
α_5 = 'TeeTotaller $\equiv$ Person $\sqcap$ $\forall$drinks.NonAlcoholicThing',
α_6 = 'Student $\sqsubseteq$ Person $\sqcap$ $\exists$hasHabitat.University',
α_7 = 'GraduateStudent $\equiv$ Student $\sqcap$ $\exists$hasDegree.{BA, BS}',
α_8 = 'Car $\sqsubseteq$ Vehicle'



Example

$\alpha_1 = $ 'Animal $\sqsubseteq (= 1\text{hasGender}.\top)$',

$\alpha_2 = $ 'Animal $\sqsubseteq (\geq 1\text{hasHabitat}.\top)$',

$\alpha_3 = $ 'Person $\sqsubseteq$ Animal',

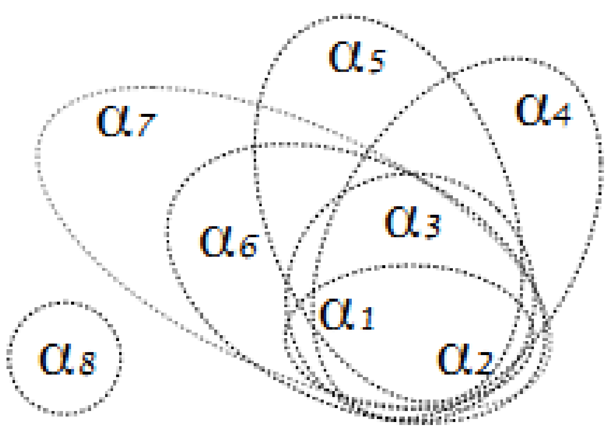
$\alpha_4 = $ 'Vegan $\equiv$ Person $\sqcap \forall$eats.(Vegetable $\sqcup$ Mushroom)',

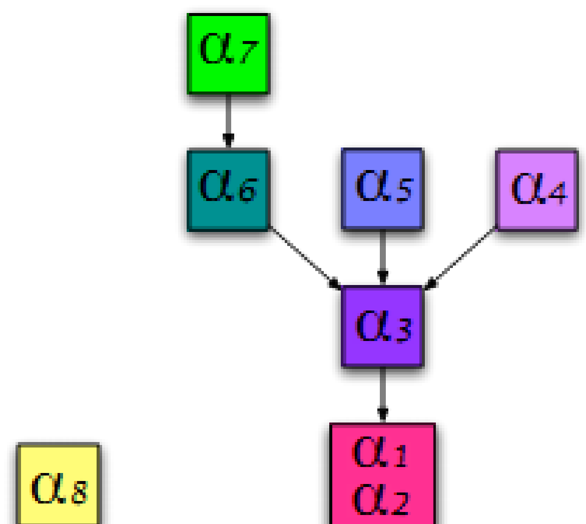
$\alpha_5 = $ 'TeeTotaller $\equiv$ Person $\sqcap \forall$drinks.NonAlcoholicThing',

$\alpha_6 = $ 'Student $\sqsubseteq$ Person $\sqcap \exists$hasHabitat.University',

$\alpha_7 = $ 'GraduateStudent $\equiv$ Student $\sqcap \exists$hasDegree.{BA, BS}',

$\alpha_8 = $ 'Car $\sqsubseteq$ Vehicle'





The University of Manchester

Next:
Fast module extraction and evaluation on BioPortal ontologies

Evaluation

Dataset: NCBO BioPortal (state-of-the-art, public API)

Goals

- how decomposable are the bio ontologies?
- how effective is syntactic modularity?
- is AD feasible in practice?
- is AD-based module extraction beneficial?

The University of Manchester

Evaluation

181 OWL and OBO ontologies

Decomposed and stored in XML DB in ~3 hrs

Average atom size is only 2.19 axioms

Most ontologies have no atoms larger than 10 axioms

The University of Manchester

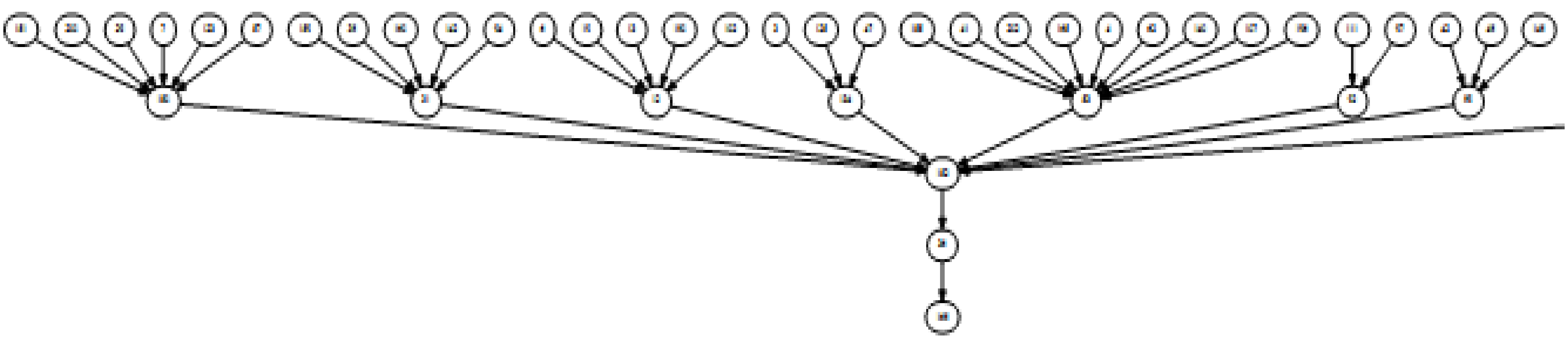
Evaluation

181 OWL and OBO ontologies

Decomposed and stored in XML DB in ~3 hrs

Average atom size is only 2.19 axioms

Most ontologies have no atoms larger than 10 axioms

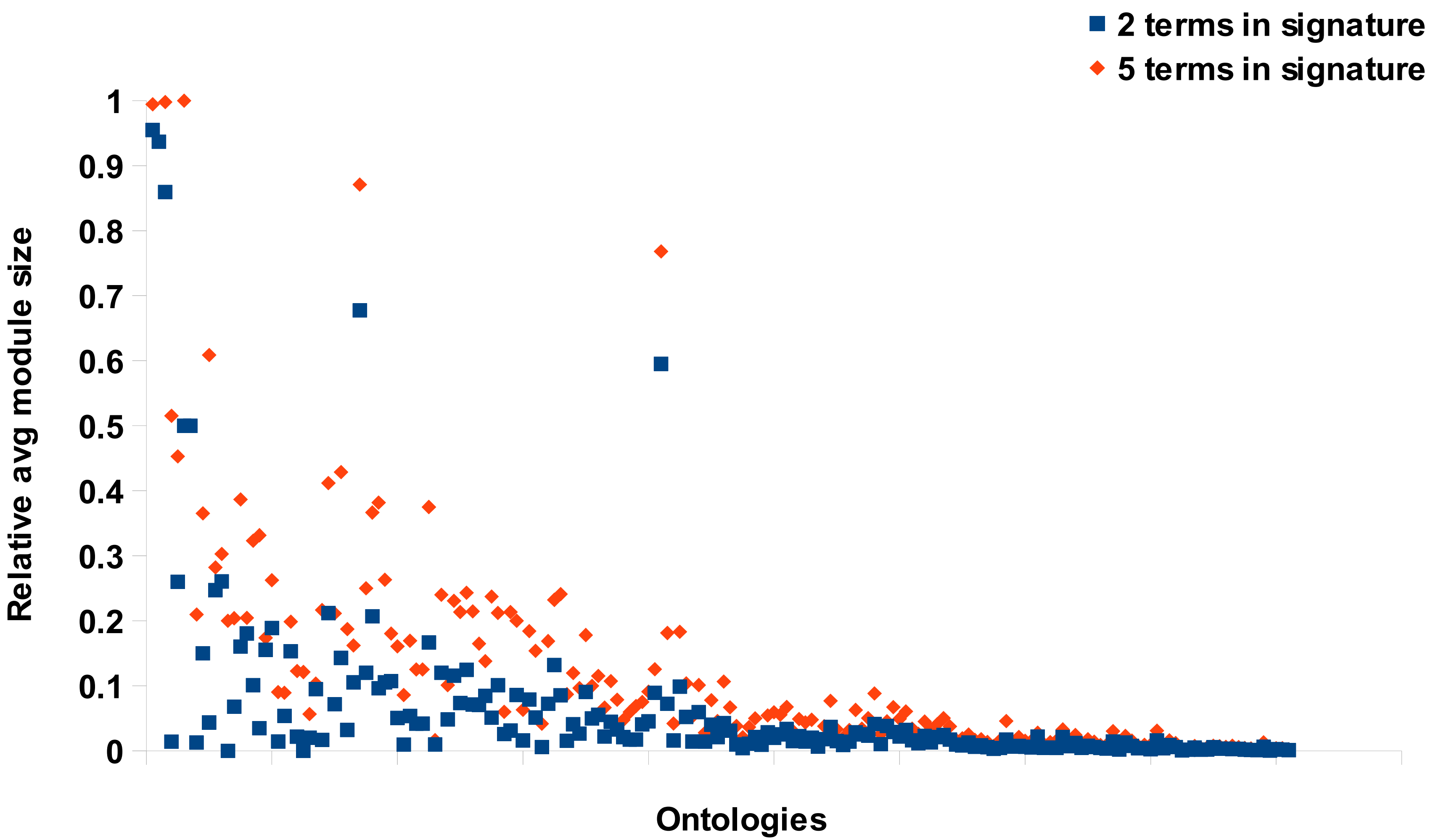


Flat, loosely connected AD graphs → modules should be small

The University of Manchester

Random modules evaluation

Modules are pretty small on average
median relative size: 2% - 6%
usually <50 axioms



Atoms and labels

To be useable, atoms are labeled with their terms

Atoms and labels

To be useable, atoms are labeled with their terms

Labels

- depend on task at hand
- ME task: which atoms compose $M(\Sigma,O)$?

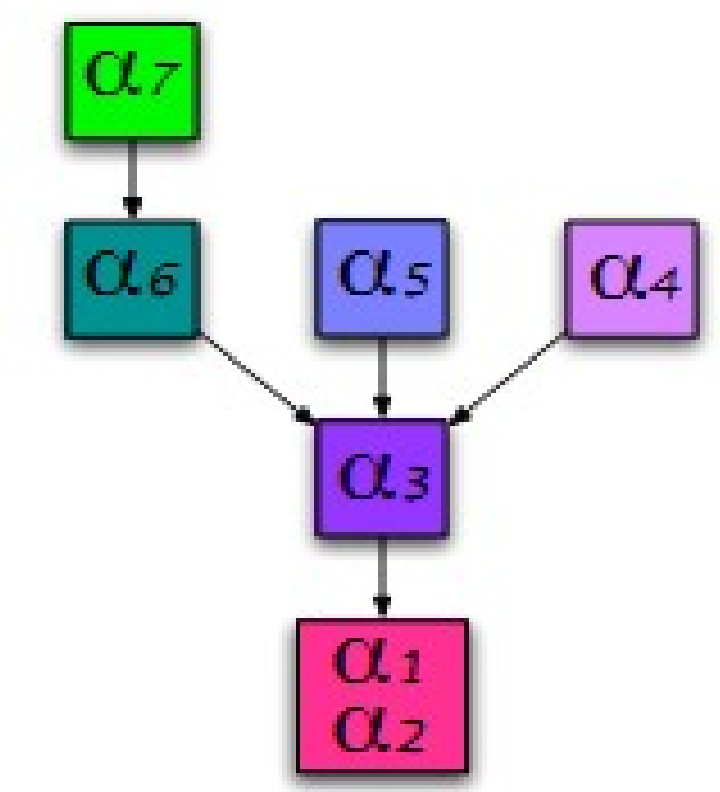
Min seed signatures (MSS) as labels

$MSS(A) = \{\text{minimal } \Sigma \mid M(\Sigma,O) \text{ contains } A\}$

MSS labels and FME

MSS(α7)={{GraduateStudent},

{Student, hasDegree}}



The University of Manchester

MSS labels and AD-based ME

Non-trivial to compute (fine for 176/181 ontologies)

Fast Module Extraction (FME) algorithm:

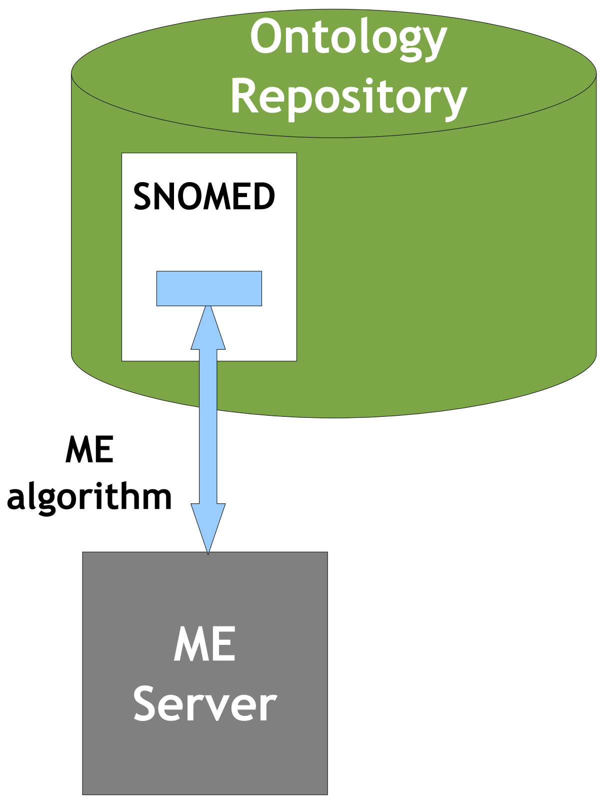
1) pick relevant atoms for Σ and their dependencies (based on labels)
2) expand Σ
3) go to 1)

No need to look at every axiom in the ontology

AD-based ME is faster

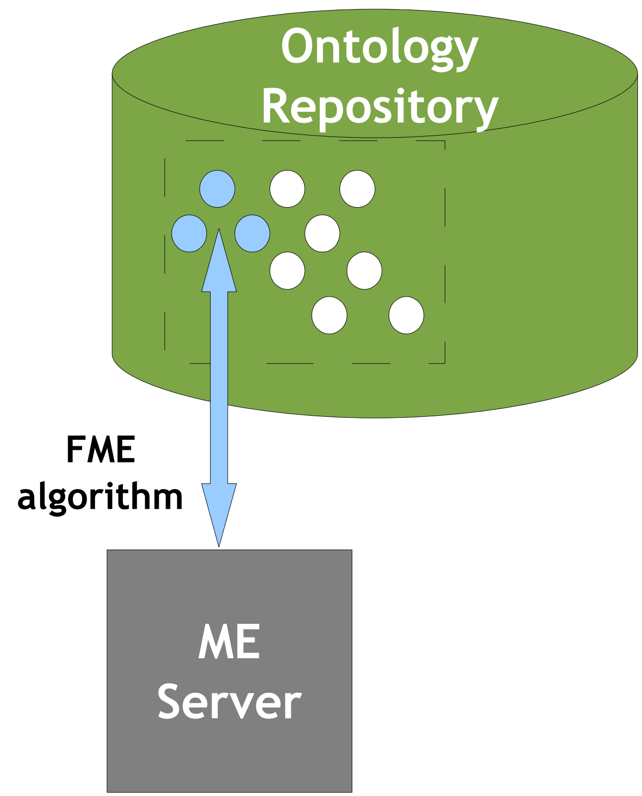
(on real ontologies)

Classical ME
(examines each axiom)



vs

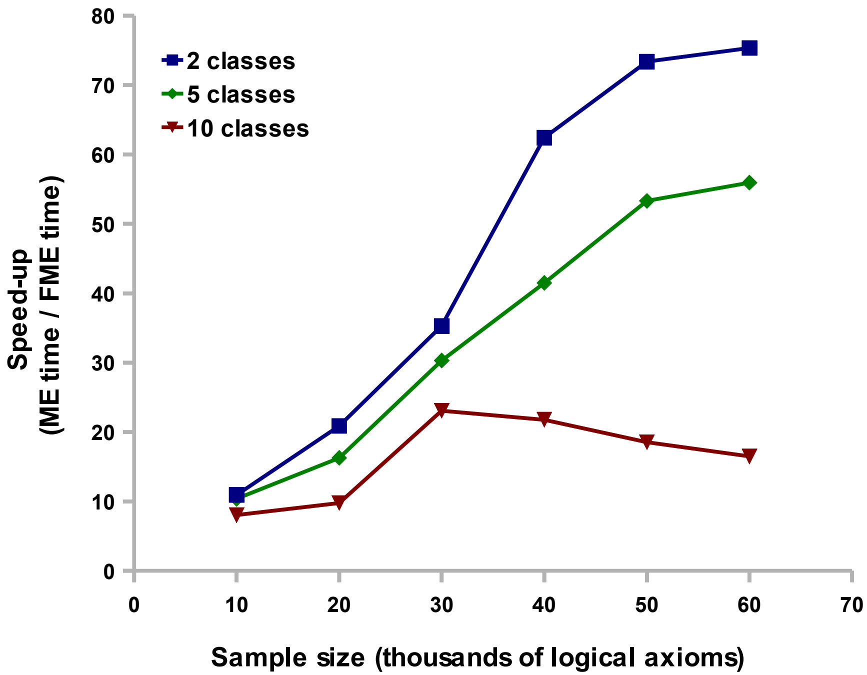
Fast AD-based ME
(picks relevant atoms)



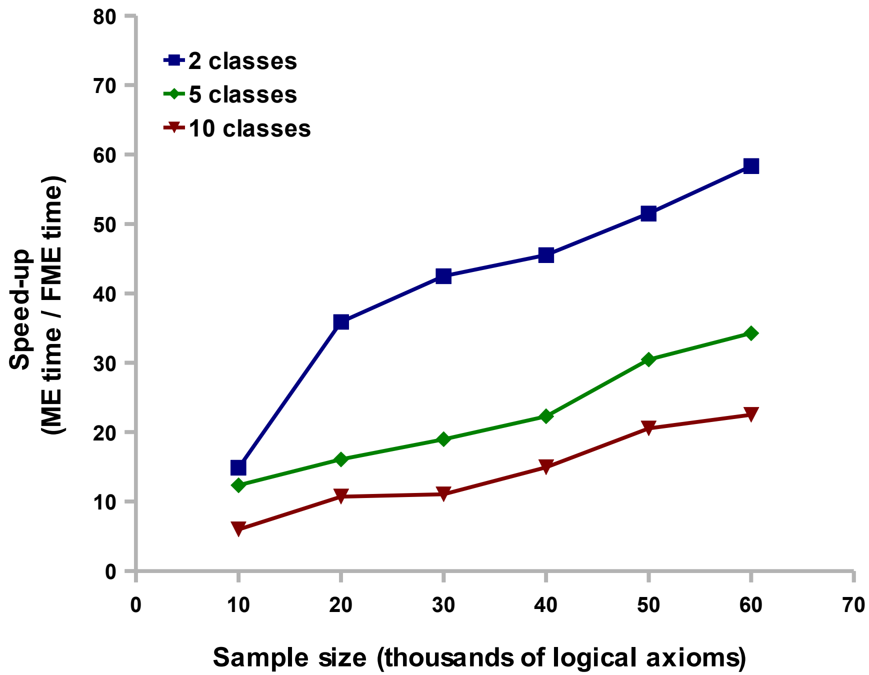
AD-based ME is faster

(on real ontologies)

Samples of ChEBI



Samples of GO



This doesn't even account for loading speed-up

Atomic decomposition: summary

AD is a promising decomposition technique

Machine oriented tasks:

- module extraction (FME)
- incremental reasoning?

Human-oriented tasks:

- comprehension and analysis (DeMoSt)
- *tell me everything about C*
- collaborative development

Conclusion

Step towards on-demand, transaction-time reasoning for Semantic Web interoperability with logical guarantees

Conclusion

Step towards on-demand, transaction-time reasoning for Semantic Web interoperability with logical guarantees

State-of-the-art ontologies support that

- axiomatically weak, sparse knowledge, loosely linked terms
- small modules for most signatures/ontologies

Work in progress and future plans

- use it for SSWAP
- incremental updates

Acknowledgements

This material is based upon work supported by the National Science Foundation (NSF) under grant #0943879 and the NSF iPlant Cyberinfrastructure Program (#EF-0735191)

For SSWAP visit http://sswap.info, http://sswap.iplantcollaborative.org

Questions?