Resolving and avoiding design conflicts in ontology development and deployment

Maria Keet

Department of Computer Science, University of Cape Town, South Africa
mkeet@cs.uct.ac.za

ISCBacademy Webinar series (bio-ontologies), 21 September 2021

Outline

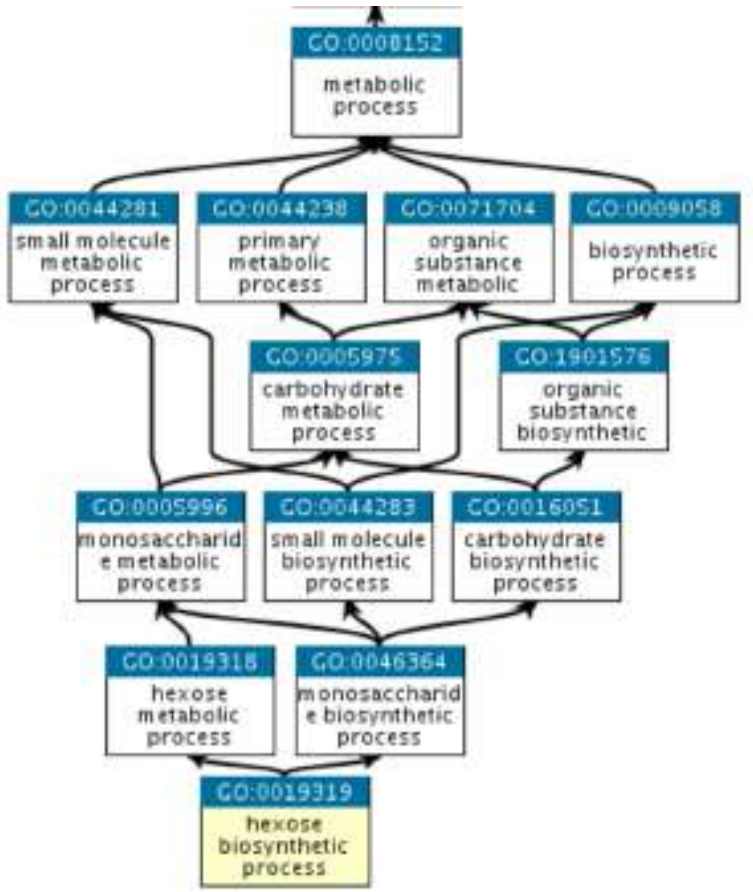
1. Context and motivation
2. Resolving conflicts
3. Implementation trade-offs
4. Conclusions

Outline

1. Context and motivation
2. Resolving conflicts
3. Implementation trade-offs
4. Conclusions

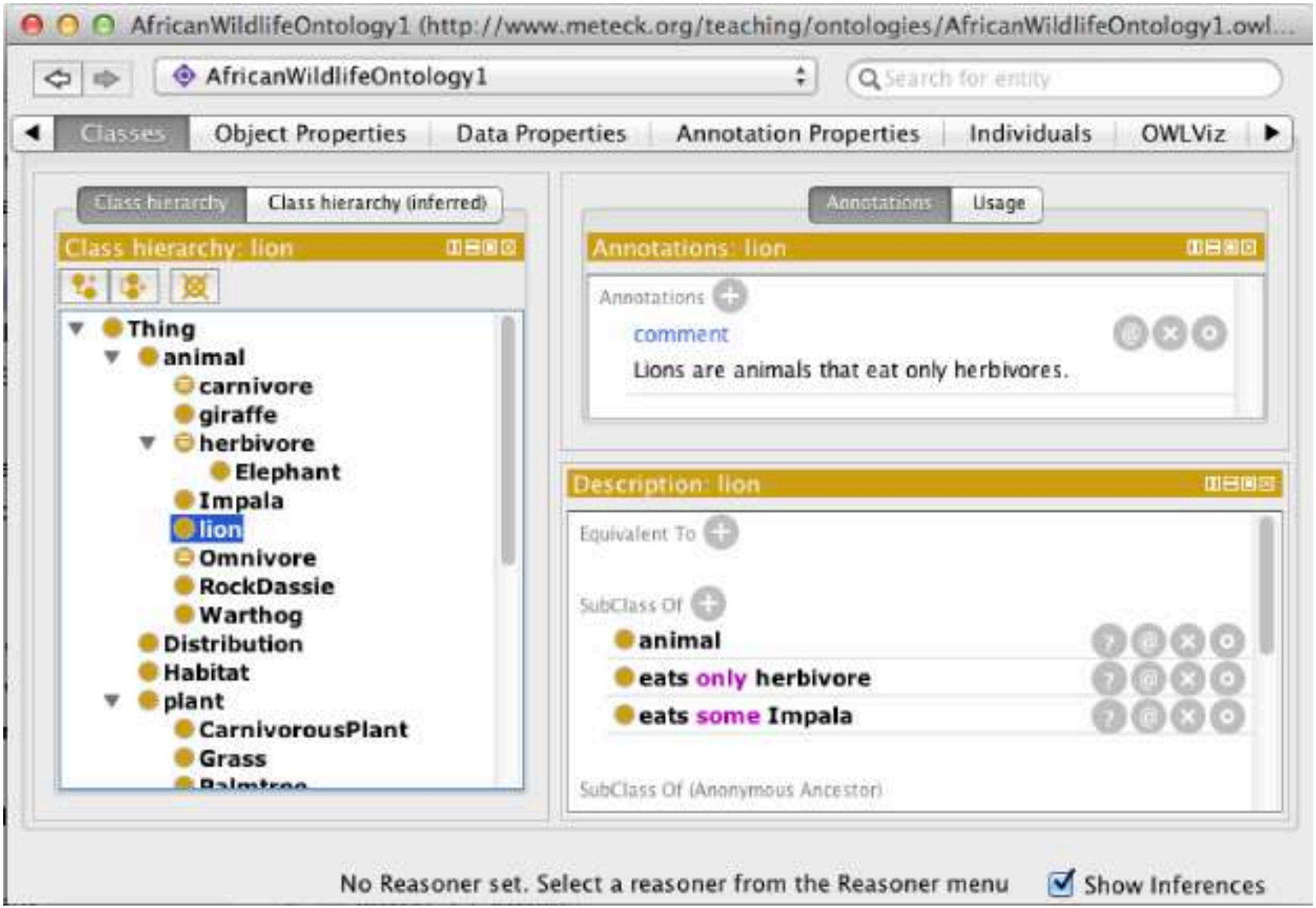
An ontology

Simplified graphical rendering of a fragment of one:

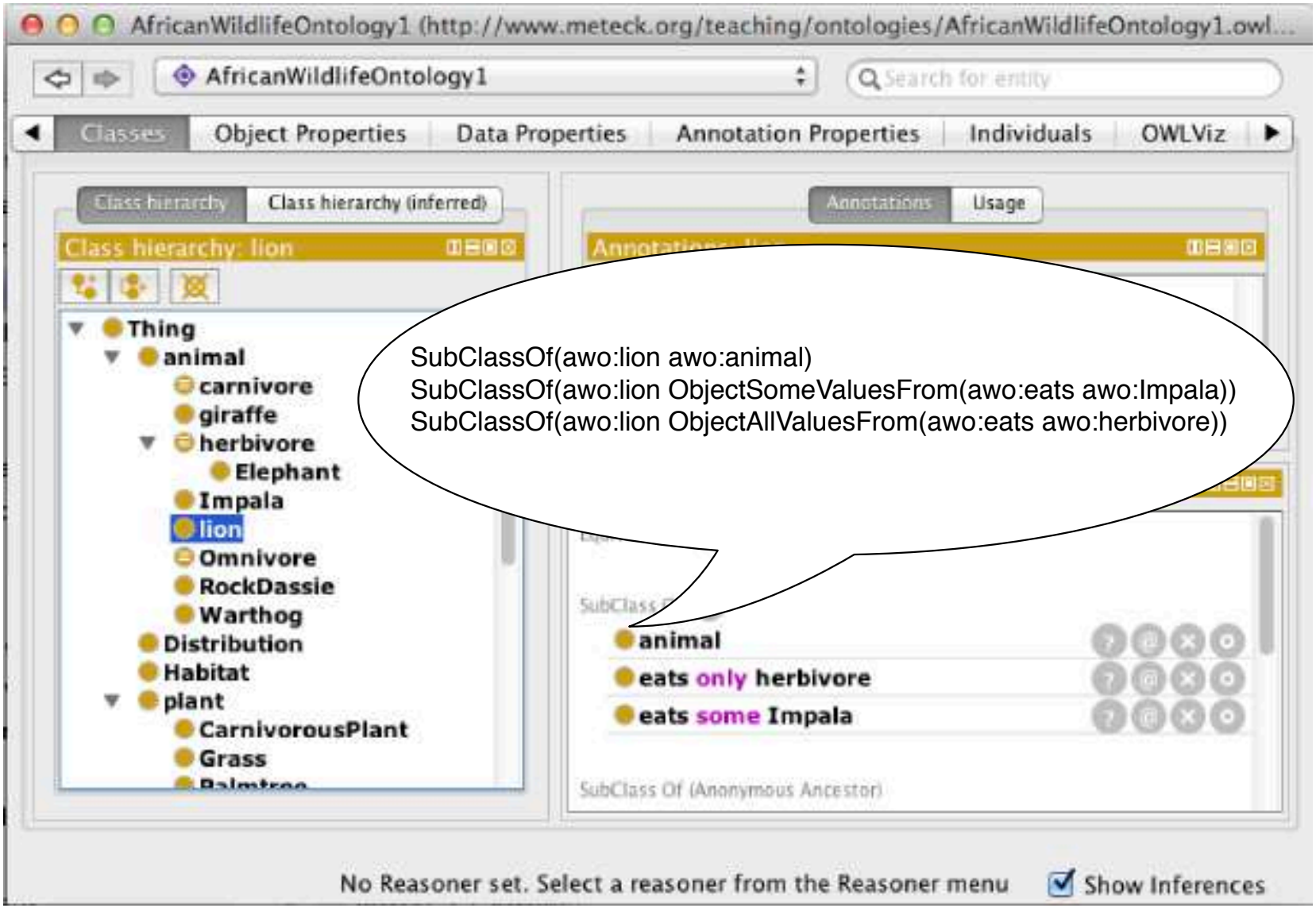


http://geneontology.org/docs/ontology-documentation/

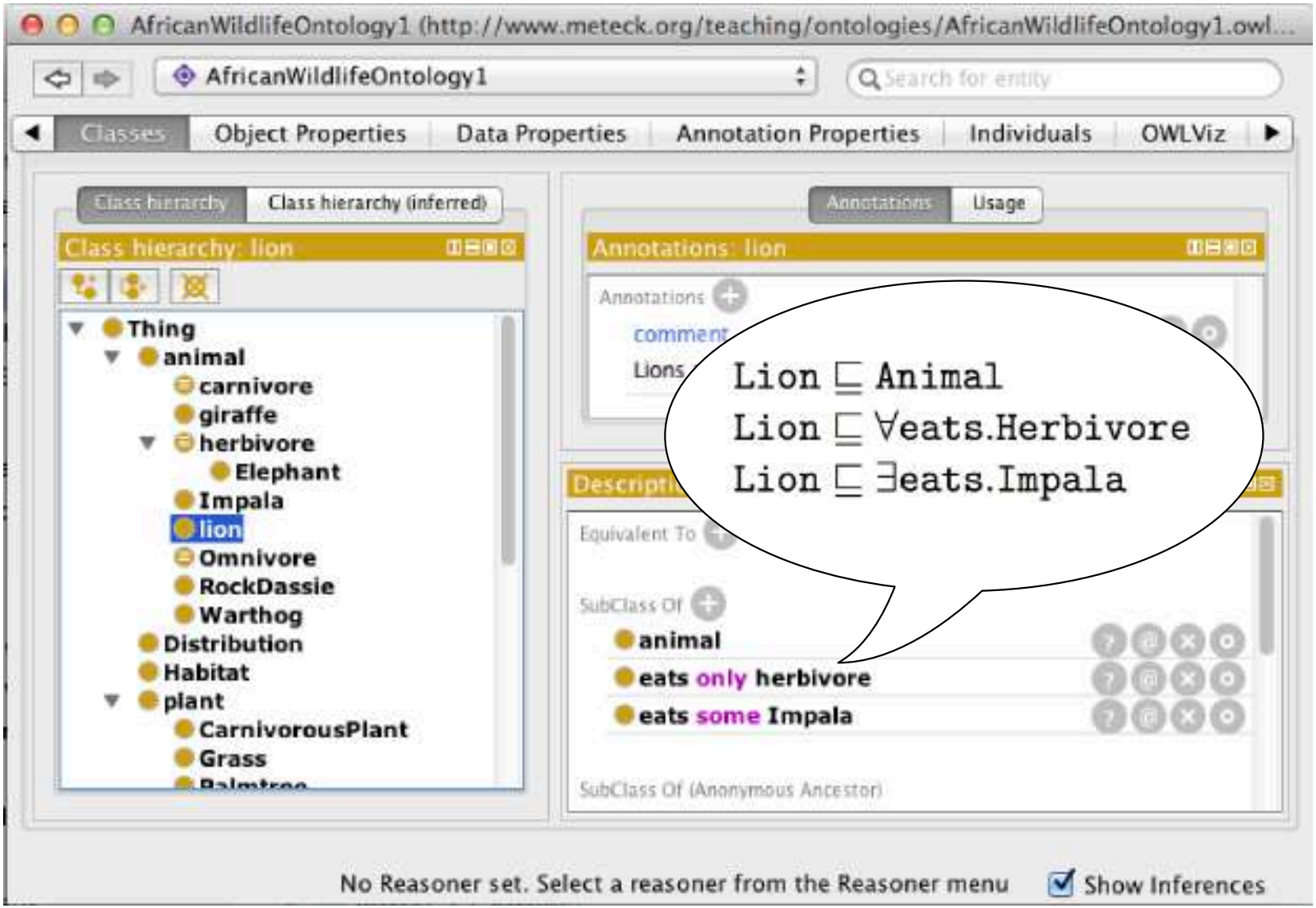
In an ODE...


AfricanWildlifeOntology1 (http://www.meteck.org/teaching/ontologies/AfricanWildlifeOntology1.owl...
AfricanWildlifeOntology1
Search for entity
Classes
Object Properties
Data Properties
Annotation Properties
Individuals
OWLViz
Class hierarchy
Class hierarchy (inferred)
Class hierarchy: lion
Thing
animal
carnivore
giraffe
herbivore
Elephant
Impala
lion
Omnivore
RockDassie
Warthog
Distribution
Habitat
plant
CarnivorousPlant
Grass
Annotations
Usage
Annotations: lion
Annotations
comment
Lions are animals that eat only herbivores.
Description: lion
Equivalent To
SubClass Of
animal
eats only herbivore
eats some Impala
SubClass Of (Anonymous Ancestor)
No Reasoner set. Select a reasoner from the Reasoner menu
Show Inferences

... happenings behind the GUI ...



... and underlying that serialisation



Introduction

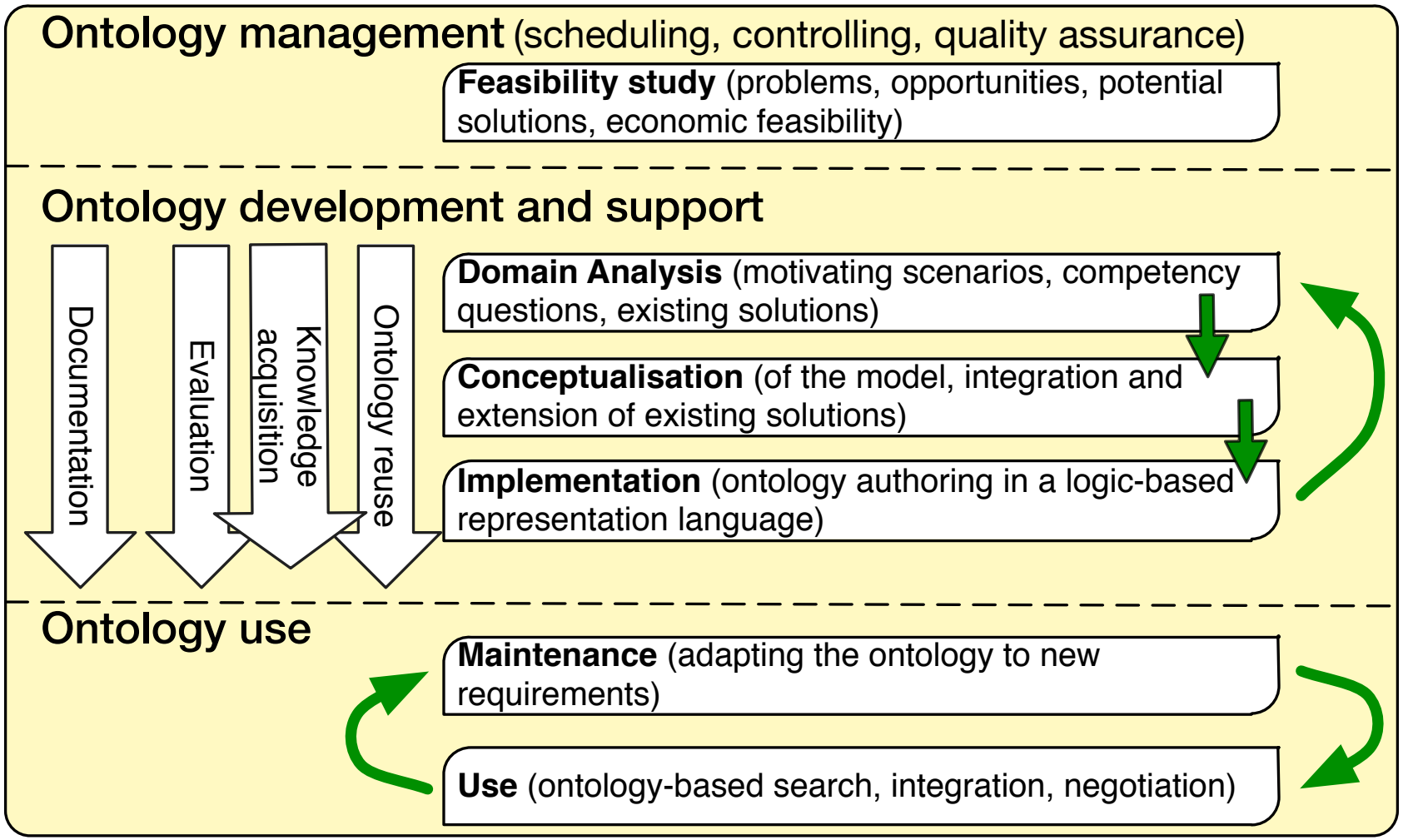
- Ontologies
  - For their own sake
  - For communication among humans
  - Used for many different ontology-driven information systems (database integration and linking, recommender systems, NLP, textbook annotation and search, question generation, Q&A systems, etc.)

Introduction

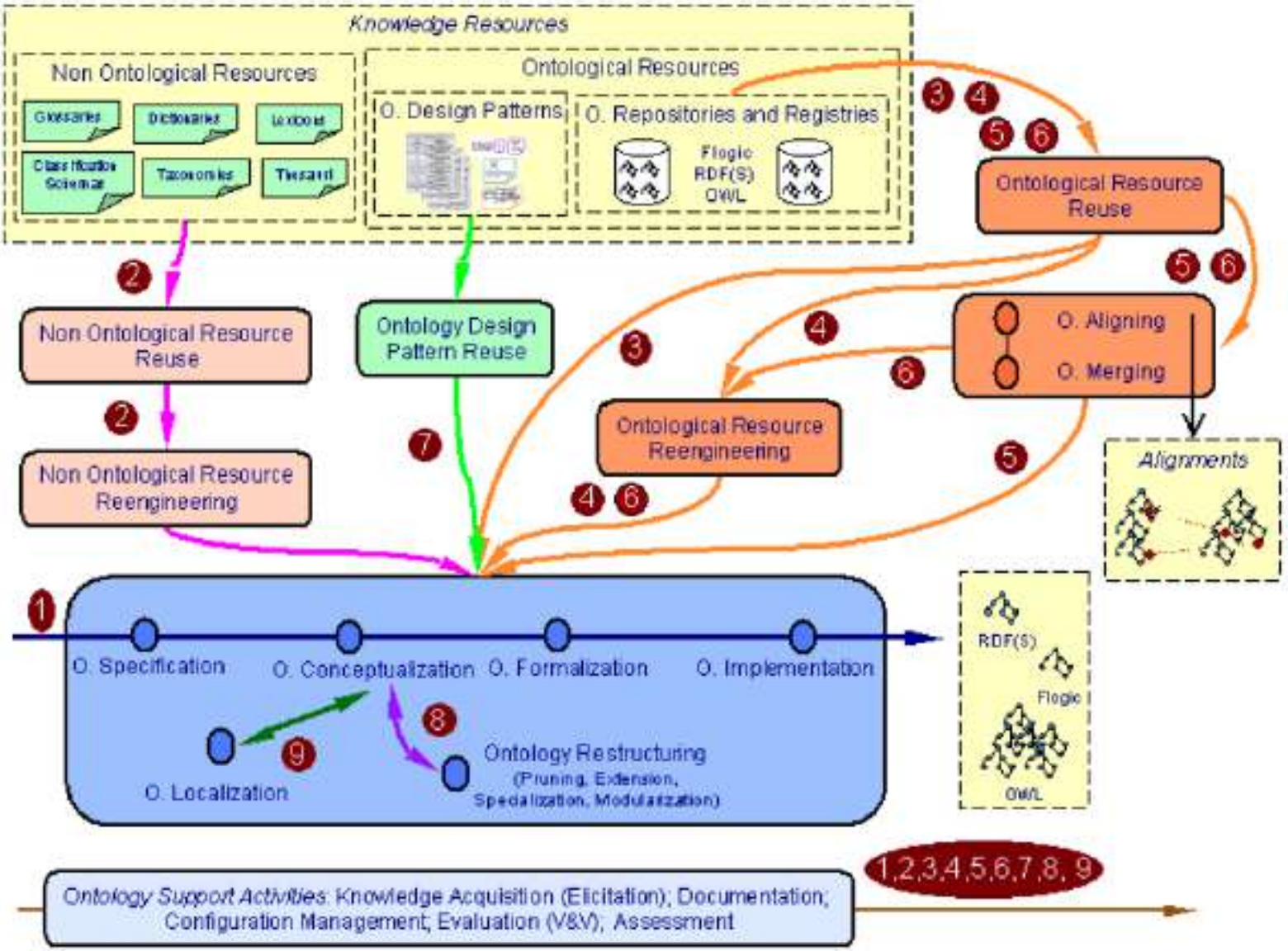
- Ontologies
  - For their own sake
  - For communication among humans
  - Used for many different ontology-driven information systems (database integration and linking, recommender systems, NLP, textbook annotation and search, question generation, Q&A systems, etc.)

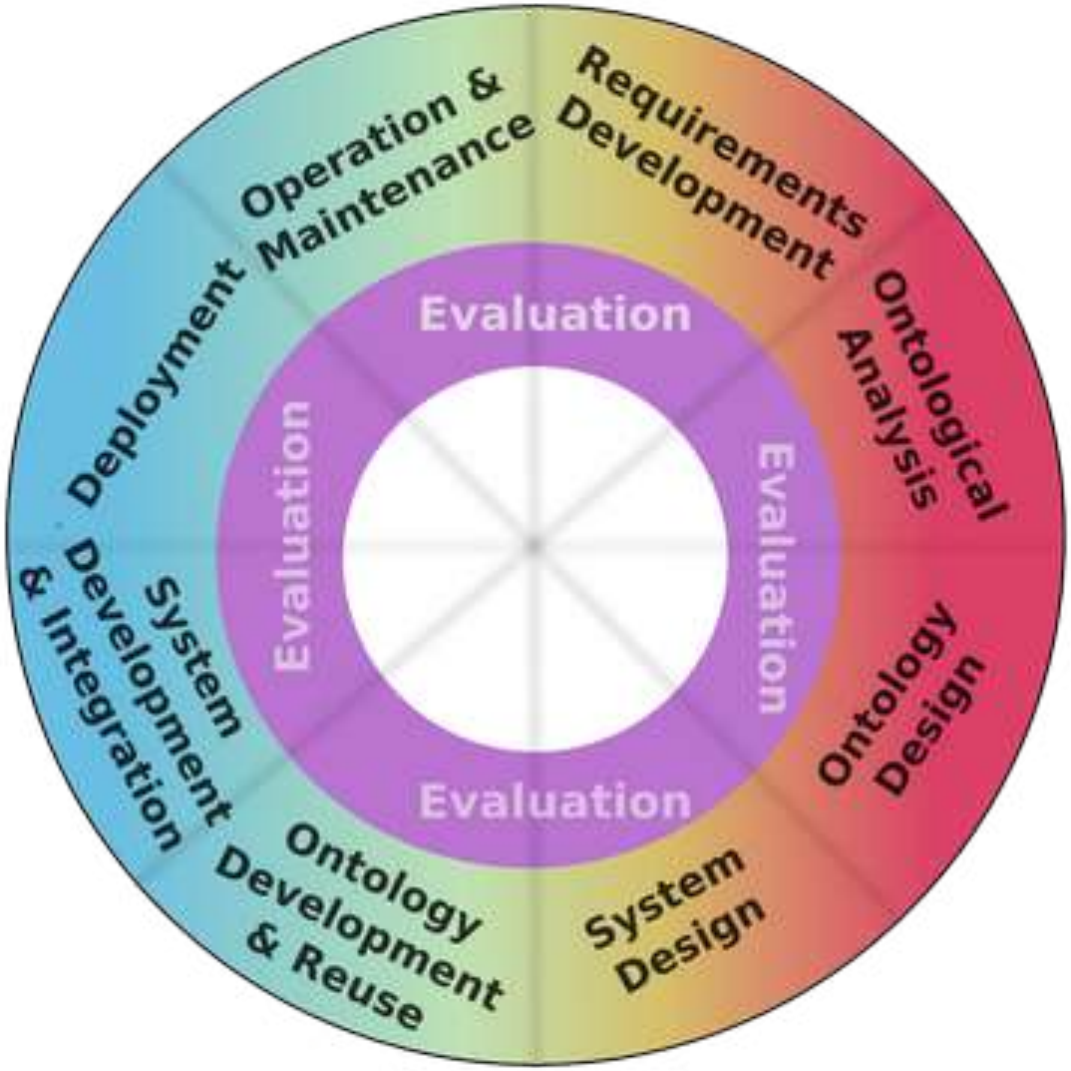
$\Rightarrow$ Someone has to build them, *somehow*

Typical stages of macro-level methodologies



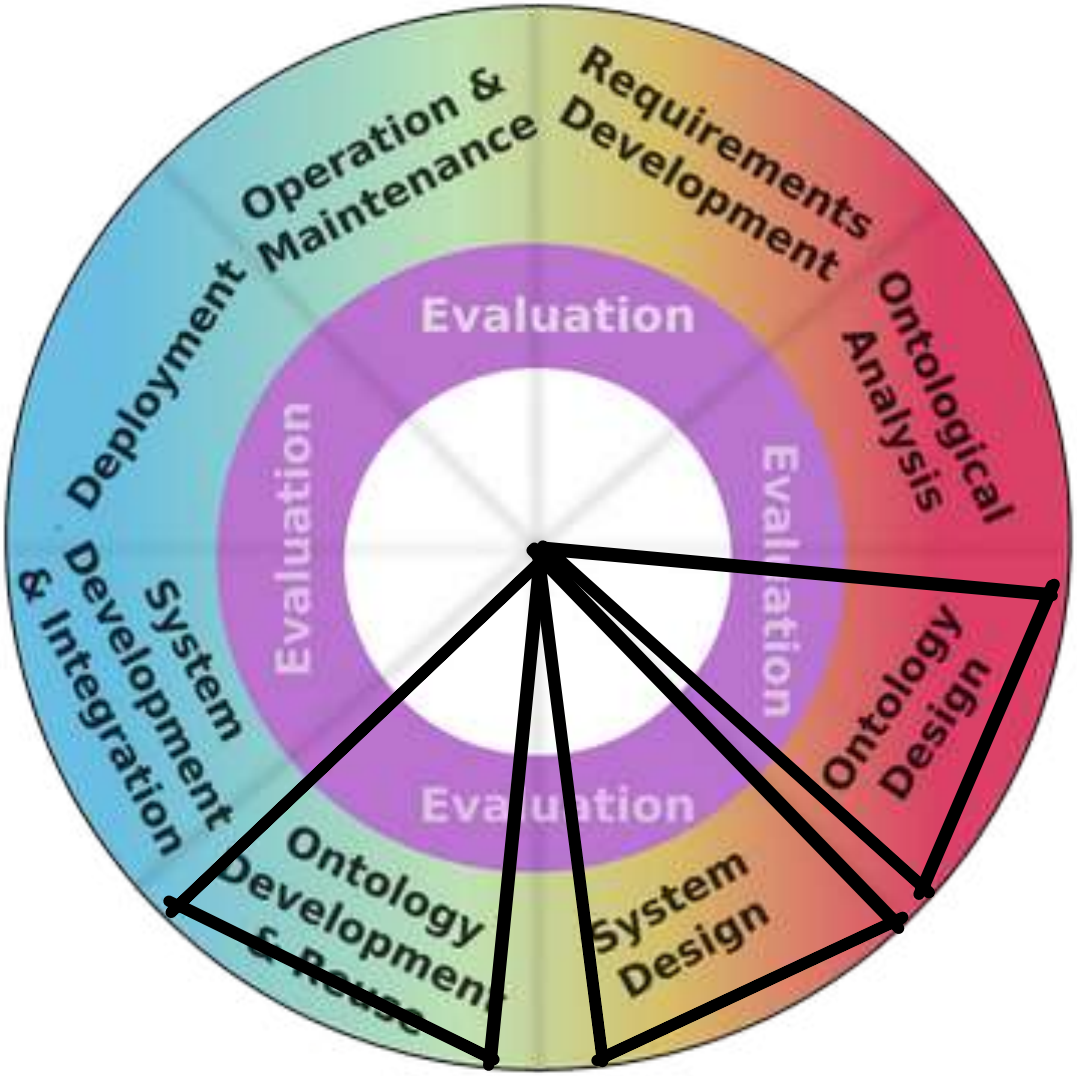
Scenarios for building Ontology Networks (NEON methodology)





Ontology Summit 2013's lifecycle model

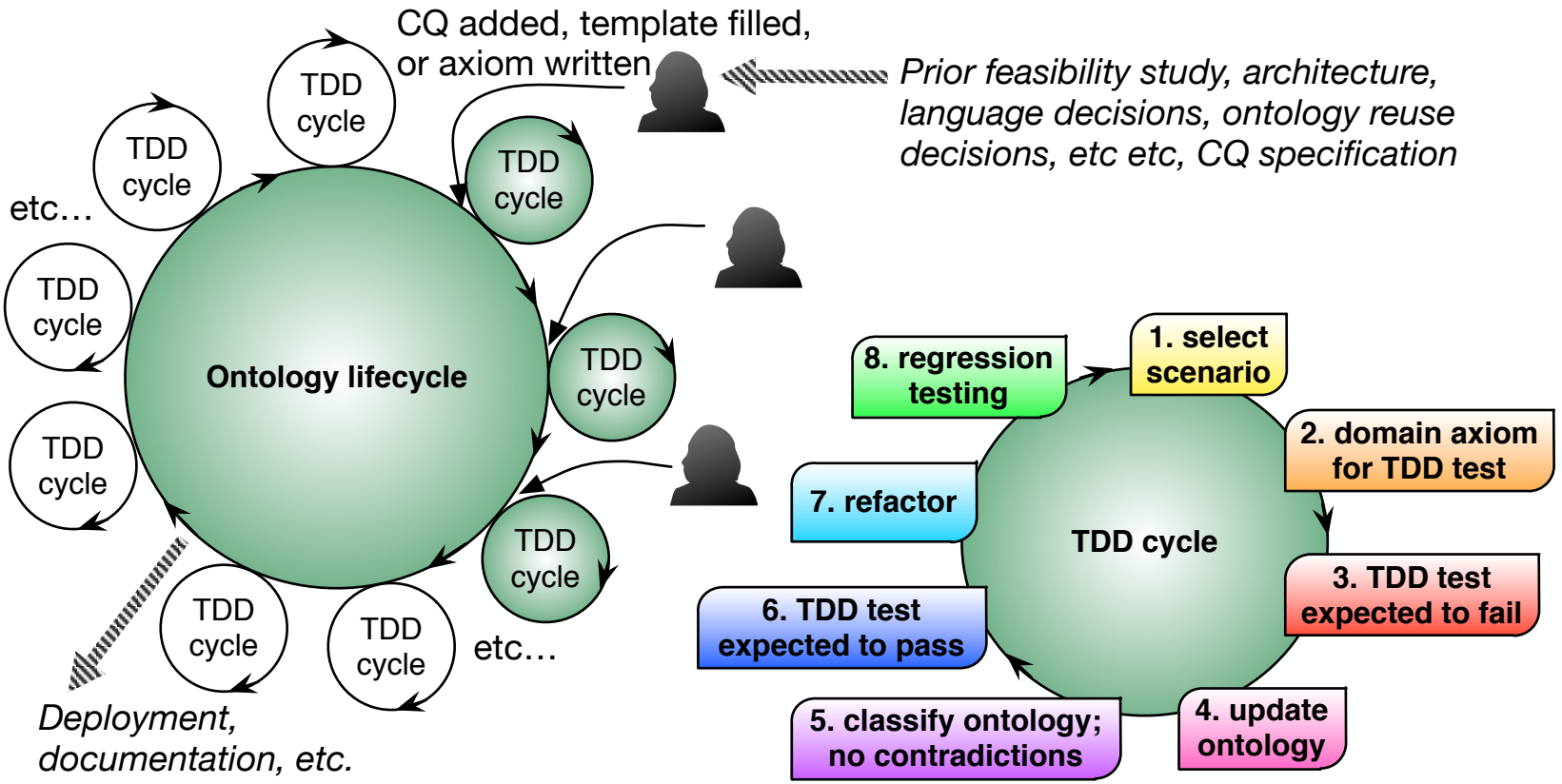
`http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Communique`



Ontology Summit 2013's lifecycle model

http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Communique

More cycles within a cycle (for "ontology design")



Keet CM, Ławrynowicz A. Test-Driven Development of Ontologies. ESWC'16.
Davies K., Keet CM, Ławrynowicz A. More Effective Ontology Authoring with Test-Driven Development and the TDDonto2 tool. IJAIT, 2019, 28(7): 1950023.

Ontology development at the 'micro-level' level (cf. macro)

- We need to get those axioms into the ontology

Ontology development at the 'micro-level' level (cf. macro)

- We need to get those axioms into the ontology
- The actual modelling, or *ontology authoring*, using micro-level guidelines, methods, and tools
  - Methods, such as reverse engineering and text mining to start, OntoClean and OntoPartS to improve an ontology's quality
  - Tools to model, to reason, to debug, to integrate, to link to data

Ontology development at the 'micro-level' level (cf. macro)

- We need to get those axioms into the ontology
- The actual modelling, or *ontology authoring*, using micro-level guidelines, methods, and tools
  - Methods, such as reverse engineering and text mining to start, OntoClean and OntoPartS to improve an ontology's quality
  - Tools to model, to reason, to debug, to integrate, to link to data

⇒ But what if you're not sure of the axioms yet? Or it leads to a conflict and possibly also an 'incoherent' ontology?

Examples

- BFO does not have `Stuff` (e.g., mucus, cytosol, water). Deny its existence? Add it as a not quite fitting subclass somewhere? Create/reuse a core ontology?
- `Virus` $\sqsubseteq$ `Organism` vs. `Virus` $\sqsubseteq$ `acellular structure`

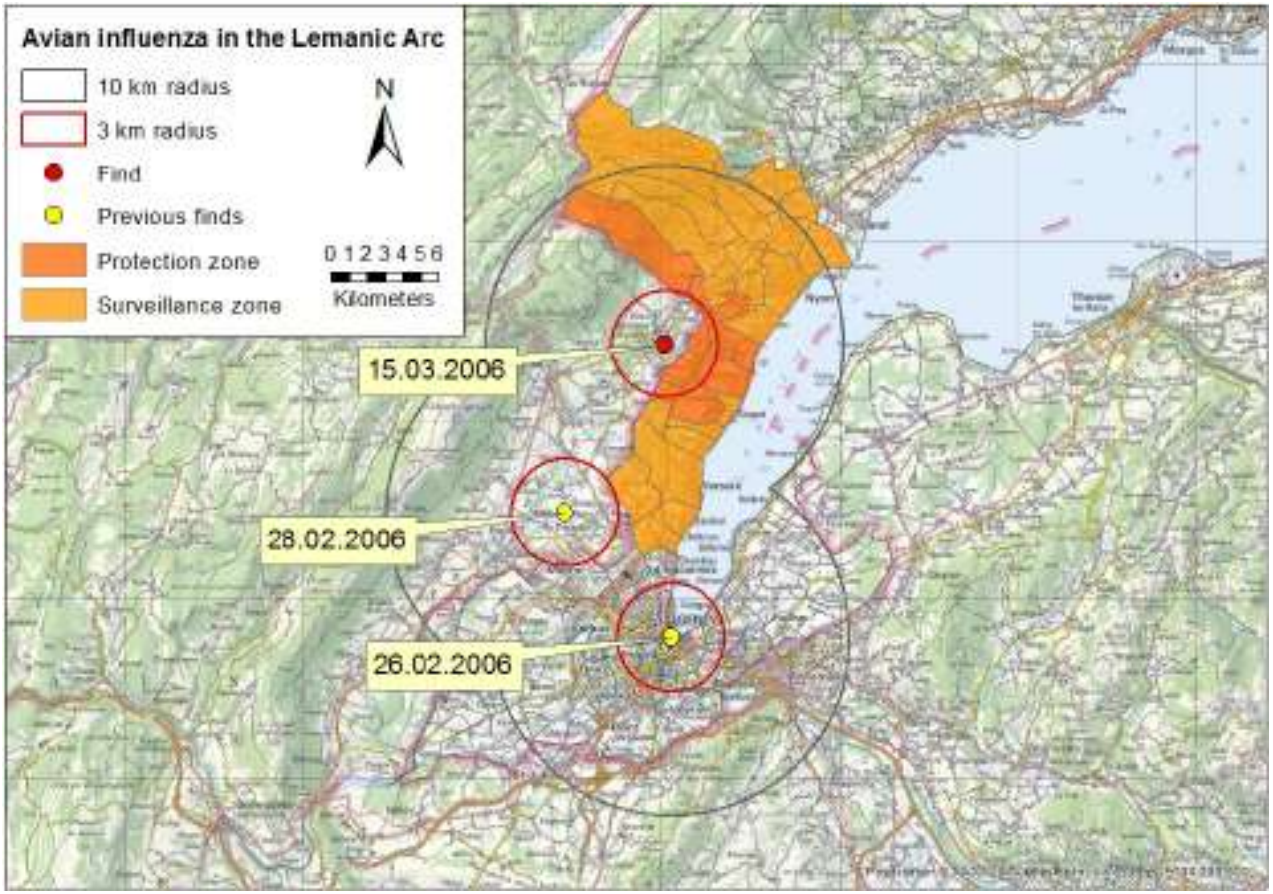
Examples

- BFO does not have `Stuff` (e.g., mucus, cytosol, water). Deny its existence? Add it as a not quite fitting subclass somewhere? Create/reuse a core ontology?
- `Virus` $\sqsubseteq$ `Organism` vs. `Virus` $\sqsubseteq$ `acellular structure`
- A class `Transformation` or a relationship `transforms`?
- `proper parthood` is transitive, irreflexive, and asymmetric. Choose one? Give up on decidable reasoning?

The real use case (thanks to Rolf Grütter)

- Epizootic disease outbreak in the Lemanic Arc (France, Switzerland) in 2006
- Human-pathogenic avian influenza H5N1, modelling & data
- Swiss authorities set up protection zones within a radius of 3km, surveillance zones within a radius of 10km.
- Rules to apply; e.g., *poultry must be kept in the henhouse*
- Need to decide which municipalities to include in the protection zones and which in the surveillance zones

One of the maps



Avian influenza in the Lemanic Arc; National Map 1:200,000 ©2008 swisstopo. Adapted from Perler L (2007).

Geflügelgrippe: Ursprung – Entwicklung – Ausblick. EVD, Bundesamt für Veterinärwesen BVET. (presentation).

The real use case (thanks to Rolf Grütter)

- How to make those decisions better and faster for a next time? (we're in mid 2019 then...)

The real use case (thanks to Rolf Grütter)

- How to make those decisions better and faster for a next time? (we're in mid 2019 then...)
- Two ontologies—epidemiology (finds, etc) and administrative (generic, with Municipality etc.)—and a geodatabase
- Municipality in exactly one region etc.
- The (small) region of the find is contained in the region(s) occupied by the protection zones that are contained in the regions occupied by the surveillance zones

The real use case (thanks to Rolf Grütter)

- How to make those decisions better and faster for a next time? (we're in mid 2019 then...)
- Two ontologies—epidemiology (finds, etc) and administrative (generic, with Municipality etc.)—and a geodatabase
- Municipality in exactly one region etc.
- The (small) region of the find is contained in the region(s) occupied by the protection zones that are contained in the regions occupied by the surveillance zones

Reasoner: .... non-simple .... beyond OWL 2 DL!

Outline

1 Context and motivation

2 Resolving conflicts

3 Implementation trade-offs

4 Conclusions

How to manage such differences?

- Identify type of conflicts that can arise
- Determine how to preempt or to detect them
- Assess options what to do with it when a conflict arises
- Specify a mechanism to keep track of these three aspects
- Devise a way to make this easy to do and document choice

Keet CM, Grütter R. Toward a systematic conflict resolution framework for ontologies. *JBMS*, 2021, 12:15.

Note: meaning negotiation vs conflict resolution

Meaning negotiation concerns deliberations to figure out the precise semantics one wants to represent in the ontology. They are all *positive choices* in the sense of "which of the options is applicable? Then we take that one".

Conflict resolution Concerns choosing one option among a set of two or more options, where that choice is deemed the *'lesser among evils' for that scenario*. It necessarily involves a compromise and making it work requires reengineering something in at least one of the ontologies or as a whole.

Sample scenario to detect and resolve conflicts

Reuse scenario

Plan: import ontology O2 into ontology O1

Tool feedback *(example)*

1. O1+2 violates OWL 2 DL language
2. O1 reifies relations but O2 does not;

Examine sources of conflict *(example)*

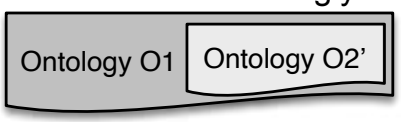
1. o1:part-of = o2:part-of, but o1:part-of is transitive and o2:part-of is used in a qualified cardinality constraint
2. This concerns o1:Vaccination and o2:vaccinates

Resolve conflicts *(choices made for example)*

1. Agree to keep both constraints and thus select a more expressive ontology language.
2. Choose O1's reification approach in line with its modelling style

Implement resolution

1. No further action needed
2. Remodel o2:vaccinates axioms accordingly
3. Import O2' into O1



What are the key sources of conflicts?

- Ontological differences between established theories
  - DOLCE vs BFO
- Ontological differences at the axiom-level
  - parthood antisymmetric or not? [Cotnoir(2010)]

What are the key sources of conflicts?

- Ontological differences between established theories
  - DOLCE vs BFO
- Ontological differences at the axiom-level
  - parthood antisymmetric or not? [Cotnoir(2010)]
- Different modelling styles
  - foundational ontology-inspired or conceptual model-influenced [Fillottrani and Keet(2017), Fillottrani and Keet(2019)]

What are the key sources of conflicts?

- Ontological differences between established theories
  - DOLCE vs BFO
- Ontological differences at the axiom-level
  - parthood antisymmetric or not? [Cotnoir(2010)]
- Different modelling styles
  - foundational ontology-inspired or conceptual model-influenced [Fillottrani and Keet(2017), Fillottrani and Keet(2019)]
- Logic limitations causing conflicts for an ontology, affecting the software ecosystem
  - OWL only or DOL [DOL(2018)] that can do FOL and HOL
- Logic limitations by design, for the purpose of scalability
  - OWL 2 EL vs. OWL 2 DL [Motik et al.(2009)]
- Certain deductions (excluding modelling mistakes) that manifest after adding the axioms, during TDD, or upon ontology matching attempts.
  - disjointness declared among some ancestor

Illustration of language profile conflicts

Requirement "The COVID-relevant medical ontology for information systems should not exceed the OWL 2 EL profile (compatibility with OBO, SNOMED CT, scalability)"

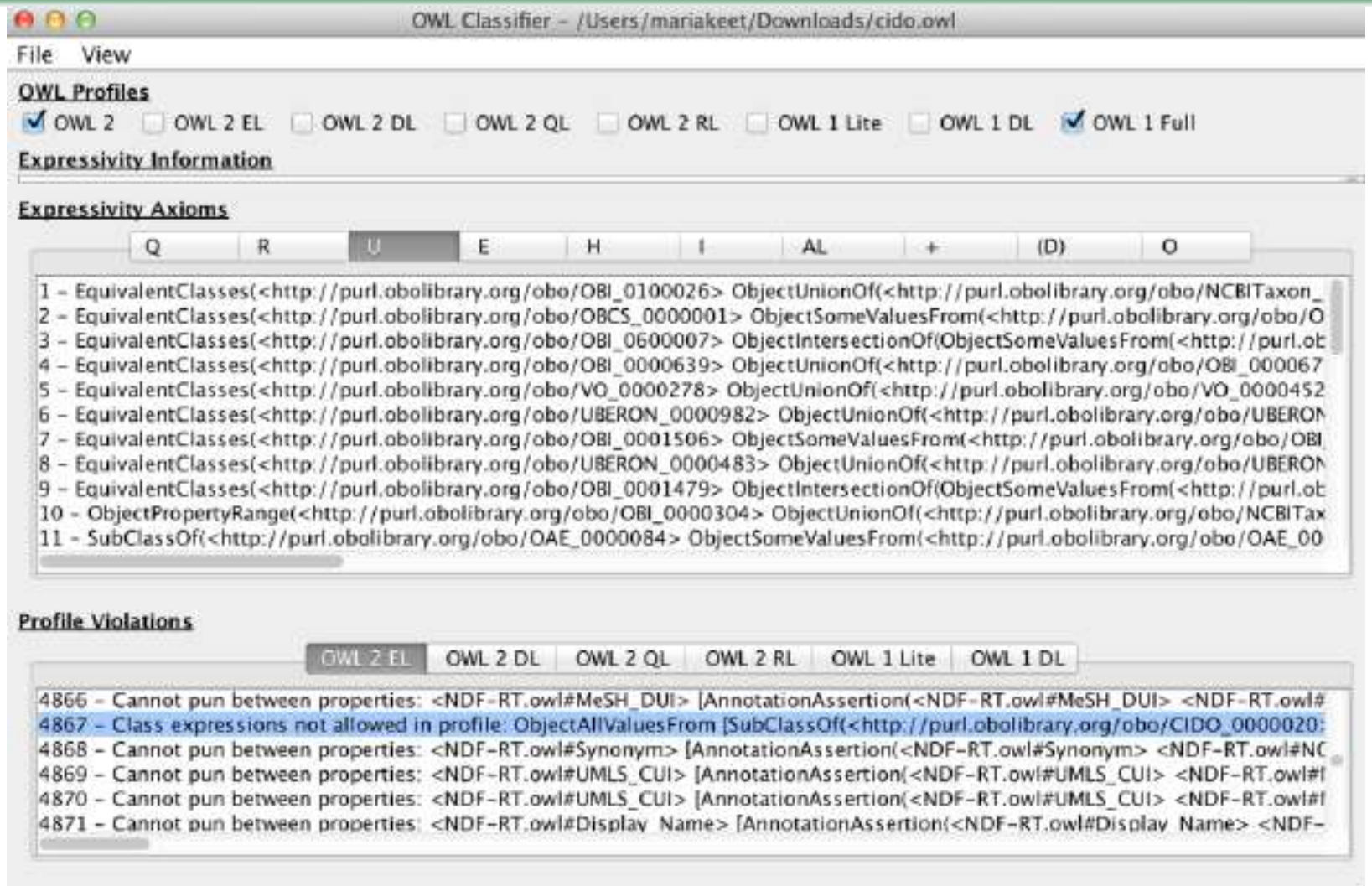
Illustration of language profile conflicts

Requirement "The COVID-relevant medical ontology for information systems should not exceed the OWL 2 EL profile (compatibility with OBO, SNOMED CT, scalability)"

- CIDO ontology for COVID-19 [He et al.(2020)] is not in OWL 2 EL
- Class expression with a universal quantifier on rhs; a.o.:

  $$\textit{'Yale New Haven Hospital SARS-CoV-2 assay'} \sqsubseteq \forall \textit{'EUA-authorized use at'}.\textit{'FDA EUA-authorized organization'}$$

- Need a tool to find violating axioms: the OWL Classifier



Section of the OWL classifier, having detected that `CIDO_0000020` (i.e., Yale New Haven Hospital SARS-CoV-2 assay) violated OWL 2 EL.

OWL Classifier `https://github.com/muhummadPatel/OWL_Classifier`

‘Library’ of common conflicts (selection – 1/2)

No.	Conflict	Description	Examples
		Conflicting theories at the top-level	
1	foundational	ontologies adhere to conflicting theories	BFO, DOLCE, GFO, SUMO, UFO, YAMATO
2	mereological	conflicting mereological theories	with Atom or not, weak vs. strong supplementation
3	topological	conflicting topological theories	region connection calculus on non-simply connected regions
4	building blocks	different ontological commitments embedded in the language	whether roles are part of the fundamental furniture of the universe, 3D + time vs. 4D ‘worms’
		...	

Some of this is ‘easy’ to figure out

- Delegate the choice: use an existing foundational ontology
- Delegate the choice: use an existing ontology language

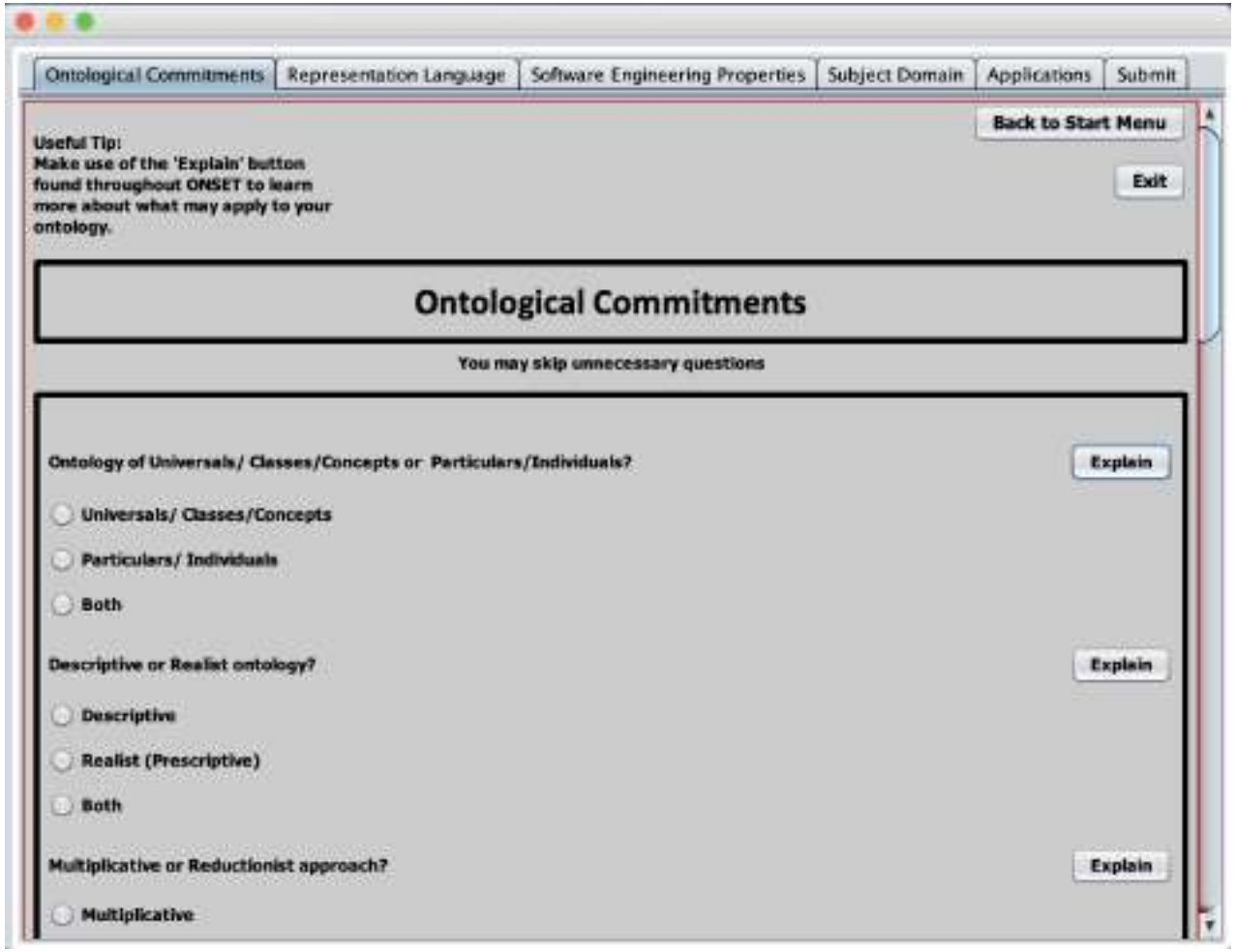
Some of this is 'easy' to figure out

- Delegate the choice: use an existing foundational ontology
  - How to choose an existing foundational ontology?
- Delegate the choice: use an existing ontology language
  - How to choose an existing language?

Some of this is ‘easy’ to figure out

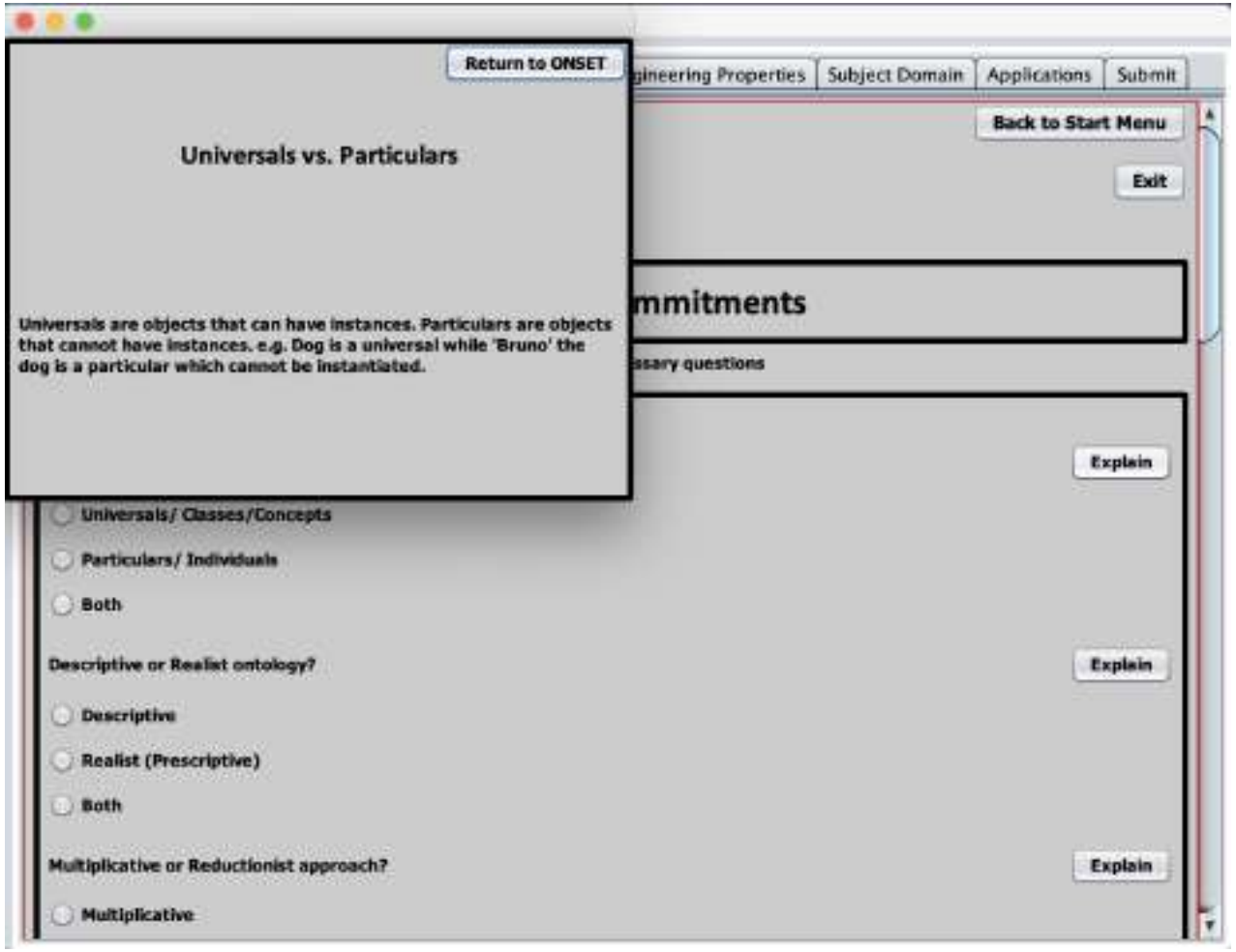
- Delegate the choice: use an existing foundational ontology
  - How to choose an existing foundational ontology?
  - What if it conflicts with the rest of the system?
- Delegate the choice: use an existing ontology language
  - How to choose an existing language?
  - What if it conflicts with the rest of the system?

Choose an existing foundational ontology



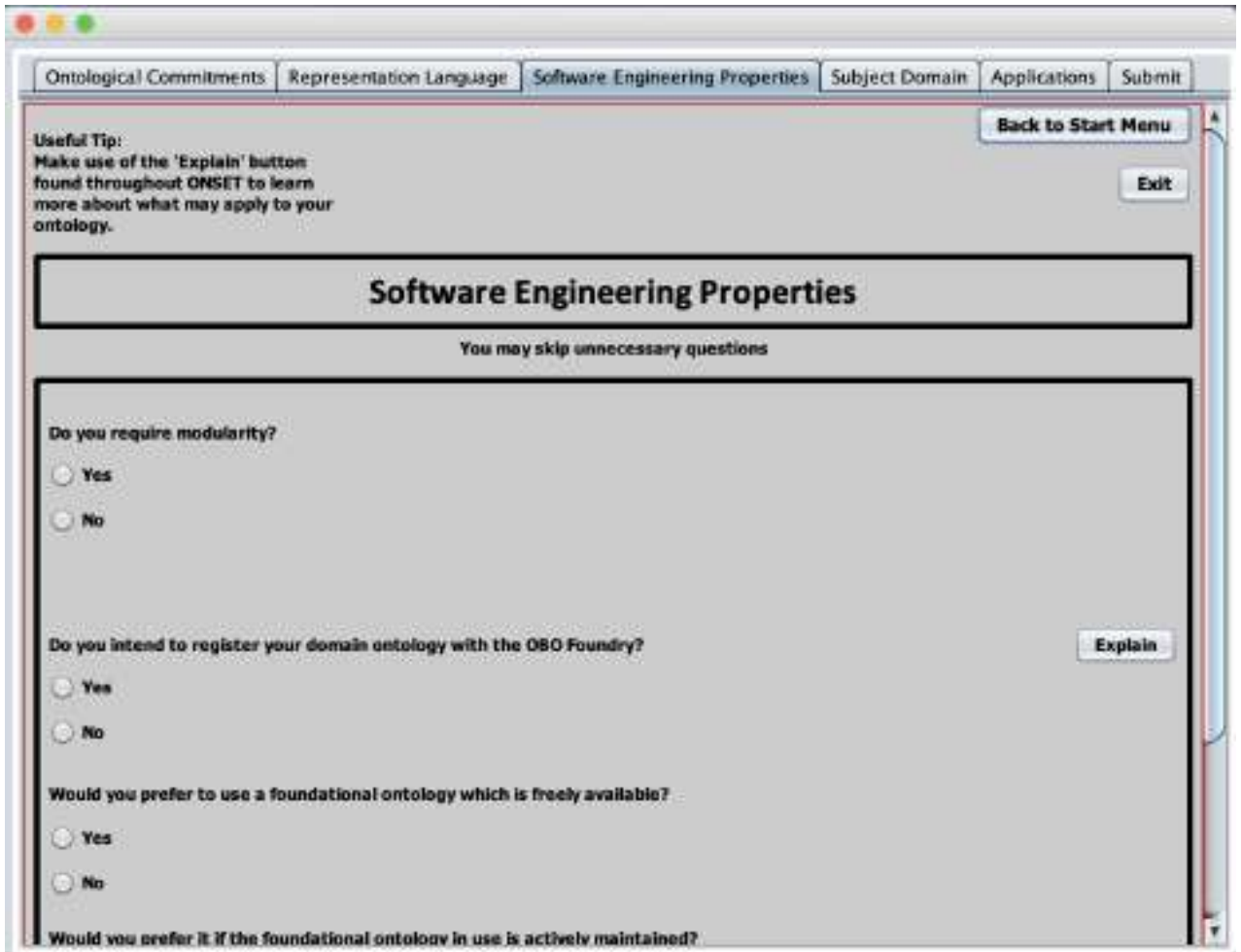
Khan Z, Keet CM. ONSET: Automated Foundational Ontology Selection and Explanation. EKAW'12.

Choose an existing foundational ontology



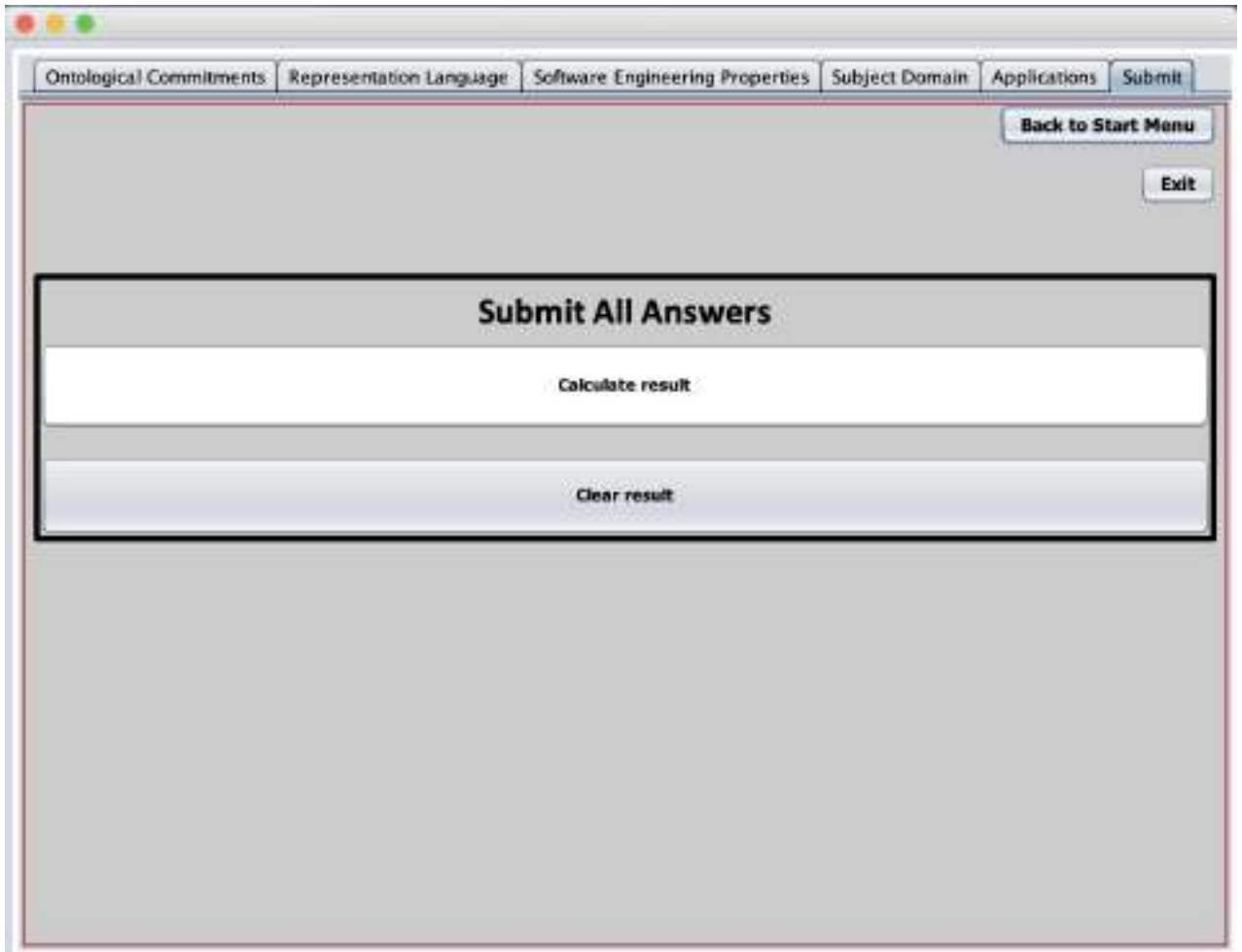
Khan Z, Keet CM. ONSET: Automated Foundational Ontology Selection and Explanation. EKAW'12.

Choose an existing foundational ontology



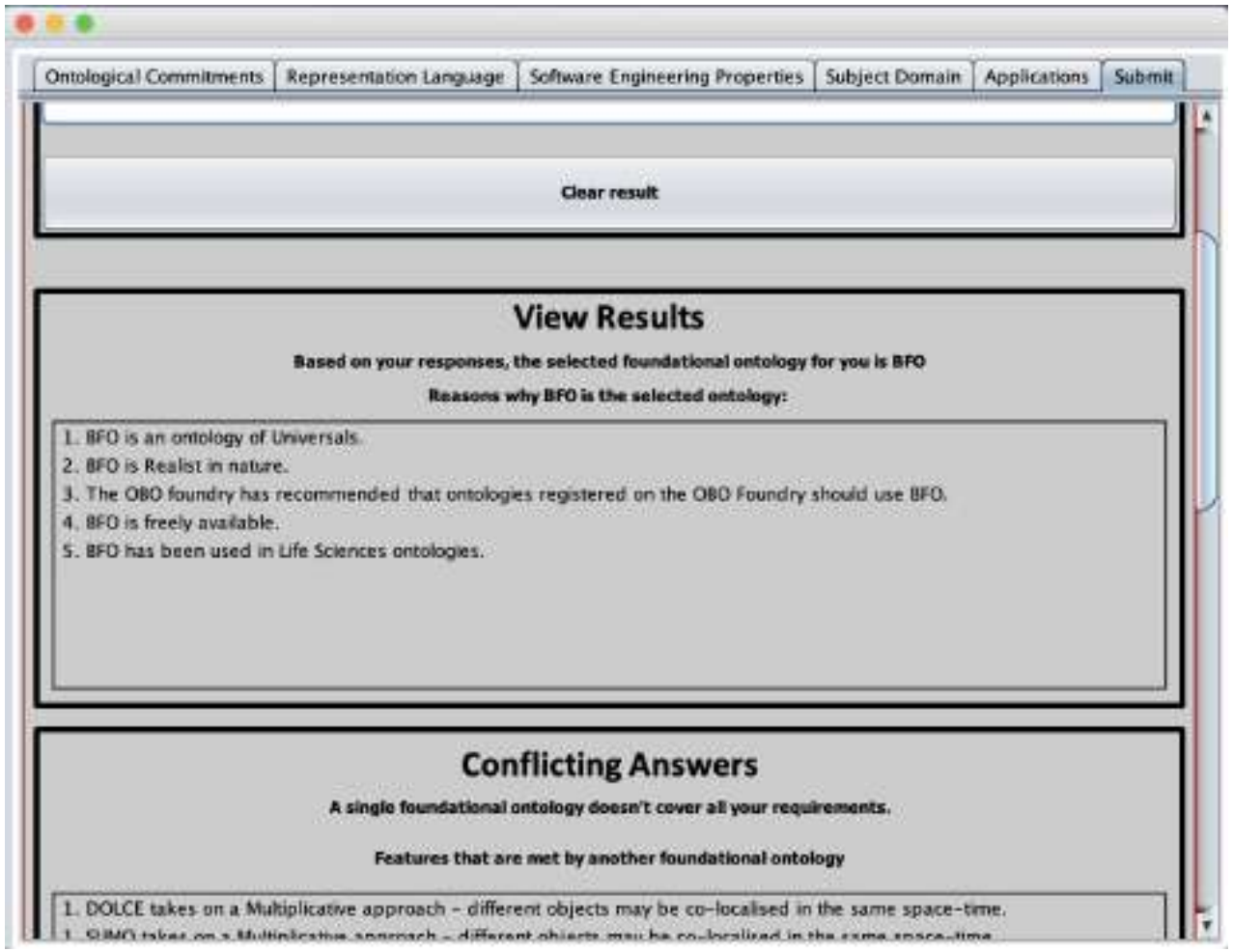
Khan Z, Keet CM. ONSET: Automated Foundational Ontology Selection and Explanation. EKAW'12.

Choose an existing foundational ontology



Khan Z, Keet CM. ONSET: Automated Foundational Ontology Selection and Explanation. EKAW'12.

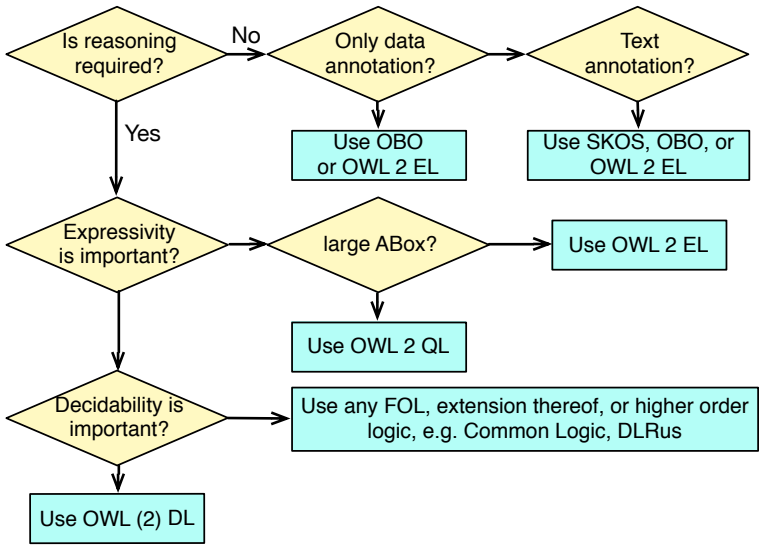
Choose an existing foundational ontology



Khan Z, Keet CM. ONSET: Automated Foundational Ontology Selection and Explanation. EKAW'12.

Consider language: simple or complicated

- Simple purpose-oriented guidance:

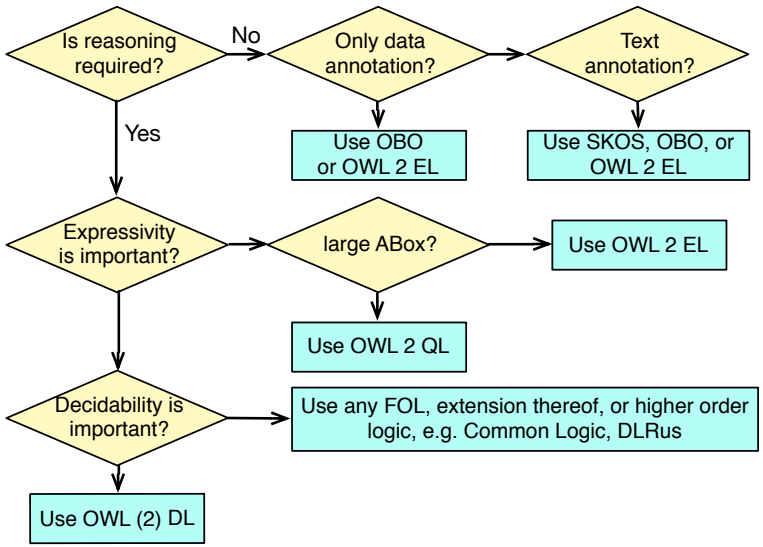


- Use 'translators' (e.g., SKOS $\rightarrow$ OWL, OBO $\rightarrow$ OWL, OWL $\rightarrow$ FOL) or DOL as 'glue'

Keet CM. Transforming semi-structured life science diagrams into meaningful domain ontologies with DiDOn. JBI, 2012, 45(3): 482-494.

Consider language: simple or complicated

- Simple purpose-oriented guidance:



- Use 'translators' (e.g., SKOS $\rightarrow$ OWL, OBO $\rightarrow$ OWL, OWL $\rightarrow$ FOL) or DOL as 'glue'
- Complicated: design your own!

Keet CM. Transforming semi-structured life science diagrams into meaningful domain ontologies with DiDOn. JBI, 2012, 45(3): 482-494.

Well-known fundamental language conflicts

- Attributes/data properties (OWL, UML) or not (OBO)
- Parthood as primitive (originally so in OBO) or not (OWL)
- Some separation of language from 'semantic layer' (OBO naming scheme of entities vs OWL, CL etc.)
- 3D+time vs. 4D (in theory at least; time is costly)

Fillottrani PR, Keet CM. An analysis of commitments in ontology language design. FOIS 2020.

‘Library’ of common conflicts (selection – 2/2)

		...	
		Other conflicts	
9	modeling style	applied vs. foundational	whether there are data property axioms, alike `height` between Person and `xsd:decimal`
		class vs. object property	Infection vs. infected-by
		subsuming roles vs. roles inhering in objects	doctor is-a person vs. doctor inheres-in person
10	language	cultural-linguistic and labeling differences, such as preferred/alt labels, orthography, language variants	population immunity vs herd immunity, color vs colour, and non-1:1 mappings (e.g., ‘river’ vs *fleuve* and *rivière*)

Modelling style example (1/2)

Requirement: Integrate the CIDO and CODO COVID-19 ontologies

Modelling style example (1/2)

Requirement: Integrate the CIDO and CODO COVID-19 ontologies

- CODO: `laboratory testfinding` $\equiv$ {`positive, pending, negative`}
- CIDO: `positive COVID-19 diagnosis` $\sqsubseteq$ `COVID-19 diagnosis`, `presumptive positive COVID-19 diagnosis` $\sqsubseteq$ `COVID-19 diagnosis`, and `negative COVID-19 diagnosis` $\sqsubseteq$ `COVID-19 diagnosis`

Modelling style example 2/2

- (Naming issue, or also ontological: finding (some fact) vs. diagnosis (conclusion drawn from the fact) — when taken in context, intention is the same)

$\Rightarrow$ Class vs. instance representations of the same idea

Modelling style example 2/2

- (Naming issue, or also ontological: finding (some fact) vs. diagnosis (conclusion drawn from the fact) — when taken in context, intention is the same)

⇒ Class vs. instance representations of the same idea

- Solution options:
  - Change CODO to use CIDO's style

Modelling style example 2/2

- (Naming issue, or also ontological: finding (some fact) vs. diagnosis (conclusion drawn from the fact) — when taken in context, intention is the same)

⇒ Class vs. instance representations of the same idea

- Solution options:
  - Change CODO to use CIDO's style
  - Change CIDO to use CODO's style

Modelling style example 2/2

- (Naming issue, or also ontological: finding (some fact) vs. diagnosis (conclusion drawn from the fact) — when taken in context, intention is the same)

⇒ Class vs. instance representations of the same idea

- Solution options:
  - Change CODO to use CIDO's style
  - Change CIDO to use CODO's style
  - A *joint outside option*; e.g.: use attribute + values instead

Record such information: the conflict set

Conflict set grammar for recording individual conflict sets in or between ontologies

```
<conflict-set> ::= <ontology> <ontology> [<diff>]
<ontology>     ::= <IRI> [<species>] <axiom>  {<axiom>} [<inference>]
<species>      ::= "OWL DL" | "OWL Lite" | "OWL Full" | "OWL 2 EL" | "OWL 2 QL" | "OWL 2 RL" |
                   "OWL 2 DL" | "OWL 2 Full" | "FOL" | "HOL"
<axiom>        ::= [<number>] <formula>  [<description>] {<theory>} {<dl-expressivity>}
<theory>       ::= <IRI> | <name> | <IRI> <name> | "none"
<diff>         ::= difference between the inferred axioms sets of the two ontologies
```

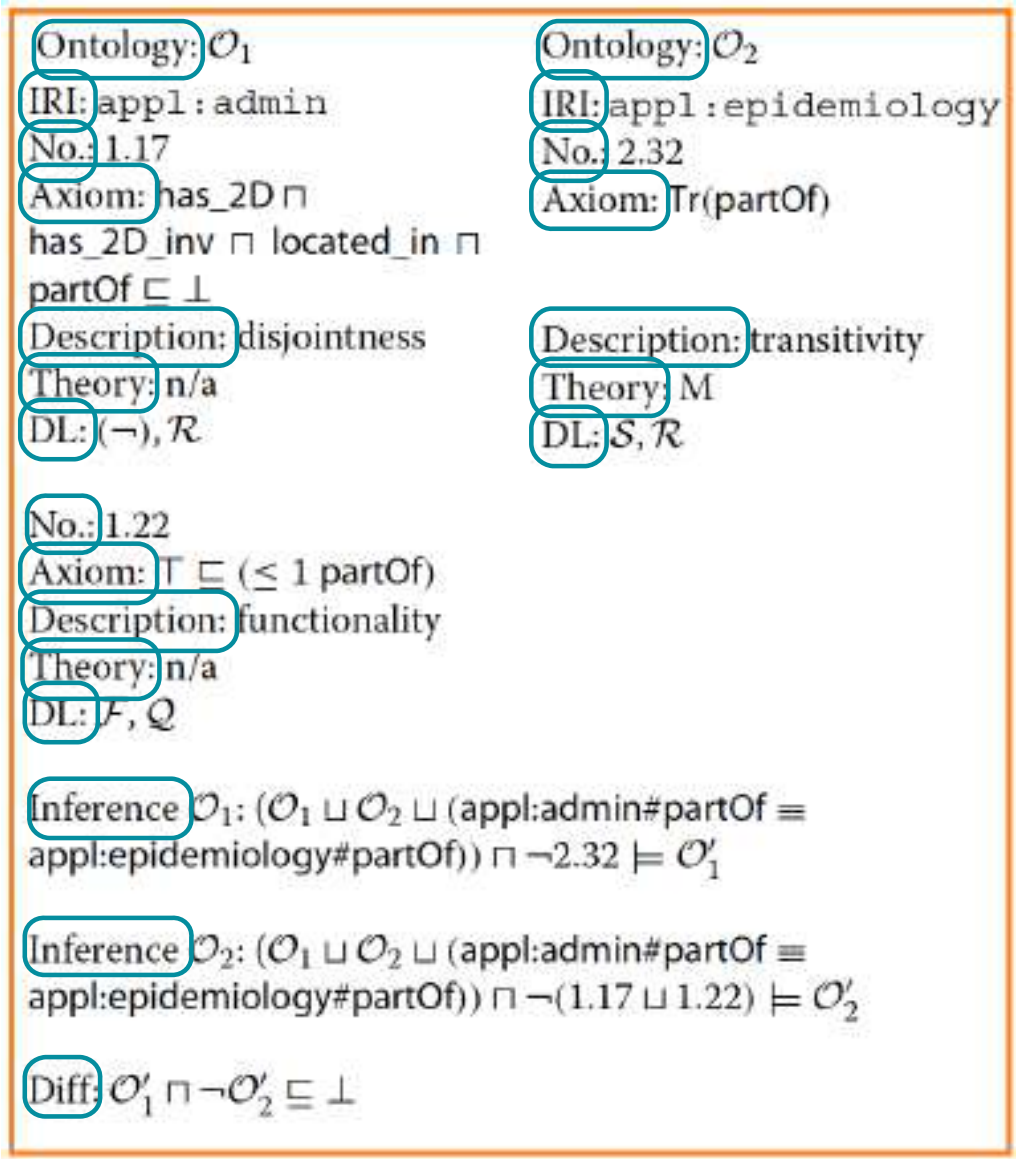
(production rules of most terminals are omitted)

Ontology: $\mathcal{O}_1$	Ontology: $\mathcal{O}_2$
IRI: `appl:admin` No.: 1.17 Axiom: has_2D ⊓ has_2D_inv ⊓ located_in ⊓ partOf ⊑ ⊥ Description: disjointness Theory: n/a DL: $(\neg), \mathcal{R}$	IRI: `appl:epidemiology` No.: 2.32 Axiom: Tr(partOf) Description: transitivity Theory: M DL: $\mathcal{S}, \mathcal{R}$
No.: 1.22 Axiom: $\top \sqsubseteq (\leq 1$ partOf) Description: functionality Theory: n/a DL: $\mathcal{F}, \mathcal{Q}$	

Inference $\mathcal{O}_1$: $(\mathcal{O}_1 \sqcup \mathcal{O}_2 \sqcup$ (appl:admin#partOf $\equiv$ appl:epidemiology#partOf)) $\sqcap \neg 2.32 \models \mathcal{O}'_1$

Inference $\mathcal{O}_2$: $(\mathcal{O}_1 \sqcup \mathcal{O}_2 \sqcup$ (appl:admin#partOf $\equiv$ appl:epidemiology#partOf)) $\sqcap \neg(1.17 \sqcup 1.22) \models \mathcal{O}'_2$

Diff: $\mathcal{O}'_1 \sqcap \neg \mathcal{O}'_2 \sqsubseteq \bot$


Ontology: $\mathcal{O}_1$
IRI: appl:admin
No.: 1.17
Axiom: has_2D ⊓ has_2D_inv ⊓ located_in ⊓ partOf ⊑ ⊥
Description: disjointness
Theory: n/a
DL: $(\neg), \mathcal{R}$
Ontology: $\mathcal{O}_2$
IRI: appl:epidemiology
No.: 2.32
Axiom: Tr(partOf)
Description: transitivity
Theory: M
DL: $\mathcal{S}, \mathcal{R}$
No.: 1.22
Axiom: ⊤ ⊑ (≤ 1 partOf)
Description: functionality
Theory: n/a
DL: $\mathcal{F}, \mathcal{Q}$
Inference $\mathcal{O}_1$: $(\mathcal{O}_1 \sqcup \mathcal{O}_2 \sqcup$ (appl:admin#partOf ≡ appl:epidemiology#partOf)) ⊓ ¬2.32 ⊨ $\mathcal{O}'_1$
Inference $\mathcal{O}_2$: $(\mathcal{O}_1 \sqcup \mathcal{O}_2 \sqcup$ (appl:admin#partOf ≡ appl:epidemiology#partOf)) ⊓ ¬(1.17 ⊔ 1.22) ⊨ $\mathcal{O}'_2$
Diff: $\mathcal{O}'_1 \sqcap \neg\mathcal{O}'_2 \sqsubseteq \bot$

Ontology: $\mathcal{O}_1$	Ontology: $\mathcal{O}_2$
IRI: `appl:admin`	IRI: `appl:epidemiology`
No.: 1.17 Axiom: has_2D ⊓ has_2D_inv ⊓ located_in ⊓ partOf ⊑ ⊥	No.: 2.32 Axiom: Tr(partOf)
Description: disjointness Theory: n/a DL: $(\neg), \mathcal{R}$	Description: transitivity Theory: M DL: $\mathcal{S}, \mathcal{R}$
No.: 1.22 Axiom: $\top \sqsubseteq (\leq 1\ \text{partOf})$ Description: functionality Theory: n/a DL: $\mathcal{F}, \mathcal{Q}$	

Inference $\mathcal{O}_1$: $(\mathcal{O}_1 \sqcup \mathcal{O}_2 \sqcup (\text{appl:admin\#partOf} \equiv \text{appl:epidemiology\#partOf})) \sqcap \neg 2.32 \models \mathcal{O}'_1$

Inference $\mathcal{O}_2$: $(\mathcal{O}_1 \sqcup \mathcal{O}_2 \sqcup (\text{appl:admin\#partOf} \equiv \text{appl:epidemiology\#partOf})) \sqcap \neg(1.17 \sqcup 1.22) \models \mathcal{O}'_2$

Diff: $\mathcal{O}'_1 \sqcap \neg\mathcal{O}'_2 \sqsubseteq \bot$

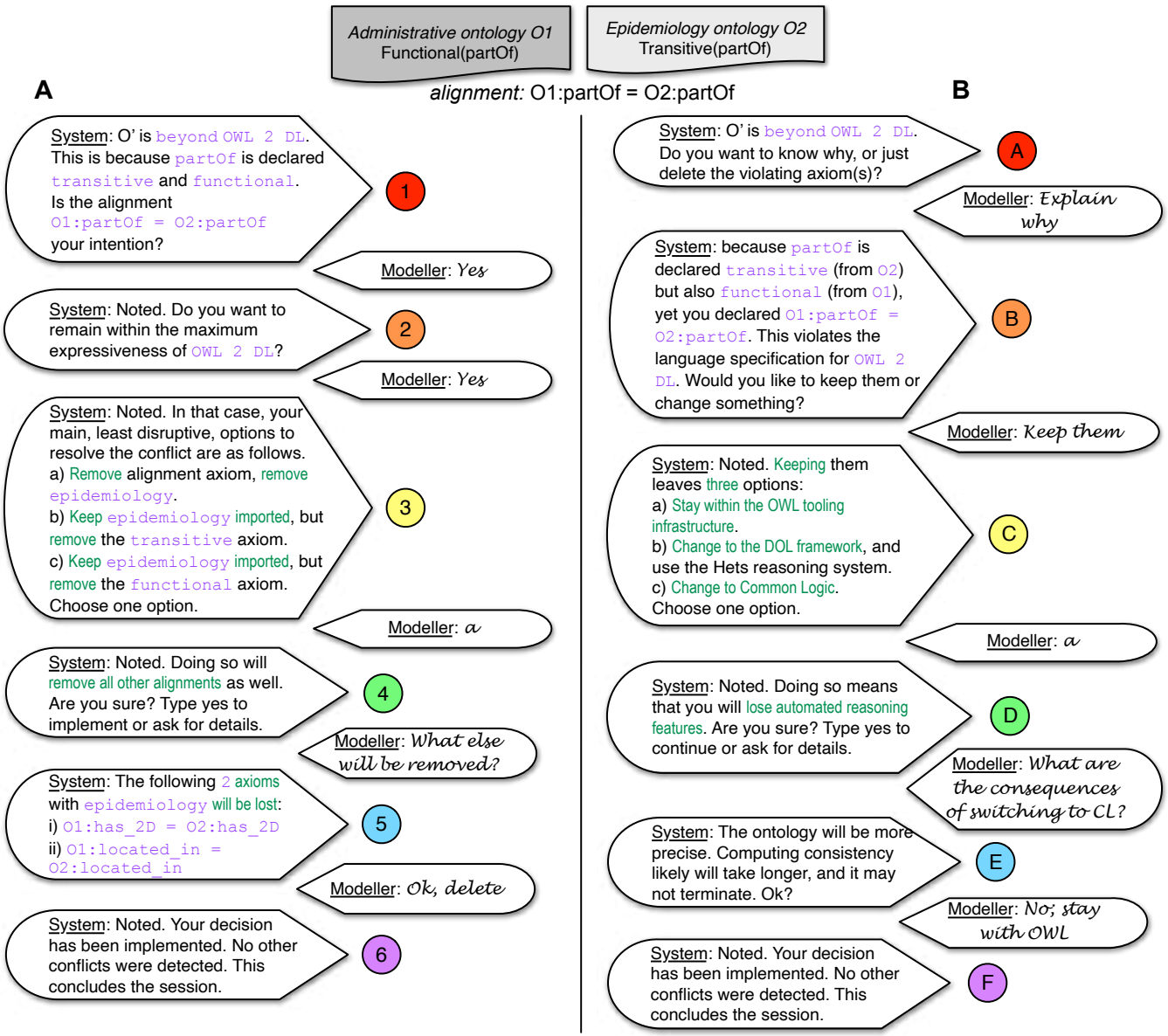
Ontology: $\mathcal{O}_1$	Ontology: $\mathcal{O}_2$
IRI: `appl:admin`	IRI: `appl:epidemiology`
No.: 1.17	No.: 2.32
Axiom: has_2D $\sqcap$ has_2D_inv $\sqcap$ located_in $\sqcap$ partOf $\sqsubseteq \bot$	Axiom: Tr(partOf)
Description: disjointness	Description: transitivity
Theory: n/a	Theory: M
DL: $(\neg), \mathcal{R}$	DL: $\mathcal{S}, \mathcal{R}$
No.: 1.22	
Axiom: $\top \sqsubseteq (\leq 1$ partOf)	
Description: functionality	
Theory: n/a	
DL: $\mathcal{F}, \mathcal{Q}$	

Inference $\mathcal{O}_1$: $(\mathcal{O}_1 \sqcup \mathcal{O}_2 \sqcup$ (appl:admin#partOf $\equiv$ appl:epidemiology#partOf)) $\sqcap \neg 2.32 \models \mathcal{O}'_1$

Inference $\mathcal{O}_2$: $(\mathcal{O}_1 \sqcup \mathcal{O}_2 \sqcup$ (appl:admin#partOf $\equiv$ appl:epidemiology#partOf)) $\sqcap \neg(1.17 \sqcup 1.22) \models \mathcal{O}'_2$

Diff: $\mathcal{O}'_1 \sqcap \neg \mathcal{O}'_2 \sqsubseteq \bot$

Cognitive walk-throughs for the Avian influenza Case Study



Minimal system requirements

- A conflict resolution workflow management system, be it a Question Answering system or another strategy that avails of a knowledge-to-text controlled language, canned text, a decision tree, and two data structures (the conflict set and the resolution options);
- Algorithms to populate the conflict set, which may avail of new wrappers for existing OWL tools to recast their computation and outputs as detection and conflict resolution functionalities;
- End-user usable DOL and CL tools;
- Software support for the language annotation models and extant assessments on modelling style and language conflicts.

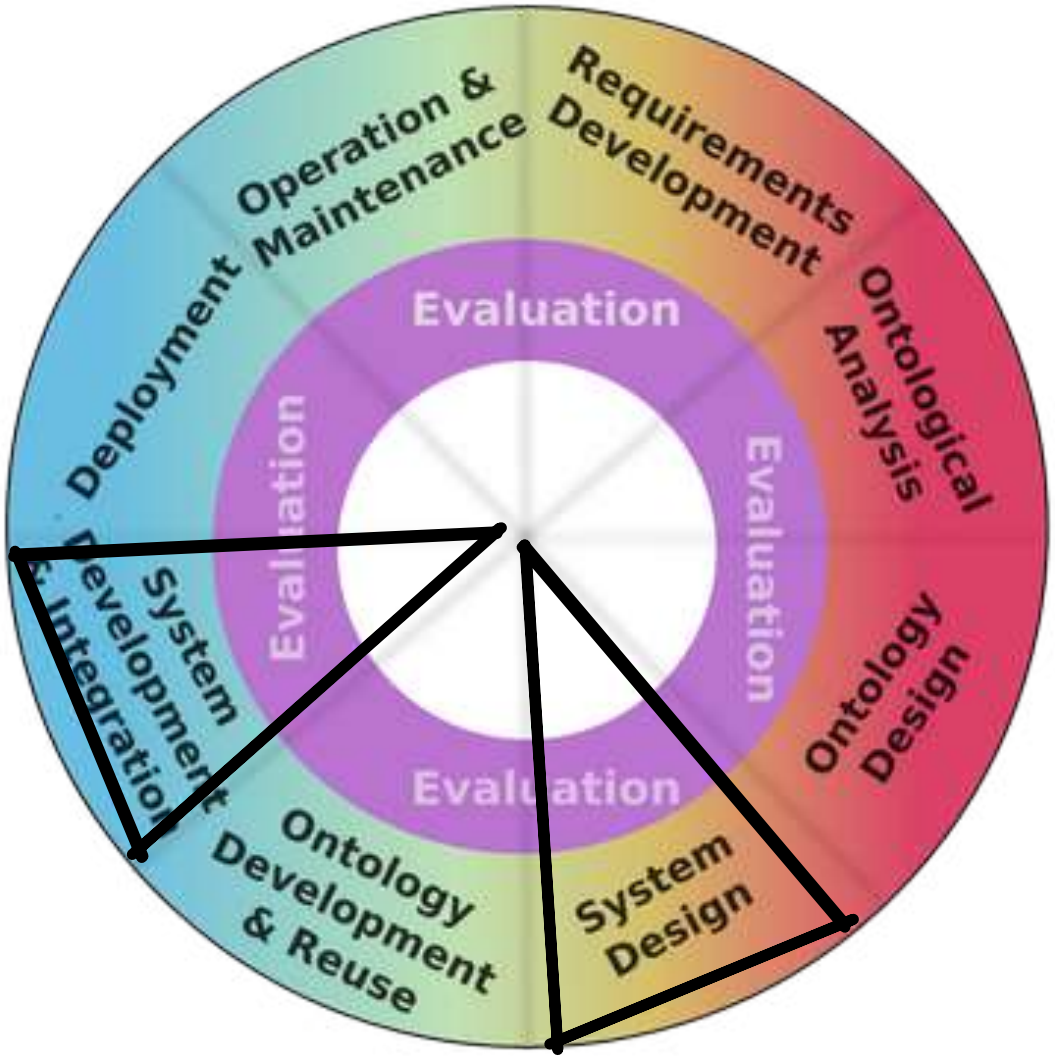
Outline

1 Context and motivation

2 Resolving conflicts

3 Implementation trade-offs

4 Conclusions

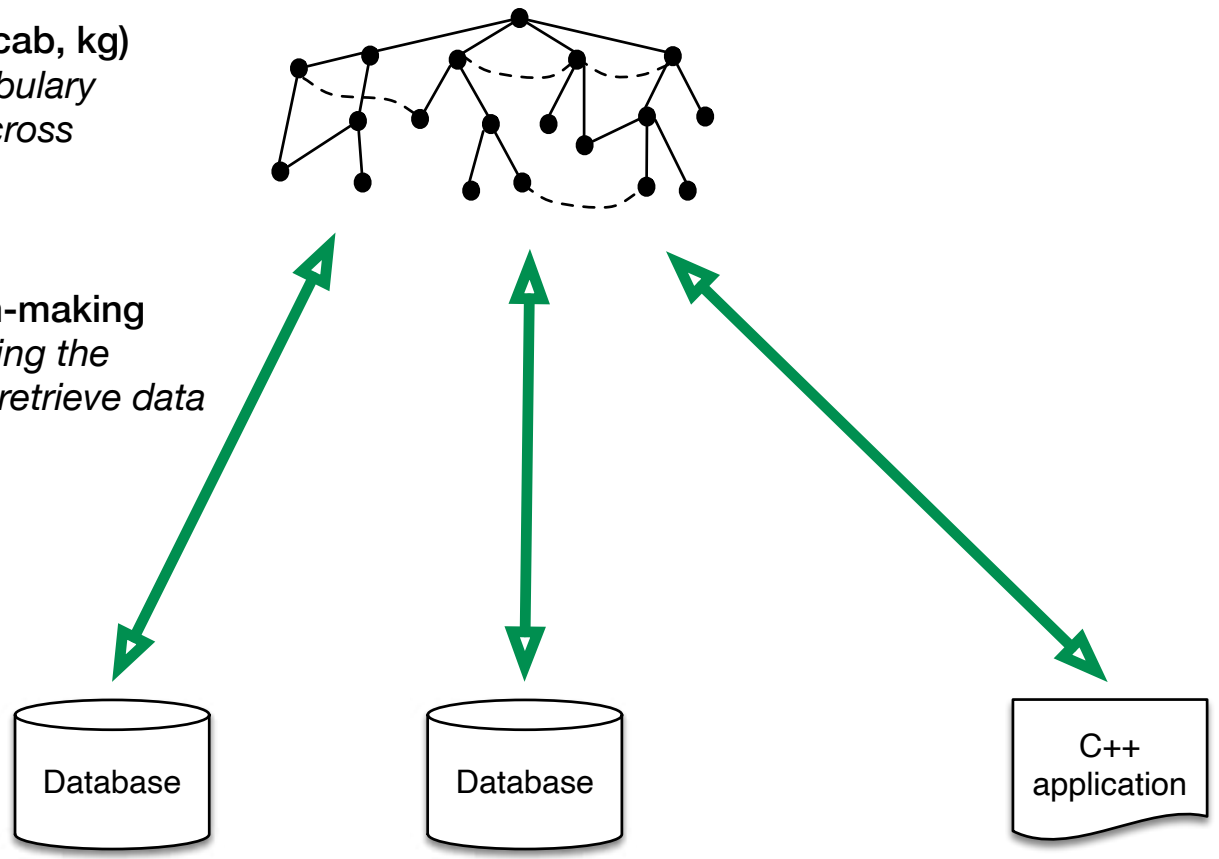

Requirements Development
Ontological Analysis
Ontology Design
System Design
Ontology Development & Reuse
System Development & Integration
Deployment
Operation & Maintenance
Evaluation
Evaluation
Evaluation
Evaluation

Connecting the knowledge to the data

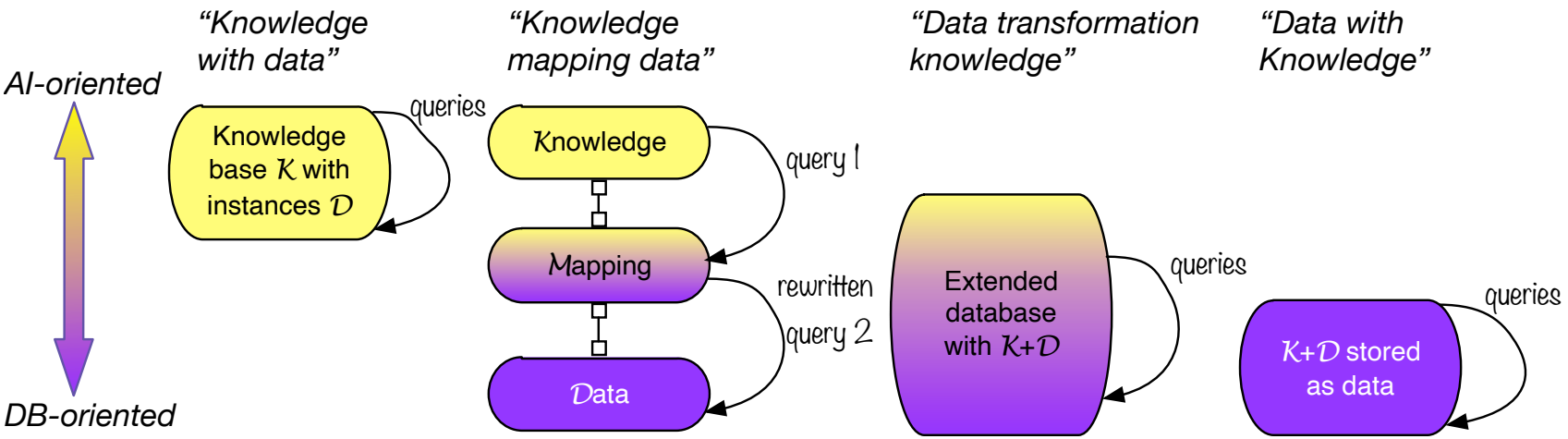
Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Queries for decision-making
formulate queries using the knowledge graph to retrieve data

Implementation
the actual information system that stores and manipulates the data



Knowledge-to-Data Pipeline options



Fillottrani, P.R., Keet, C.M. KnowID: An architecture for efficient Knowledge-driven Information and Data access. Data Intelligence, 2020, 2(4): 487-512.

Key distinguishing features of varying computational cost

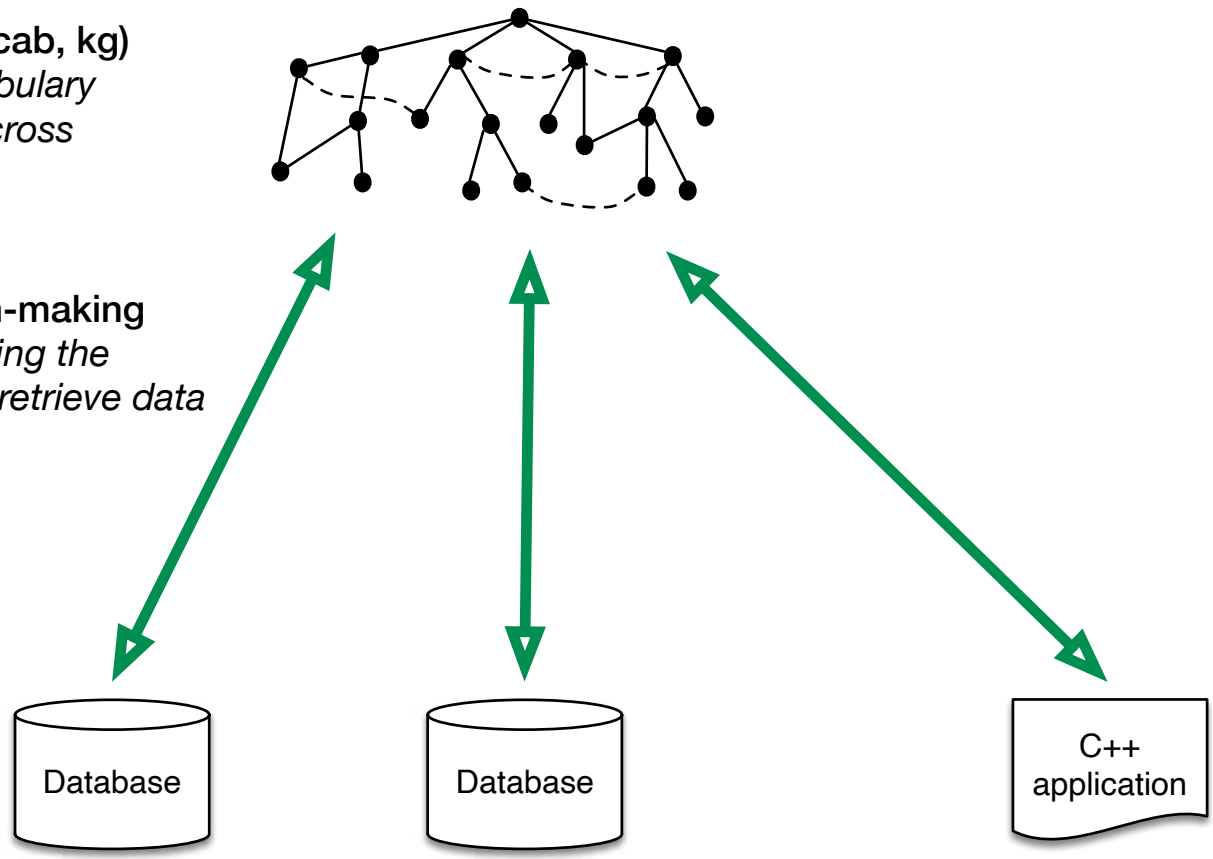
Feature	K @ D	$K \Leftrightarrow D$	$D \bowtie K$	D @ K
World	OWA	OWA+CWA	CWA	CWA
Language for $\mathcal{K}$	OWL	OWL	relational, DL	relational
Language for $\mathcal{D}$	OWL	relational	relational	relational
Query language	SPARQL	SPARQL + SQL (fragment)	SQLP	SQL
Automated reasoning	yes	yes	yes	depends on system
Reasoning w.r.t. data	no separate approach	query rewriting	data completion	data completion
Mapping layer	no	yes	no	no
Transformations	no	no	yes	yes
Entity recasting	no	yes	no	yes
Syntactic sugar	available	available	possible	possible

"Knowledge mapping data": OBDA

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Queries for decision-making
formulate queries using the knowledge graph to retrieve data

Implementation
the actual information system that stores and manipulates the data

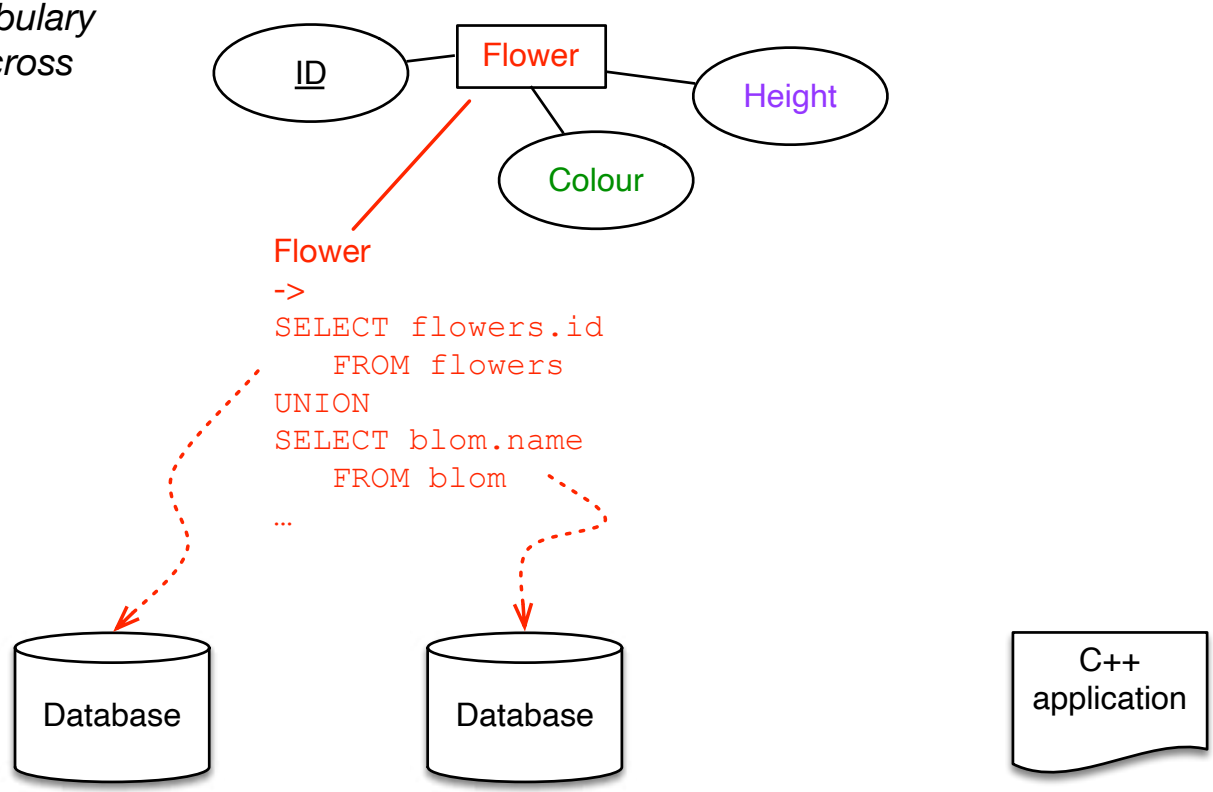


"Knowledge mapping data": OBDA

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Mapping layer
links each entity to a query over the data source(s)

Implementation
the actual information system that stores and manipulates the data

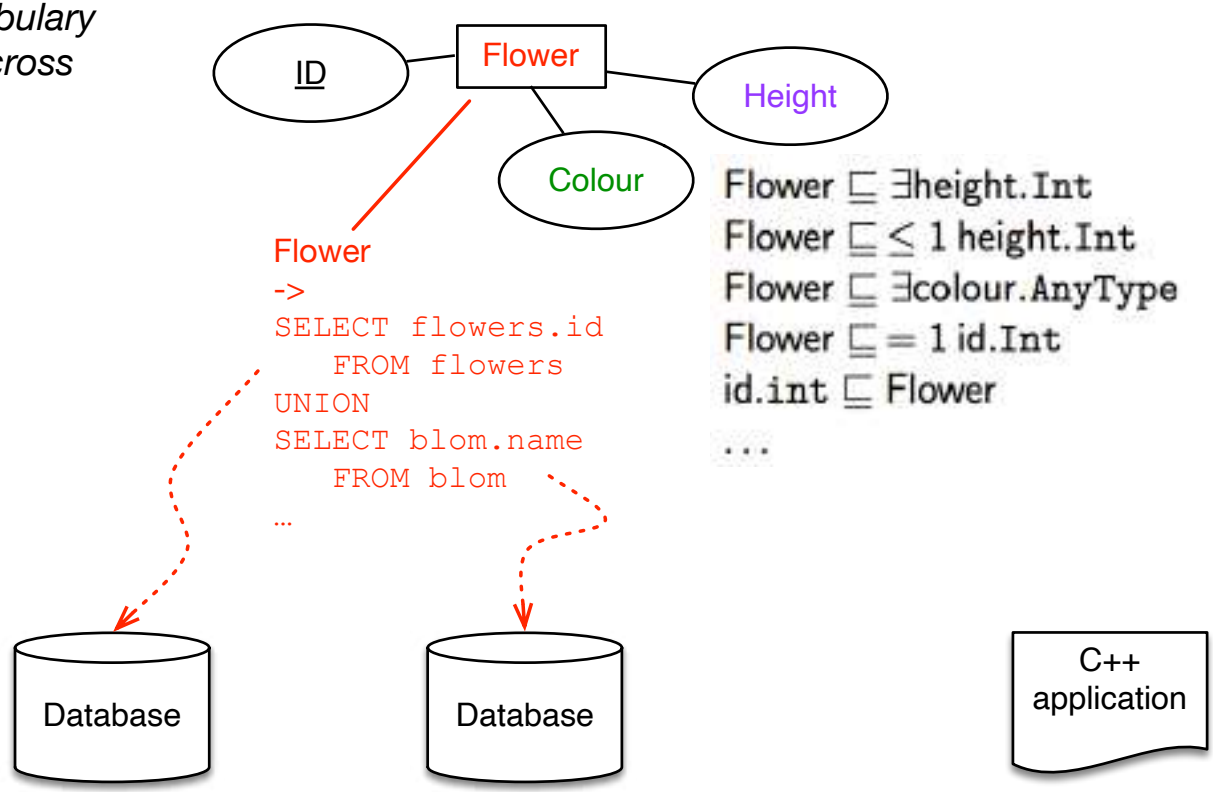


"Knowledge mapping data": OBDA

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

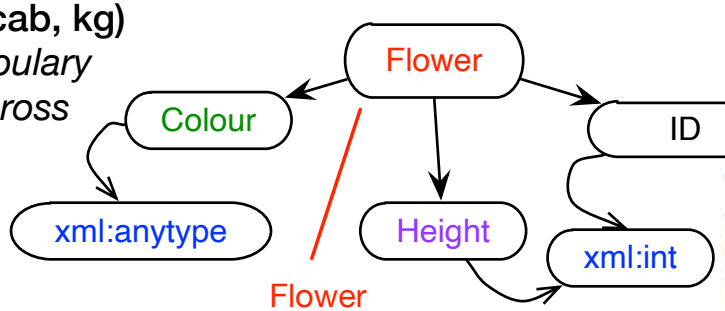
Mapping layer
links each entity to a query over the data source(s)

Implementation
the actual information system that stores and manipulates the data



"Knowledge mapping data": OBDA

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications



$$\begin{aligned}
&\mathtt{Flower} \sqsubseteq \exists \mathtt{height.Int}\\
&\mathtt{Flower} \sqsubseteq \leq 1\ \mathtt{height.Int}\\
&\mathtt{Flower} \sqsubseteq \exists \mathtt{colour.AnyType}\\
&\mathtt{Flower} \sqsubseteq = 1\ \mathtt{id.Int}\\
&\mathtt{id.int} \sqsubseteq \mathtt{Flower}\\
&\ldots
\end{aligned}$$

Mapping layer
links each entity to a query over the data source(s)

```
Flower
->
SELECT flowers.id
   FROM flowers
UNION
SELECT blom.name
   FROM blom
...
```

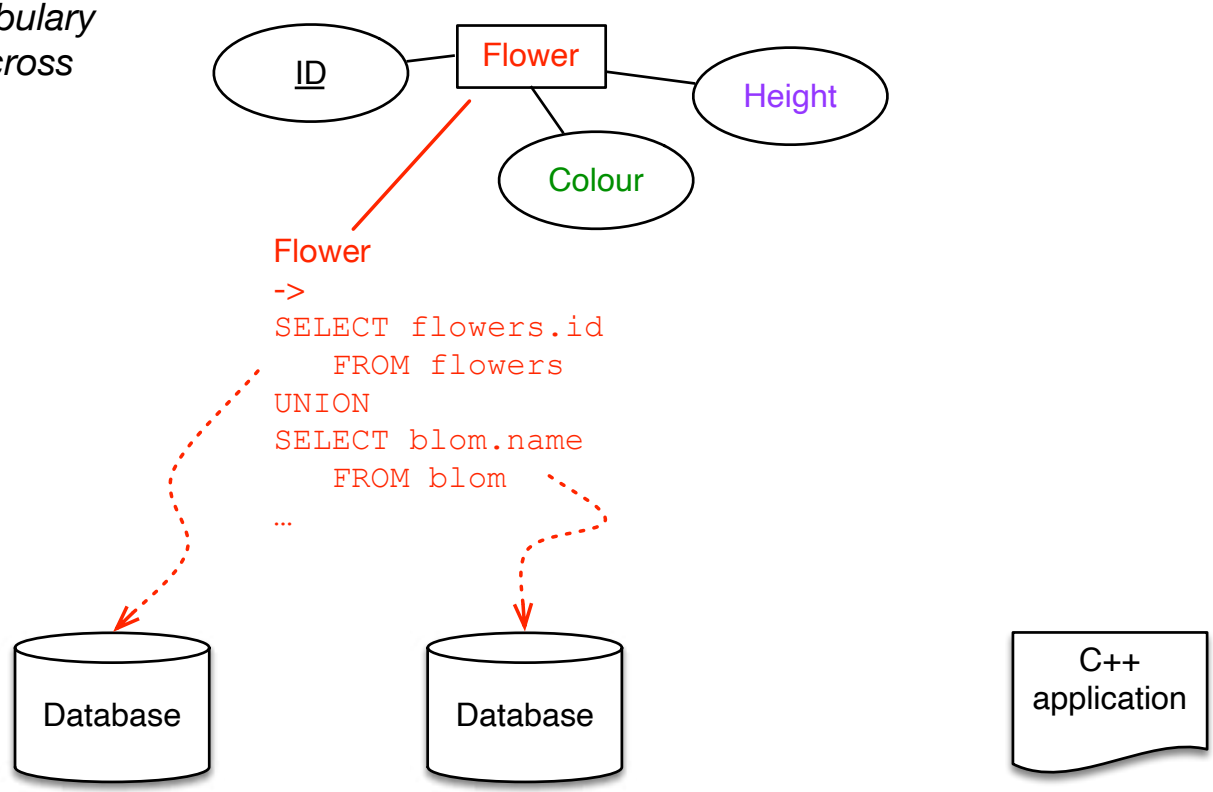
Implementation
the actual information system that stores and manipulates the data

"Knowledge mapping data": OBDA

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Mapping layer
links each entity to a query over the data source(s)

Implementation
the actual information system that stores and manipulates the data

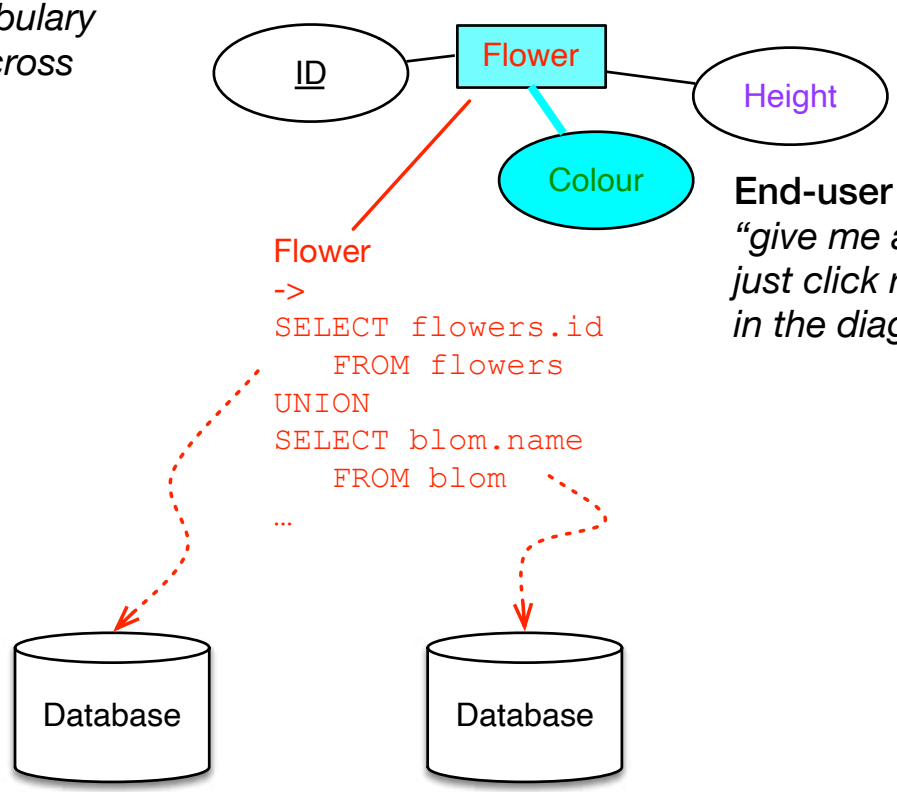


"Knowledge mapping data": OBDA

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Mapping layer
links each entity to a query over the data source(s)

Implementation
the actual information system that stores and manipulates the data



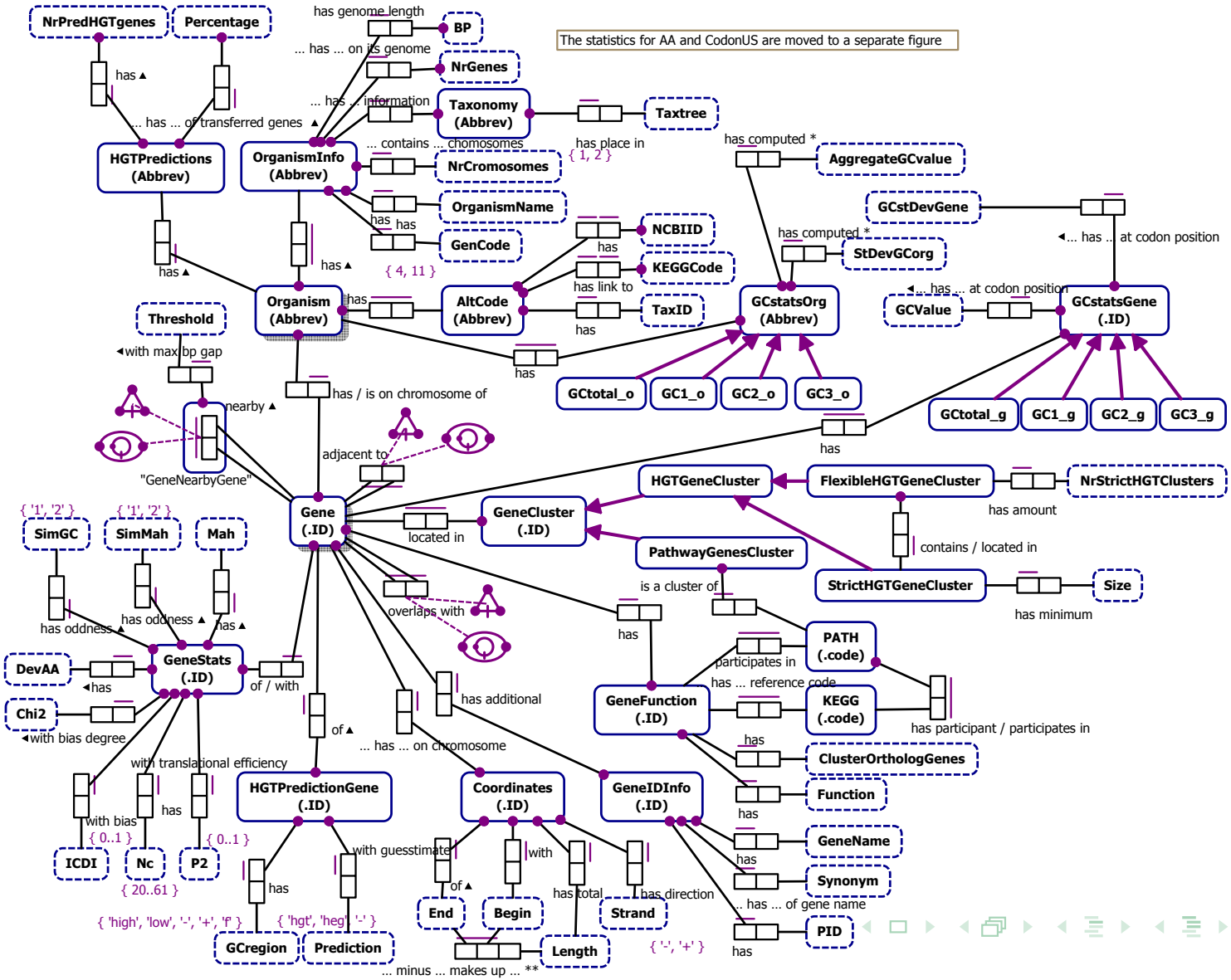
End-user query
"give me all red flowers"
just click relevant elements in the diagram

The WONDER System with the early version

- Horizontal Gene Transfer (HGT) database [Garcia-Vallvé et al.(2003)]
- Reverse engineer the conceptual data model
- Formalise it in OWL 2 QL
- Create mappings
- Create (web-based) interface for browsing, querying, and answering as front-end to OBDA back-end

Calvanese D, Keet CM, Nutt W, Rodriguez-Muro M, Stefanoni G. Web-based Graphical Querying of Databases through an Ontology: the WONDER System. ACM SAC 2010.

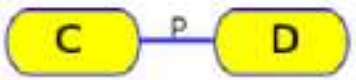
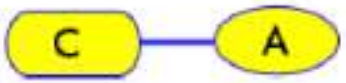
"Knowledge mapping data": OBDA example in genomics



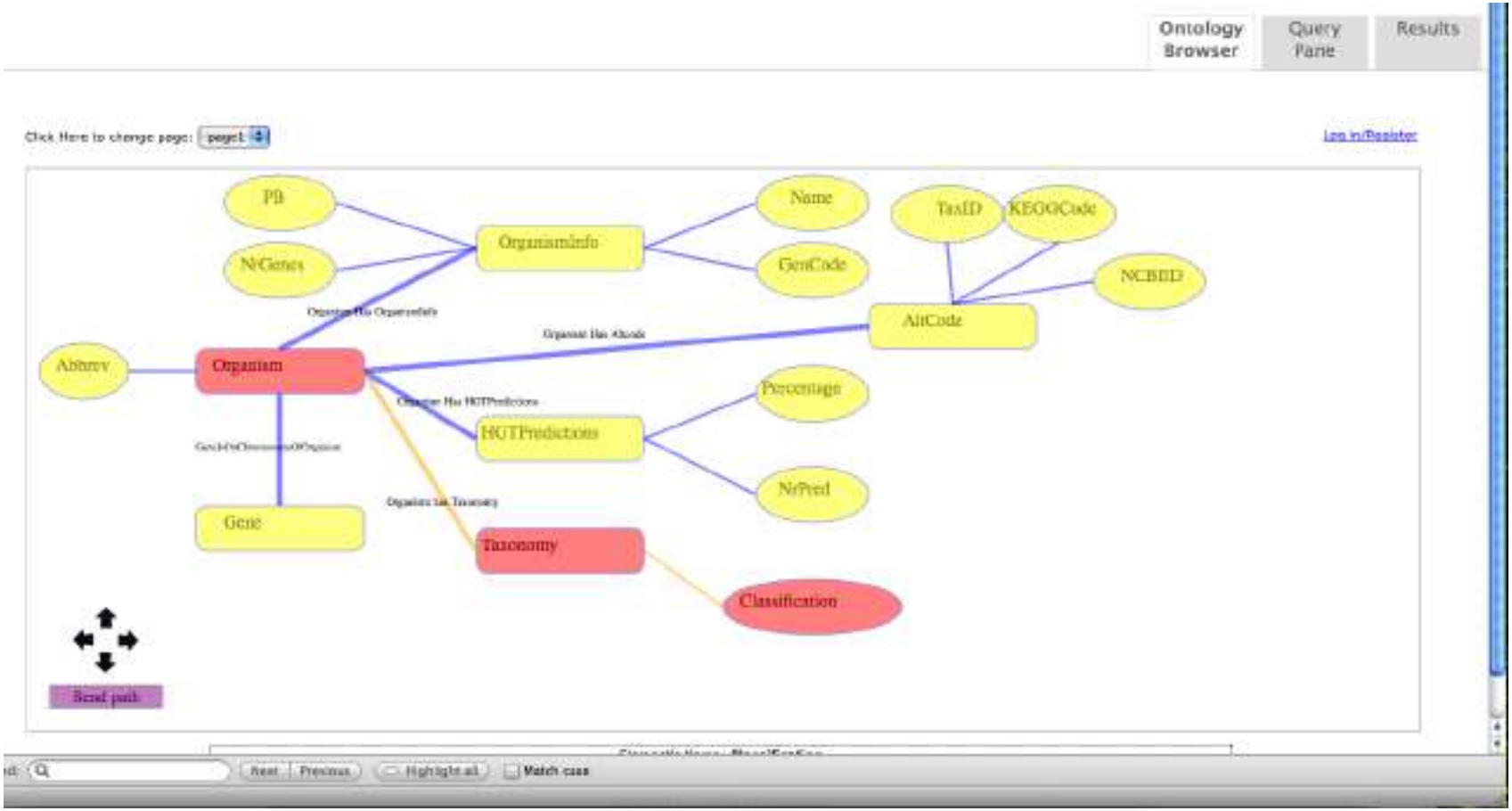
"Knowledge mapping data": OBDA example in genomics

```
        </mapping>
        <mapping id="Ending codons">
            <CQ string="EndingCodon(getEndingCodon($CODON)), TripletCode(getEndingCodon($CODON),$CODON)"/>
            <SQLQuery string="SELECT CODON FROM ENDINGCODONS"/>
        </mapping>
        <mapping id="OrganismHasCodon">
            <CQ string="OHCHasOrganism(getOrganismHasCodon($BUNDLEID),getOrganism($ID)),
OHCHasCodon(getOrganismHasCodon($BUNDLEID),getCodon($CODON)), OrganismHasCodon(getOrganismHasCodon($BUNDLEID)),
CodonValueOrg(getOrganismHasCodon($BUNDLEID),$CODONVALUE), CodonSD(getOrganismHasCodon($BUNDLEID),$CODONSD),
RSCUorg(getOrganismHasCodon($BUNDLEID),$RSCU), SDRSCU(getOrganismHasCodon($BUNDLEID),$SDRSCU)"/>
            <SQLQuery string="SELECT ID, CODON, CODONVALUE, CODONSD, BUNDLEID, RSCU, SDRSCU FROM
ORGANISMHASCODON"/>
        </mapping>
        <mapping id="GeneHasCodon">
            <CQ string="GHCHasGene(getGeneHasCodon($BUNDLEID),getGene($ID)),
GHCHasCodon(getGeneHasCodon($BUNDLEID),getCodon($CODON)), GeneHasCodon(getGeneHasCodon($BUNDLEID)),
CodonValueGene(getGeneHasCodon($BUNDLEID),$CODONVALUE), RSCUgene(getGeneHasCodon($BUNDLEID),$RSCU)"/>
            <SQLQuery string="SELECT ID, CODON, CODONVALUE, BUNDLEID, RSCU FROM GENEHASCODON"/>
        </mapping>
```

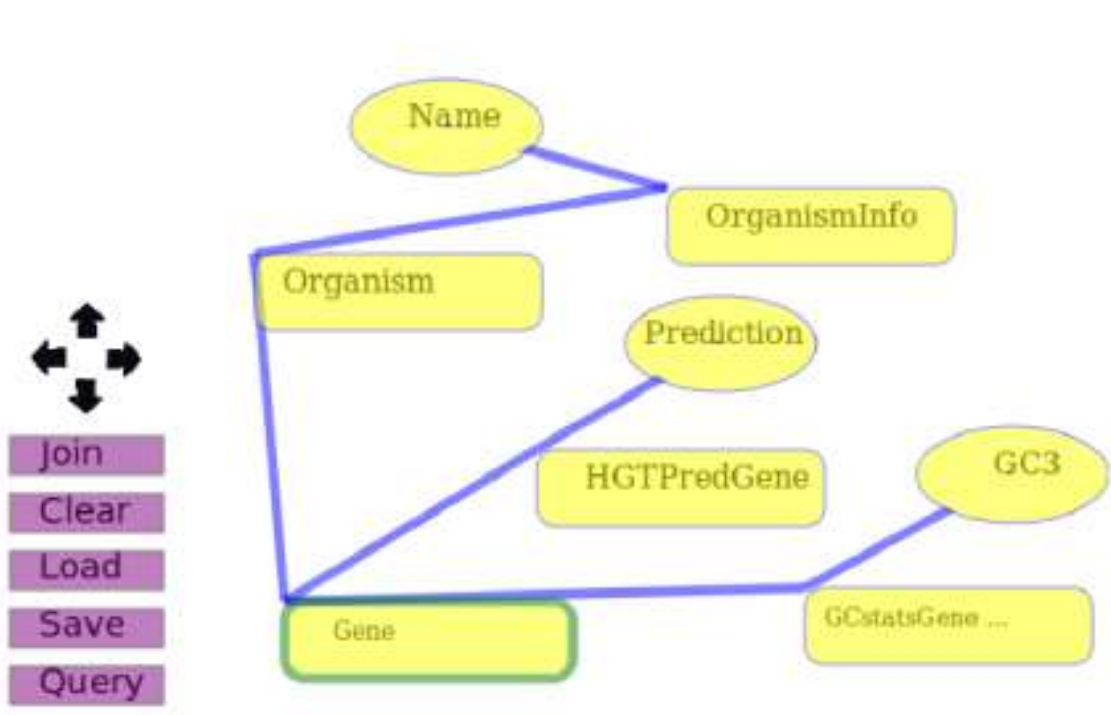
"Knowledge mapping data": OBDA example in genomics

Construct	Graphical Element	Semantic
Class		$C \sqsubseteq \top$
Object Property		$\exists P \sqsubseteq C$ $\exists P^- \sqsubseteq D$
Data Property		$\delta(A) \sqsubseteq C$ $\rho(A) \sqsubseteq \top_d$
SubClass Relationship		$C \sqsubseteq D$

"Knowledge mapping data": OBDA example in genomics



"Knowledge mapping data": OBDA example in genomics

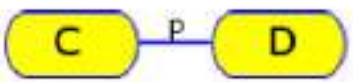
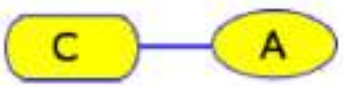


Type	Concept
Concept	OrganismInfo
Label	$3 (edit)
Selected?	false (edit)
Query	SELECT q1.cb FROM sparqltable (SELECT $cb $e $bb $bi WHERE { $c rdf:type 'Organism' . $d rdf:type 'OrganismInfo' . $bh rdf:type 'HGTPredictionGene' . $bf rdf:type 'GCstatsGene' . $bf rdf:type 'GCstats' . $cb rdf:type 'Gene' . $cb :GeneHasOrganism $c . $c :OrganismHasOrganismInfo $d . $cb :GeneHasGCstatsGene $bf . $cb :GeneHasHGTPredictionGene $bh . $d :OrganismName $e . $bf :GC3 $bb . $bh :Prediction $bi }) q1 WHERE { q1.e LIKE '%Neisseria%' AND (q1.bi = 'hgt' OR q1.bb > '80'))

Manage Constraints

Retrieve all genes of the organisms Neisseria for which horizontal gene transfer is predicted or have a GC3 value > 80

"Knowledge mapping data": OBDA example in genomics

Construct	Graphical Element	Semantic
Class node		$C(x), D(x)$
Object Property link		$C(x), P(x,y), D(y)$
Data Property node and link		$C(x), A(x,y)$

"Knowledge mapping data": OBDA

- OBDA with Ontop [Calvanese et al.(2017)] now more elaborate and more robust
- More case studies: Statoil and EPnet [Calvanese et al.(2016)]

"Knowledge mapping data": OBDA

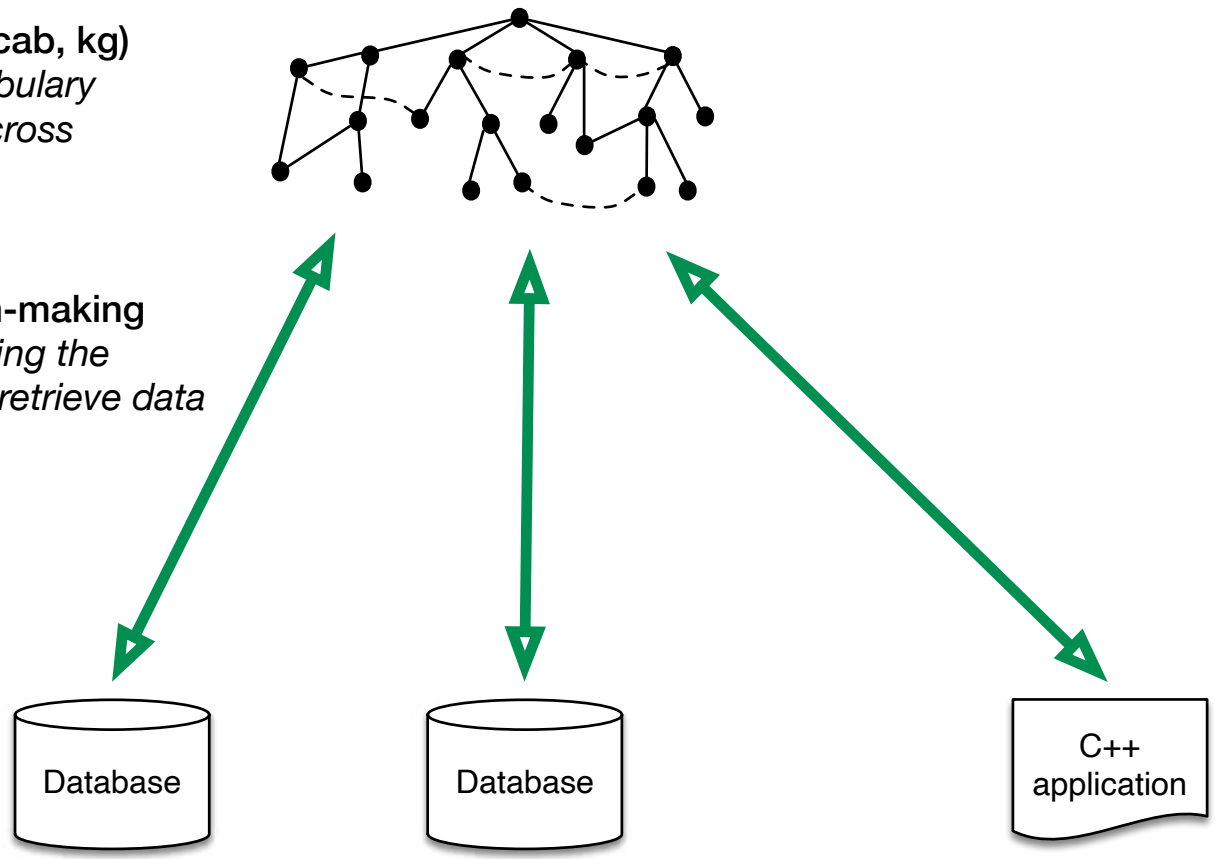
- OBDA with Ontop [Calvanese et al.(2017)] now more elaborate and more robust
- More case studies: Statoil and EPnet [Calvanese et al.(2016)]
- Downsides
  - The mapping layer: cumbersome construction and maintenance
  - Low expressiveness for ontology language
  - Limitations on types of queries

"Data-transformation-knowledge" example: KnowID

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Queries for decision-making
formulate queries using the knowledge graph to retrieve data

Implementation
the actual information system that stores and manipulates the data

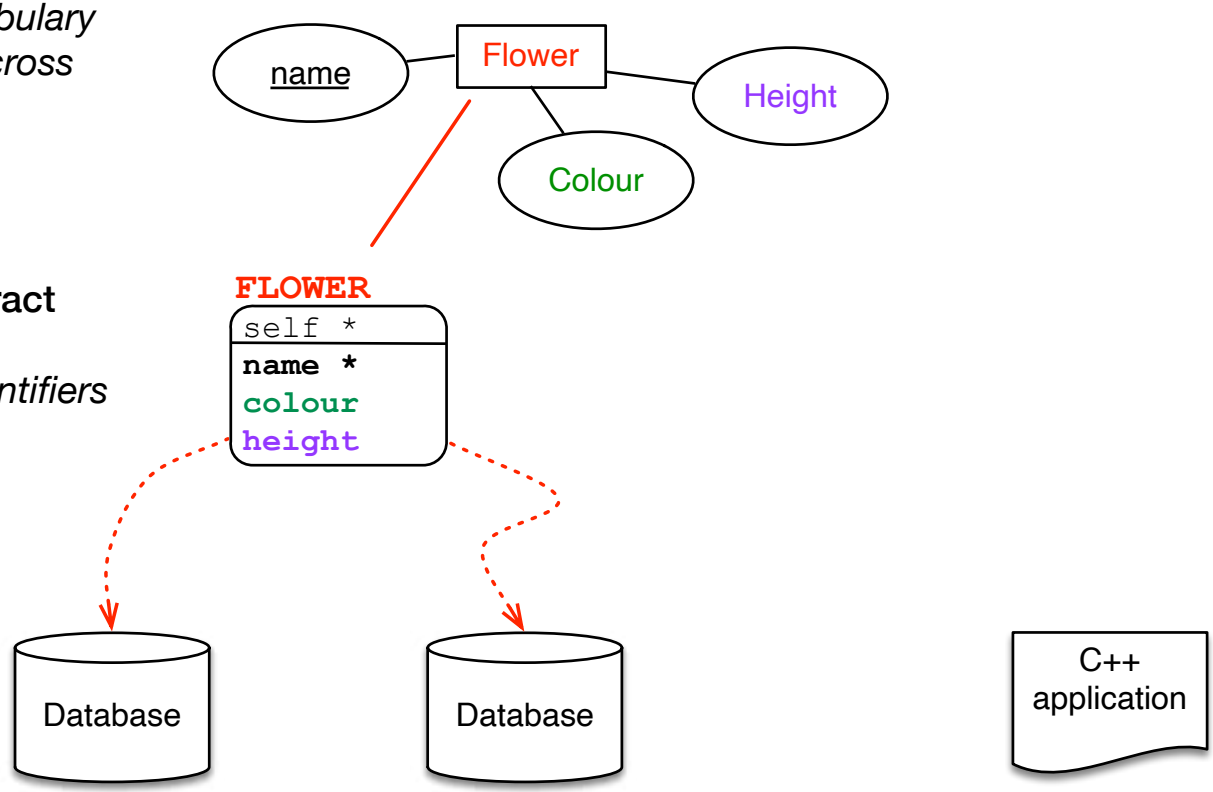


"Data-transformation-knowledge" example: KnowID

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

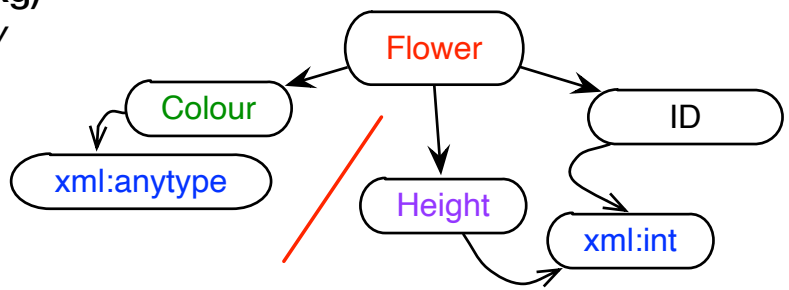
Transformation via abstract relational model
with additional virtual *identifiers*

Implementation
the actual information system that stores and manipulates the data

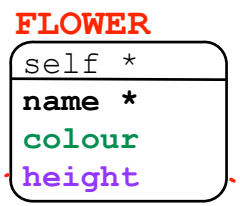


"Data-transformation-knowledge" example: KnowID

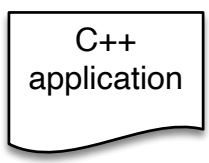
Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications



Transformation via abstract relational model
with additional virtual *identifiers*

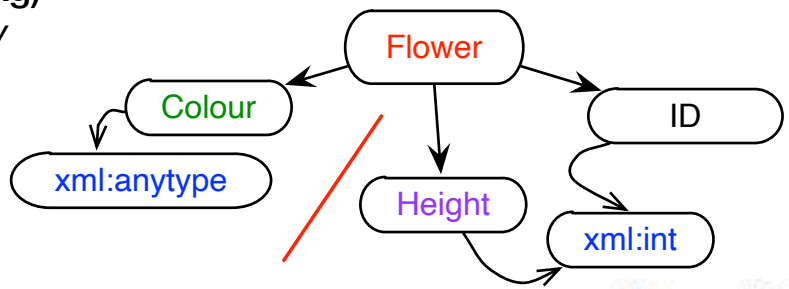


Implementation
the actual information system that stores and manipulates the data

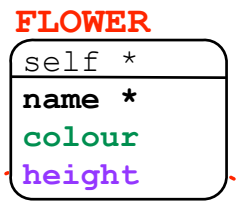


"Data-transformation-knowledge" example: KnowID

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

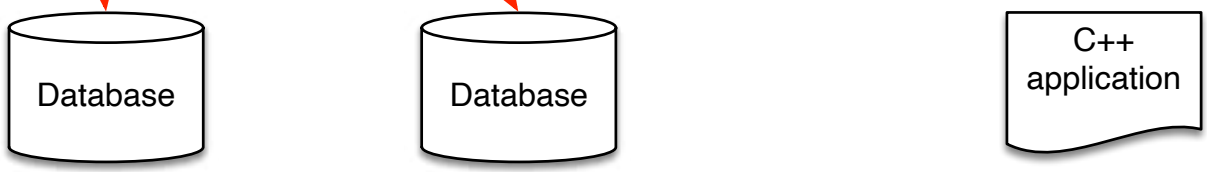


Transformation via abstract relational model
with additional virtual *identifiers*

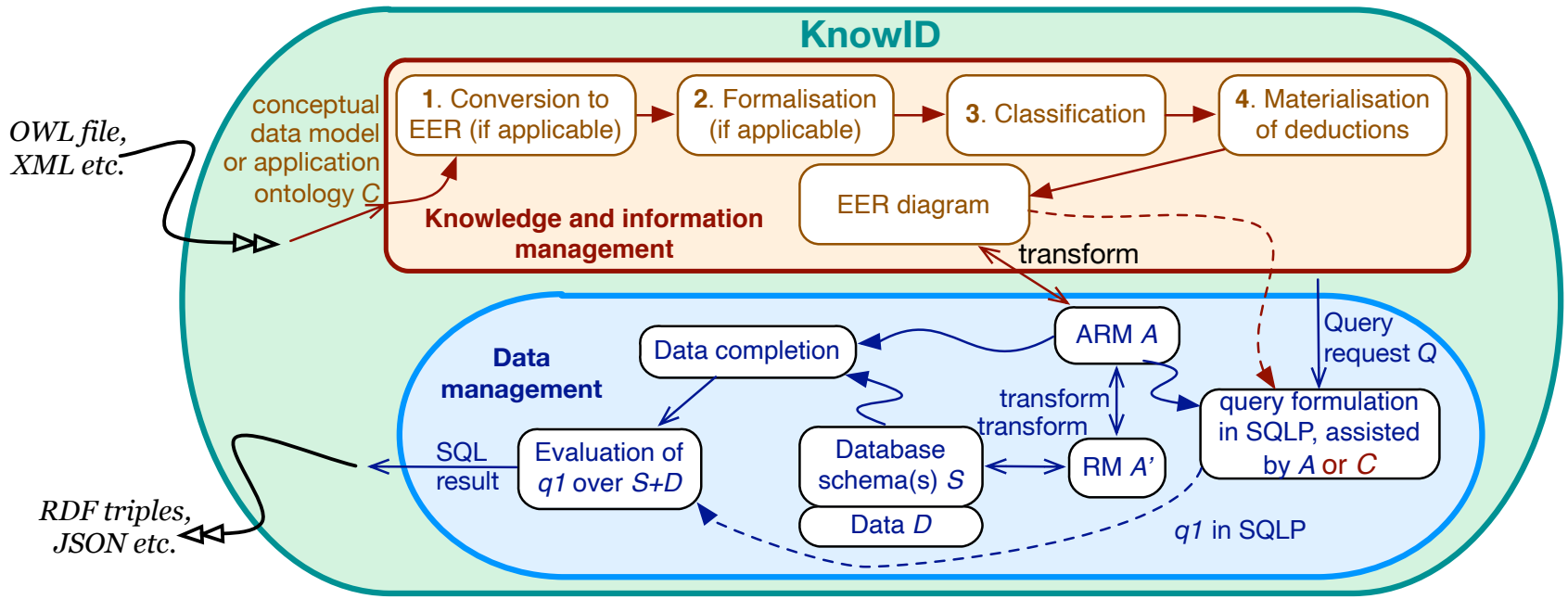


$$\begin{aligned}
&\mathtt{Flower} \sqsubseteq \exists \mathtt{height.Int}\\
&\mathtt{Flower} \sqsubseteq \leq 1\ \mathtt{height.Int}\\
&\mathtt{Flower} \sqsubseteq \exists \mathtt{colour.AnyType}\\
&\mathtt{Flower} \sqsubseteq = 1\ \mathtt{id.Int}\\
&\mathtt{id.int} \sqsubseteq \mathtt{Flower}\\
&\ldots
\end{aligned}$$

Implementation
the actual information system that stores and manipulates the data

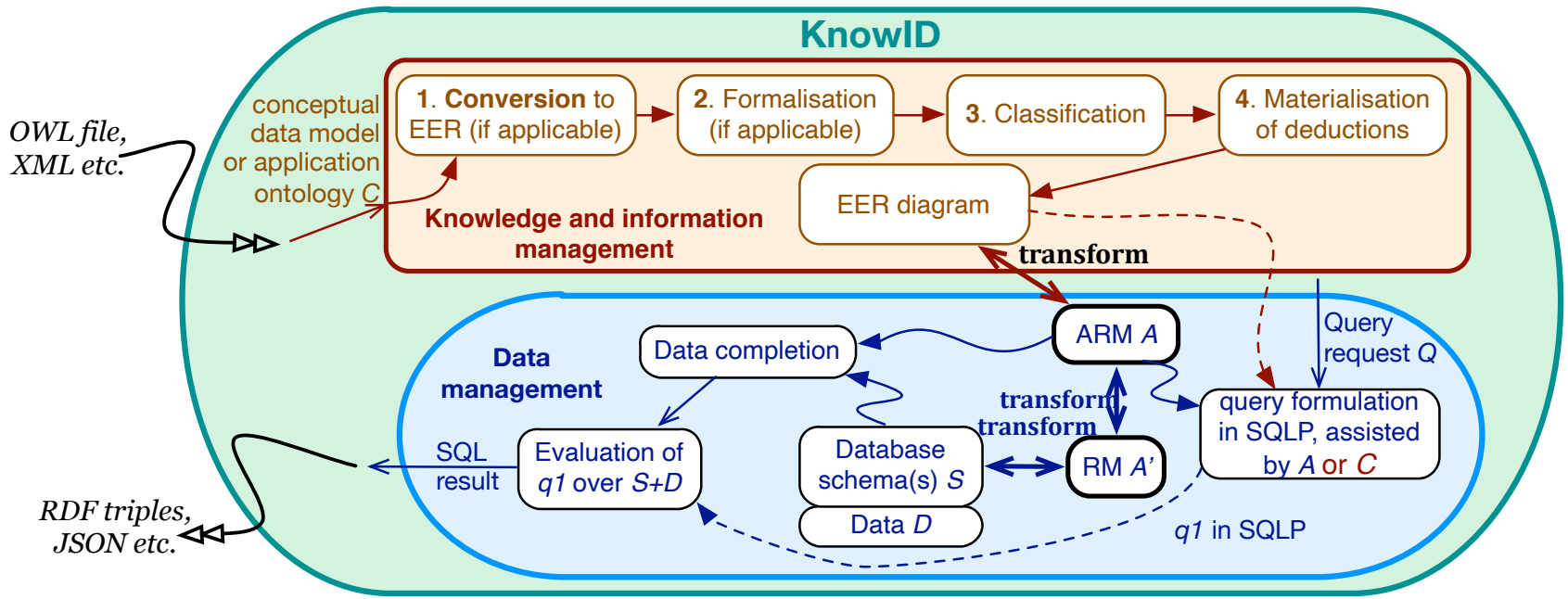


Knowledge-driven Information and Data access (KnowID)



Fillottrani, P.R., Keet, C.M. KnowID: An architecture for efficient Knowledge-driven Information and Data access. Data Intelligence, 2020, 2(4): 487-512.
Fillottrani, P.R., Jamieson, S., Keet, C.M. Connecting knowledge to data through transformations in KnowID: system description. Künstliche Intelligenz, 2020, 2020, 34, 373-379.

Knowledge-driven Information and Data access (KnowID)



Fillottrani, P.R., Keet, C.M. KnowID: An architecture for efficient Knowledge-driven Information and Data access. Data Intelligence, 2020, 2(4): 487-512.
Fillottrani, P.R., Jamieson, S., Keet, C.M. Connecting knowledge to data through transformations in KnowID: system description. Künstliche Intelligenz, 2020, 2020, 34, 373-379.

Knowledge-driven Information and Data access (KnowID)

- There's more on the 'knowledge and information management' module:
  - Swap between EER, UML, ORM
    [Keet and Fillottrani(2015), Fillottrani and Keet(2014)]
  - DL (OWL) with reasoner at the back-end
  - Tool: `crowd 2.0` (beta)
    `http://crowd.fi.uncoma.edu.ar:3335/`
    [Braun et al.(2020)]
  - More in the pipeline, such as integrating NOMSA for summarisation and modularisation of ontologies
- Querying with SQLP: SQLP requires less time for understanding and authoring queries, with no loss in accuracy [Ma et al.(2018)]
- Data Completion TBD

Outline

1 Context and motivation

2 Resolving conflicts

3 Implementation trade-offs

4 Conclusions

Recap and future work

- Foundational steps towards a framework that can deal in a systematic way with modelling conflicts through conflict resolution
- Notion of conflict set, with a data structure
- A first step towards a library of conflicts
- Some supporting tools for conflict resolution; more needed
- System design trade-offs in connecting the ontologies to the data; more needed

Main collaborators (on the works included in this talk)

- Collaborators: Diego Calvanese and Werner Nutt (FUB, Italy), Pablo Fillottrani (UNS, Argentina), Santi Garcia-Vellvé (URV, Spain), Rolf Grütter (WSL, Switzerland), Stephan Jamieson (UCT) Agnieszka Ławrynowicz (PUT, Poland), David Toman (UW, Canada)
- Current and former students: Zubeida Khan, Mandisa Baleni, Kieren Davies, Bradley Malgas, Brian McGeorge, Aashiq Parker, and Muhummad Patel, Giorgio Stefanoni

References I

Distributed ontology, model, and specification language, February 2018.
URL `http://www.omg.org/spec/DOL/`.

Germán Braun, Christian Gimenez, Laura Cecchi, and Pablo Fillottrani.
`crowd`: A Visual Tool for Involving Stakeholders into Ontology Engineering Tasks.
Künstliche Intelligenz, 2020.

D. Calvanese, P. Liuzzo, A. Mosca, J. Remesal, M. Rezk, and G. Rull.
Ontology-based data integration in epnet: Production and distribution of food during the roman empire.
Engineering Applications of Artificial Intelligence, 51:212–229, 2016.

Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao.
Ontop: Answering SPARQL queries over relational databases.
Semantic Web Journal, 8(3):471–487, 2017.

Aaron J. Cotnoir.
Anti-symmetry and non-extensional mereology.
The Philosophical Quarterly, 60(239):396–405, 2010.

P. R. Fillottrani and C. M. Keet.
Dimensions affecting representation styles in ontologies.
In *1st Iberoamerican conference on Knowledge Graphs and Semantic Web (KGSWC'19)*, volume 1029 of *CCIS*, pages 186–200. Springer, 2019.
24-28 June 2019, Villa Clara, Cuba.

References II

Pablo R. Fillottrani and C. Maria Keet.

Patterns for heterogeneous tbox mappings to bridge different modelling decisions.
In E. Blomqvist et al., editors, *Proceeding of the 14th Extended Semantic Web Conference (ESWC'17)*, volume 10249 of *LNCS*, pages 371–386. Springer, 2017.
30 May - 1 June 2017, Portoroz, Slovenia.

Pablo Rubén Fillottrani and C. Maria Keet.

Conceptual model interoperability: a metamodel-driven approach.
In A. Bikakis et al., editors, *Proceedings of the 8th International Web Rule Symposium (RuleML'14)*, volume 8620 of *LNCS*, pages 52–66. Springer, 2014.
August 18-20, 2014, Prague, Czech Republic.

S. Garcia-Vallvé, E. Guzman, M.A. Montero, and A. Romeu.

HGT-DB: a database of putative horizontally transferred genes in prokaryotic complete genomes.
Nucleic Acids Research, 31(1):187–189, 2003.

Yongqun He, Hong Yu, Edison Ong, Yang Wang, Yingtong Liu, Anthony Huffman, Hsin hui Huang, John Beverley, Asiyah Yu Lin, William D. Duncan, Sivaram Arabandi, Jiangan Xie, Junguk Hur, Xiaolin Yang, Luonan Chen, Gilbert S. Omenn, Brian Athey, and Barry Smith.

Cido: The community-based coronavirus infectious disease ontology.
In Janna Hastings and Frank Loebe, editors, *Proceedings of the 11th international Conference on Biomedical Ontologies*, volume 2807. CEUR-WS, 2020.

References III

C. Maria Keet and Pablo Rubén Fillottrani.

An ontology-driven unifying metamodel of UML Class Diagrams, EER, and ORM2.
Data & Knowledge Engineering, 98:30–53, 2015.
doi: 0.1016/j.datak.2015.07.004.

Weicong Ma, C. Maria Keet, Wayne Oldford, David Toman, and Grant Weddell.

The utility of the abstract relational model and attribute paths in sql.
In Catherine Faron Zucker, Chiara Ghidini, Amedeo Napoli, and Yannick Toussaint, editors, *Proceedings of the 21st International COnference on Knowledge Engineering and Knowledge Management (EKAW'18))*, volume 11313 of *LNAI*, pages 195–211. Springer, 2018.
ISBN 978-3-030-03667-6.
12-16 Nov. 2018, Nancy, France.

Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz.

OWL 2 Web Ontology Language Profiles.
W3C recommendation, W3C, 27 Oct. 2009.
http://www.w3.org/TR/owl2-profiles/.

Thank you!

Questions?

Some self-promotion:

- My textbook on ontology engineering (aimed at computer scientists)
- Free pdf (and slides and exercises) at `https://people.cs.uct.ac.za/~mkeet/OEbook/`
- Also available in paperback: