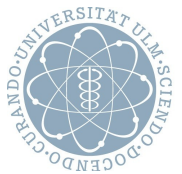
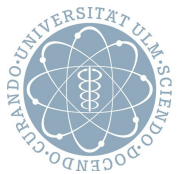


OWL 2
Profiles, Applications, and Modeling Issues

Pavel Klinov



This talk is about..

The OWL 2 Family

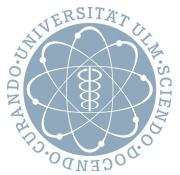
- QL/EL/RL/DL (+ Full, if you care)
- Target applications and use cases for each

Some modeling OWL issues

- A very incomplete and subjective list

This talk is NOT about

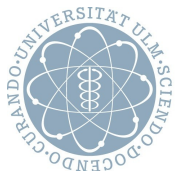
- Listing all features of OWL
- Deep logical foundations of OWL



So why Semantic Web needs OWL?

First, we've got RDF

- A simple graph language to express facts (LD)
- A simple data model + low-level data integration tools



So why Semantic Web needs OWL?

First, we've got RDF

- A simple graph language to express facts (LD)
- A simple data model + low-level data integration tools

No schema? But we have RDFS!

- A lightweight schema, good for simple vocabularies
- Some simple inferencing (transitivity of rdfs:subClassOf)

RDF(S) Not Quite Sufficient

Schemas are often just too weak

- Can say: hasFinger *rdfs:domain* Hand (*rdfs:range* Finger)
- Cannot say: Hand hasFinger *some* Finger (*only* Finger)

RDF(S) Not Quite Sufficient

Schemas are often just too weak

- Can say: hasFinger *rdfs:domain* Hand (*rdfs:range* Finger)
- Cannot say: Hand hasFinger *some* Finger (*only* Finger)

Reasoning is weak but very hard

- NP-hard even though no negation, no disjunction, etc.

RDF(S) Not Quite Sufficient

Schemas are often just too weak

- Can say: hasFinger *rdfs:domain* Hand (*rdfs:range* Finger)
- Cannot say: Hand hasFinger *some* Finger (*only* Finger)

Reasoning is weak but very hard

- NP-hard even though no negation, no disjunction, etc.

So we need a language(s) that:

- Provides adequate balance between expressivity and computational complexity
- “pay-as-you-go” behavior

RDF(S) Not Quite Sufficient

Schemas are often just too weak

- Can say: hasFinger *rdfs:domain* Hand (*rdfs:range* Finger)
- Cannot say: Hand hasFinger *some* Finger (*only* Finger)

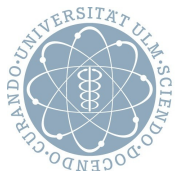
Reasoning is weak but very hard

- NP-hard even though no negation, no disjunction, etc.

So we need a language(s) that:

- Provides adequate balance between expressivity and computational complexity
- “pay-as-you-go” behavior

That language is called OWL 2



A Very Brief Dive Into OWL

OWL Basics

Conceptual model vs. data separation (familiar, right?)

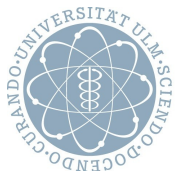
- TBox = schema (classes and properties)
- ABox = data or facts (this is close to RDF)

Ontology

- A collection of schema axioms and data axioms (facts)
- Axiom is a logical statement about the world

Semantics

- Purely declarative, model-theoretic
- Firmly based on Description Logics



A Simple Example (TBox, ABox)

Healthy beings are not dead

Healthy *DisjointWith* Dead

Every cat is either alive or dead

Cat *EquivalentTo* Alive *or* Dead

If you own something, you care for it

owns *SubPropertyOf* caresFor

Cat owner is someone who owns a cat

`CatOwner SubClassOf owns some Cat`

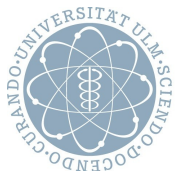
Schrödinger is a cat owner

schrödinger: CatOwner

Reasoning Support

Main inference problems:

- Consistency: is there a world in which all axioms are true?
- Entailment: what is implicitly true?
- TBox classification: compute the class hierarchy
- ABox realization: compute types for all objects



Reasoning Support

Main inference problems:

- Consistency: is there a world in which all axioms are true?
- Entailment: what is implicitly true?
- TBox classification: compute the class hierarchy
- ABox realization: compute types for all objects

Good and bad news:

- Good: it's decidable → you'll get your answers
- Bad: it may take a while
- Good: there are useful & tractable fragments

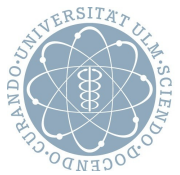
Data(types|properties)

OWL is a two-sorted language

- The world of logic
  - classes, properties, named objects

    Museum SubClassOf CulturalInstitution
- The world of data
  - strings, numbers, dates, etc.

    hermitage: hasAge “248”^^xsd:integer



Why the Separation?

Abstract part: developing theories about the world

- Modelers often cautious and pedantical
- Open World Assumption, no Unique Name Assumption

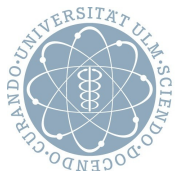
Why the Separation?

Abstract part: developing theories about the world

- Modelers often cautious and pedantical
- Open World Assumption, no Unique Name Assumption

Datatype part: re-using standard data theories

- “We don't need 43 half-baked integer ontologies!” ©
- Have excellent theories about numbers, etc.
- Know how to use them, how to compute with them



OWL 2 Profiles

(… or how one size does not fit all)

How Things Are Used to Be...

2004: the standardization of OWL 1

- OWL Lite

  Supposed to be simple for modelers and reasoners

- OWL DL

  Supposed to be as rich as possible but still decidable

- OWL Full

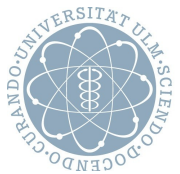
  Fully expressive (any RDF graph is an OWL Full ontology)

  Fully undecidable

… and Why It Wasn't Great …

OWL Lite was, in short, a failure

- *"...provide a minimal useful subset of language features, that are relatively straightforward for tool developers to support..."*
- Too vague, the subset is not well (logically) defined
  - no union but there's intersection and (implicit) negation!
  - no disjointness but there's intersection and owl:Nothing!
- Lots of unintuitive restrictions but still no guarantees (ExpTime)



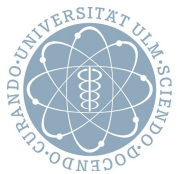
… and Why It Wasn't Great …

OWL Lite was, in short, a failure

- *"...provide a minimal useful subset of language features, that are relatively straightforward for tool developers to support..."*
- Too vague, the subset is not well (logically) defined
  - no union but there's intersection and (implicit) negation!
  - no disjointness but there's intersection and owl:Nothing!
- Lots of unintuitive restrictions but still no guarantees (ExpTime)

OWL DL misses some important features too

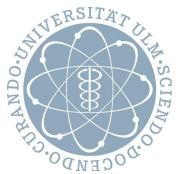
- Qualified cardinality restrictions
- Property chains: hasParent · hasBrother → hasUncle
- Data facets (e.g. integer intervals)



… and How It Was Fixed

2007: OWL 2 came out as a family of languages (profiles)

- RL: for those who love rules
- QL: for those who work with data
- EL: for those who work with large taxonomies
- DL: for those who need complex modeling
- Full: for those who *think* they need it



… and How It Was Fixed

2007: OWL 2 came out as a family of languages (profiles)

- RL: for those who love rules
- QL: for those who work with data
- EL: for those who work with large taxonomies
- DL: for those who need complex modeling
- Full: for those who *think* they need it

The principal difference from Lite/DL/Full:

Each profile is based on a carefully designed logic(s), all restrictions justified for computational guarantees

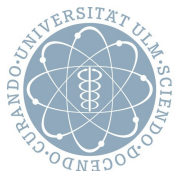
OWL 2 RL

Fragment for which rule-based reasoning is complete

- asymmetric restrictions on left/right sides of subclass axioms so they behave like rule implications

  Museum or Theater → ≤ 1 hasLocation (rule-ish syntax)

- Allows for such *dangerous* constructs as:
  - Union (left)
  - Negation (right)
  - Universal restriction (right)
- But no existentials on the right (rules aren't generative)



OWL 2 RL Use Cases

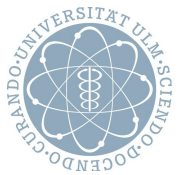
Already have a good rule engine (e.g. in a DB)

Re-modeling of a rule-based system

Convenient for RDF-based reasoning

Example implementations:

- Oracle Prime
- OWLIM
- Jena
- Stardog (no rules here though!)



OWL 2 QL

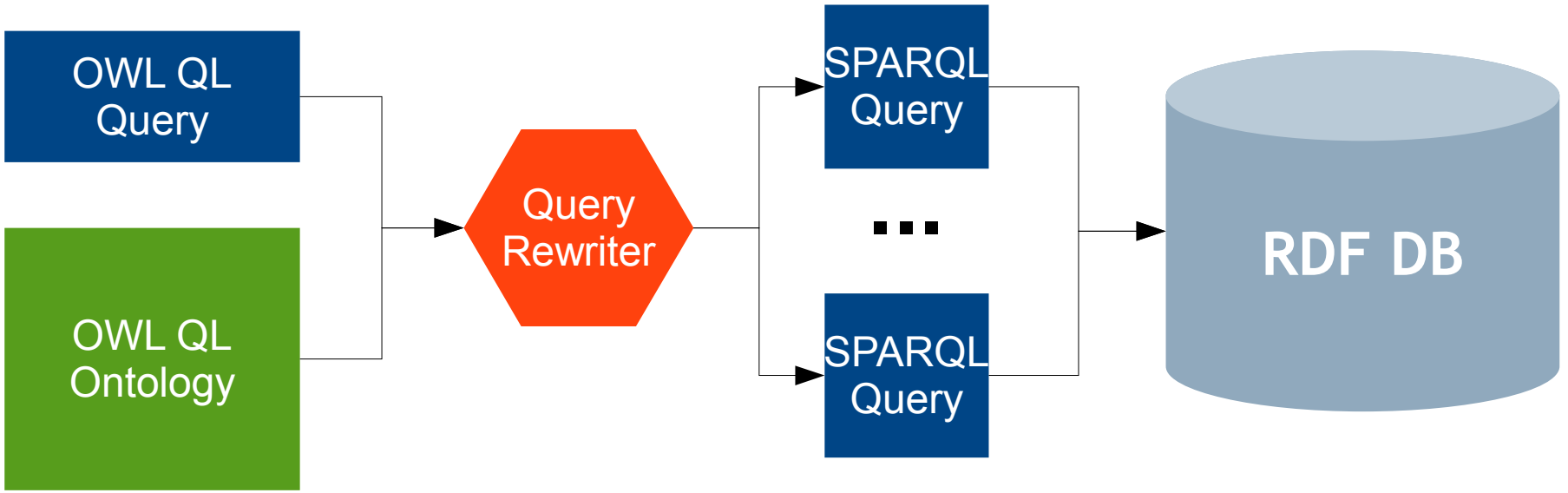
Mostly about querying in data-rich apps

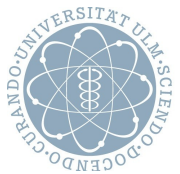
- Very low data complexity
- FO-rewritability: query translatable to SPARQL or SQL

OWL 2 QL

Mostly about querying in data-rich apps

- Very low data complexity
- FO-rewritability: query translatable to SPARQL or SQL





OWL 2 QL Use Cases

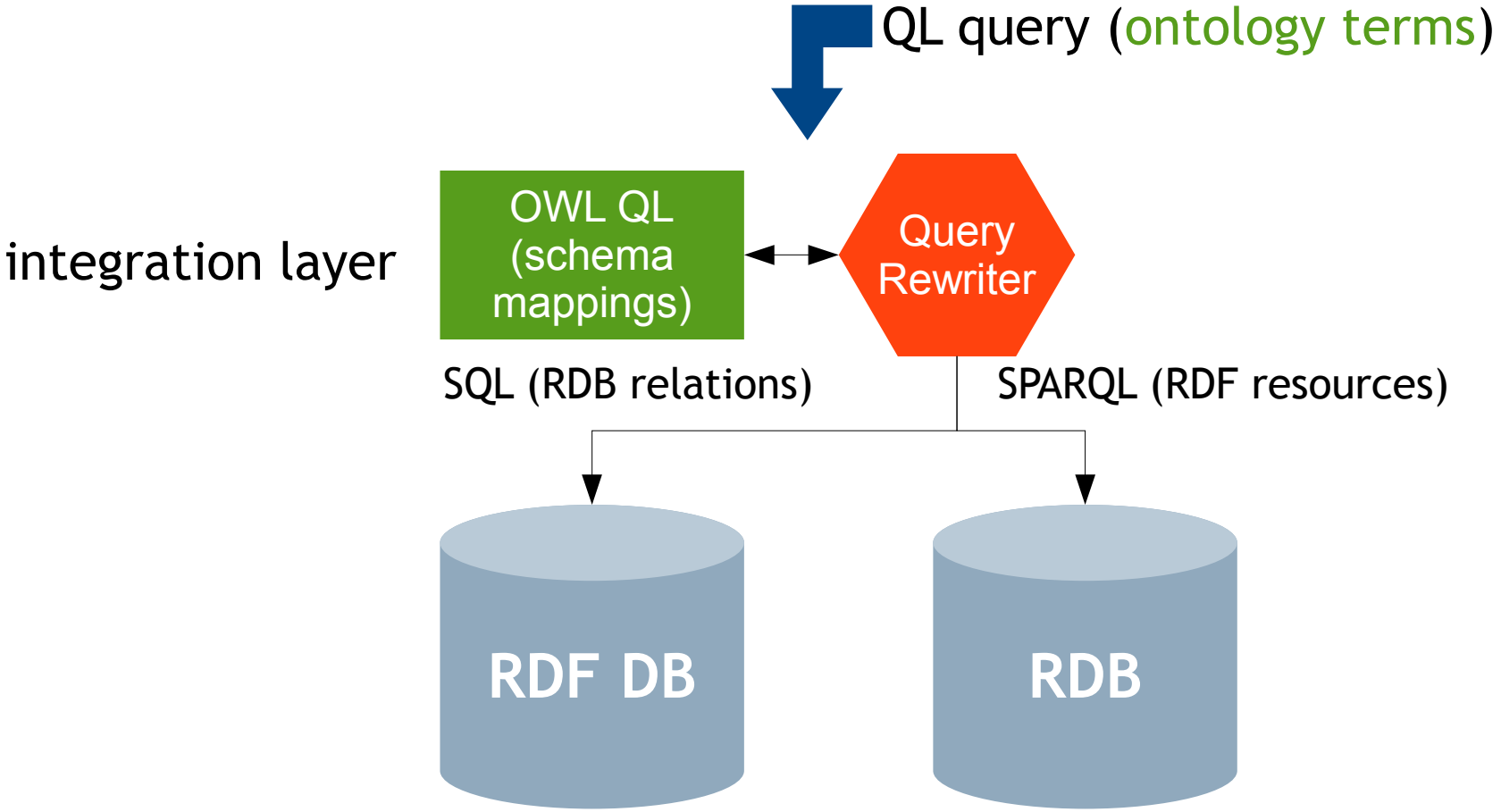
Querying, querying, and more querying!

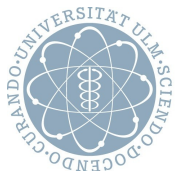
Ontology Based Data Access and data integration

OWL 2 QL Use Cases

Querying, querying, and more querying!

Ontology Based Data Access and data integration





OWL 2 EL

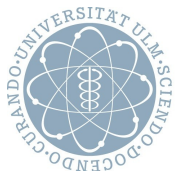
Many scientific ontologies are lightweight but large

- Medical terminologies (SNOMED CT, >300K terms)
- Chemical and bio ontologies (GO/PO/SO, etc.)

Need fast and scalable schema management

- Consistency and coherence
- Classification (also incremental)
- Debugging and explanations

OWL 2 EL is PTime for the main tasks



OWL 2 EL Projects

ELK: the world-fastest EL reasoner

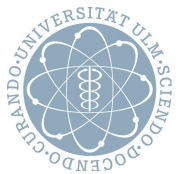
- Massively parallel
- Classifies SNOMED CT in ~10s on a $1K laptop

Snow Owl: authoring of clinical terminologies

- Browsing, searching, reasoning, reporting, etc.
- Used for SNOMED CT, ELK under the hood

Virtual Fly Brain: look into *Drosophila* brain!

- 3D viewer, interactive explorer, EL ontology behind that



OWL 2 EL Projects

ELK: the world-fastest EL reasoner

- Massively parallel
- Classifies SNOMED CT in ~10s on a $1K laptop

Snow Owl: authoring of clinical terminologies

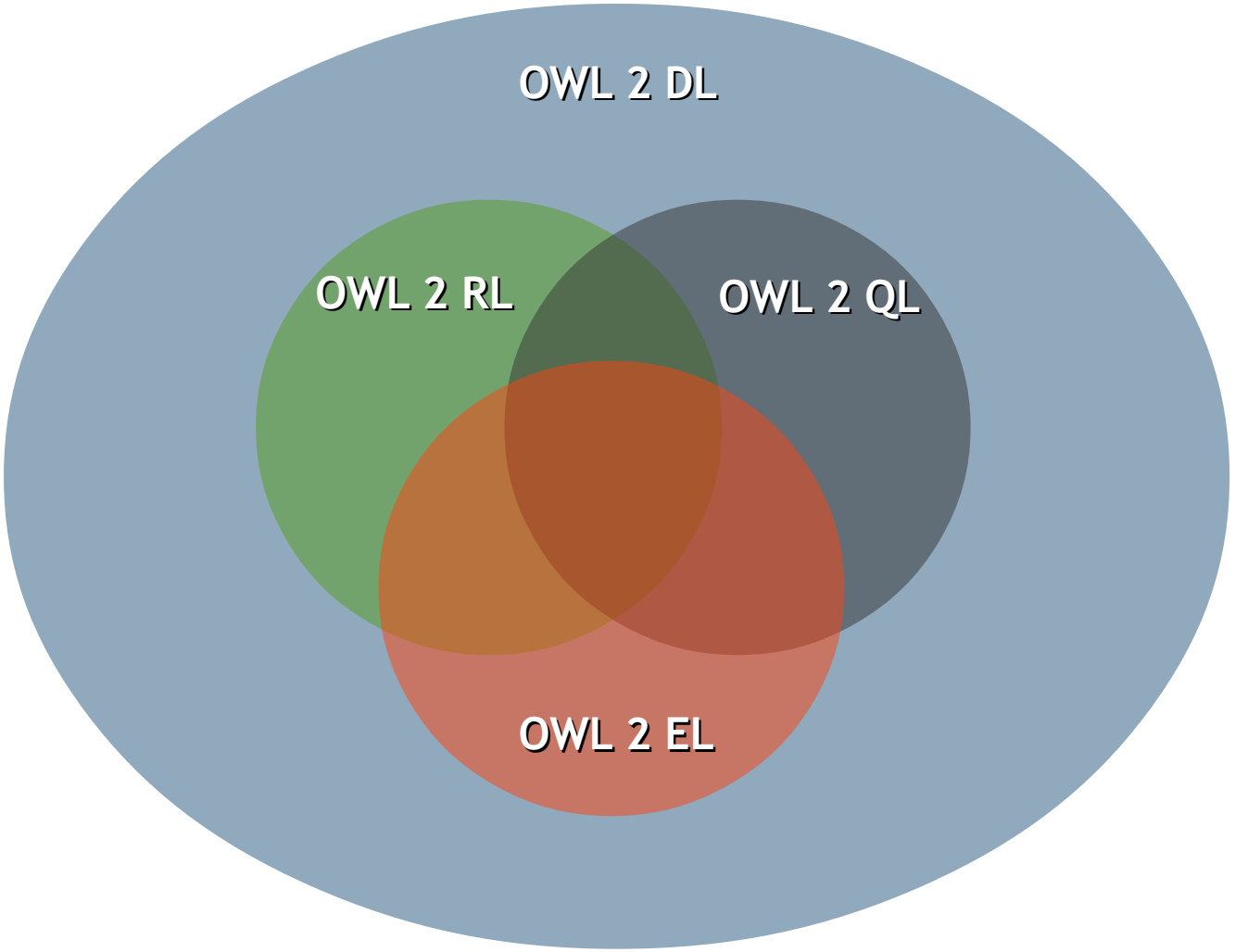
- Browsing, searching, reasoning, reporting, etc.
- Used for SNOMED CT, ELK under the hood

Virtual Fly Brain: look into *Drosophila* brain!

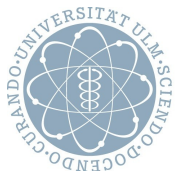
- 3D viewer, interactive explorer, EL ontology behind that

Need to grow beyond 100K terms? Consider OWL 2 EL

Summary on OWL 2 Profiles



Combining any two of the tractable profiles results in a intractable language



Some Modeling Issues

(a very-very incomplete list)

OWL Can't Represent Everything

It can't represent what FOL can't (naturally) represent

- Temporal knowledge
- Quantitative uncertainty
- Higher-order knowledge

It has troubles with knowledge beyond the 2-var FOL

- N-ary relationships of sorts

OWL 2 DL has limitations to ensure decidability

N-ary is problematic

OWL (and RDF) are 2-variable logics

- Schema restrictions and properties are binary

  Hand hasFinger some Finger

  Museum hasLocation some Location

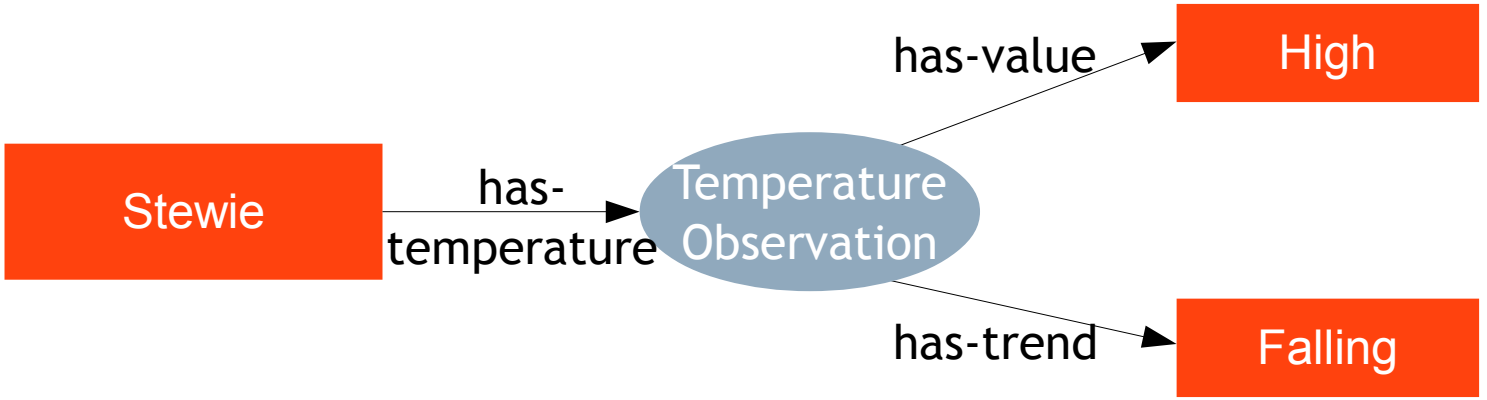
- Facts are binary

How do we say:

- Stewie has a high (but falling) temperature?
- Megan bought a book A from store B?
- Lois visited LAX, JFK, and BOS on a single trip?

Workarounds

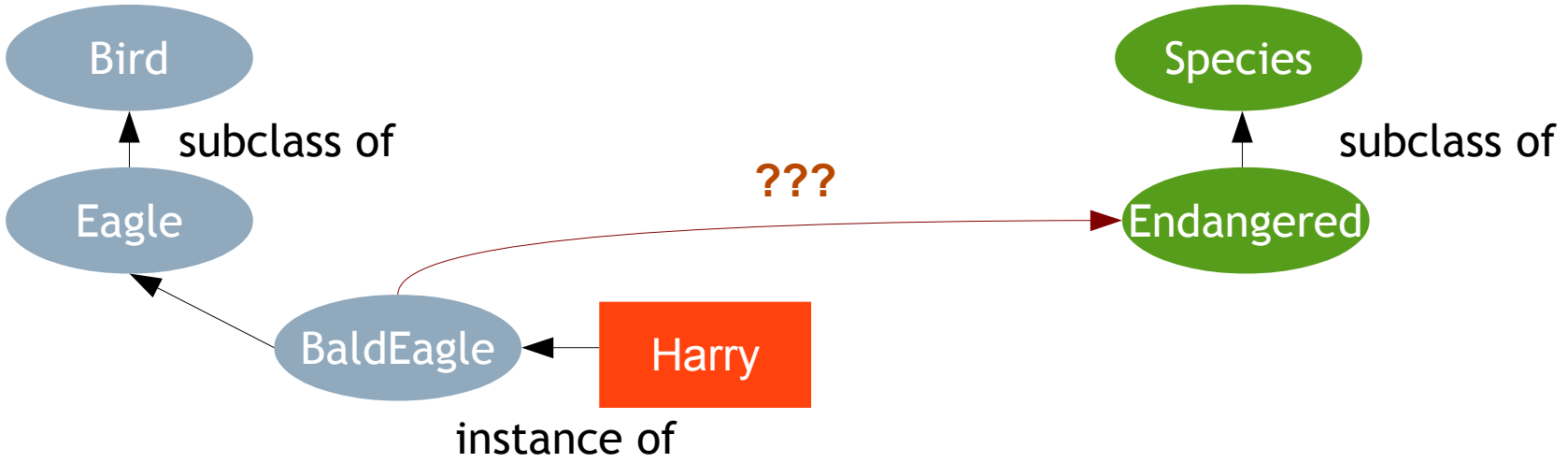
Via classes that work like reified properties



In OWL 2 property chains help

has-temperature ∘ has-trend → has-temperature-trend

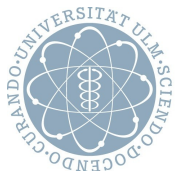
Metamodeling



BaldEagle subclass of Endangered would imply Harry is a Species

Species and Endangered are meta-classes

OWL 2 is first-order so meta-modeling in the higher-order sense is tricky



Metamodeling

Some limited support is available:

- Annotations (isEndangered could be semantic-free)
- Punning (BaldEagle-as-class vs. BaldEagle-as-instance)

Powerful solutions exist but lacking tool support:

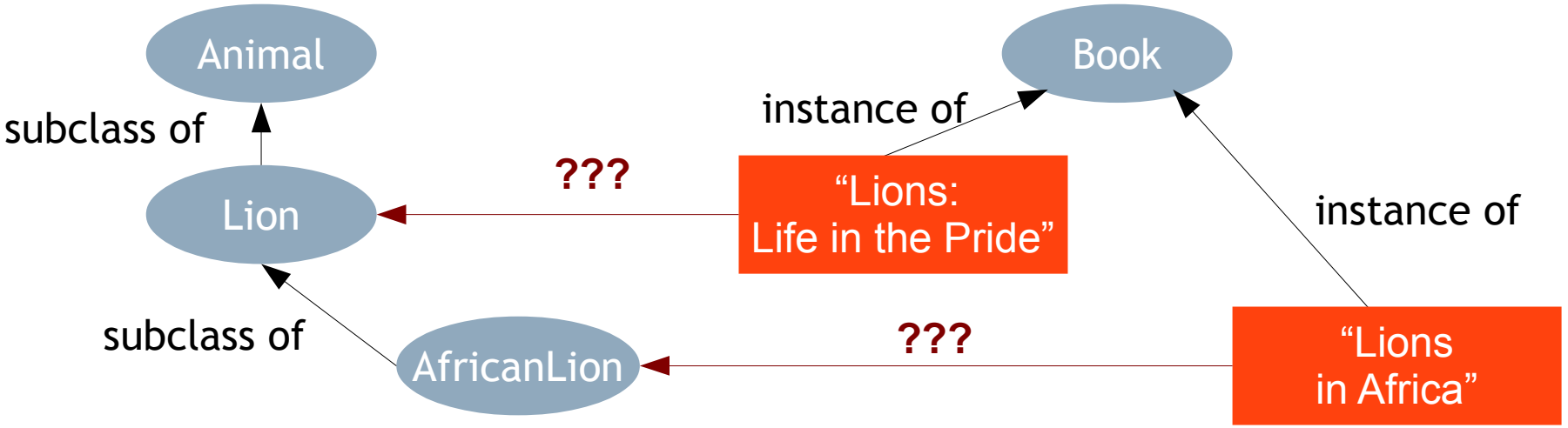
- Alternative semantics (first-order or higher-order)
- Complex encoding within OWL 2

What's often done:

- Parallel hierarchy of meta-classes and extra-logical linking
- OWL Full

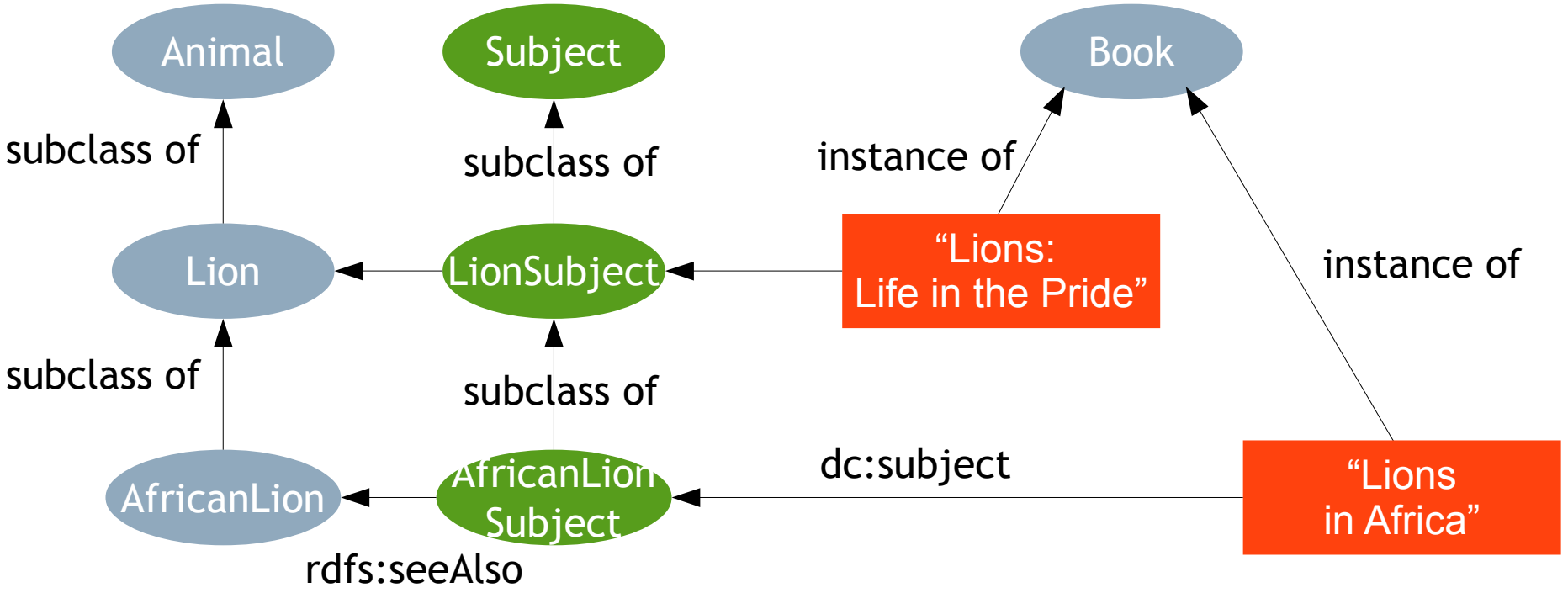
Classes as Property Values

Another example of meta-modeling



The books are not about some specific lions but about (African)Lion as a class

Workarounds



Obvious maintenance overhead

Or (you guessed it!) OWL Full

Time

OWL doesn't support temporal concepts:

- Class of people who were employed *before* the crisis
- Everyone will be *eventually* dead
- A *was* true, *will be* true, *will be true after* B... etc.

Time

OWL doesn't support temporal concepts:

- Class of people who were employed *before* the crisis
- Everyone will be *eventually* dead
- A *was* true, *will be* true, *will be true after* B... etc.

Available out-of-the-box? XSD datatypes

- xsd:date, xsd:dateTime, xsd:time
- Facts are expressible:

  meeting: start-time “2002-09-24-06:00”^^xsd:time

Time: “Solutions”

OWL Time (formerly DAML Time)

- Ontology on top of the existing logical model

  Process *SubClassOf* hasDuration *some* time:Interval
- Helps standardize temporal vocabulary
- Very limited temporal reasoning

Various extensions based on temporal logics

SWRL built-ins

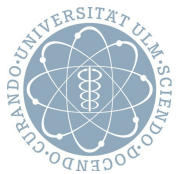
Uncertainty

Similar to Time: first-order logical model provides very limited means to capture uncertainty:

- Disjunction
- Open World Assumption, no Unique Name Assumption

But what about:

- Statistics (disease incidence, risk factors, etc.)
- All sorts of vagueness (YoungPerson)
- Measurement errors



Not Even “Solutions”

Would an “OWL Uncertainty” ontology be useful?

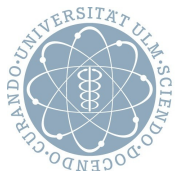
Logical extensions are difficult

- computationally
- cognitively (frequentist or subjective probabilities?)

Disparate modeling dominates:

- OWL for terms, Bayesian (Markov) for uncertainty
- OWL for terms, statistical model for uncertainty

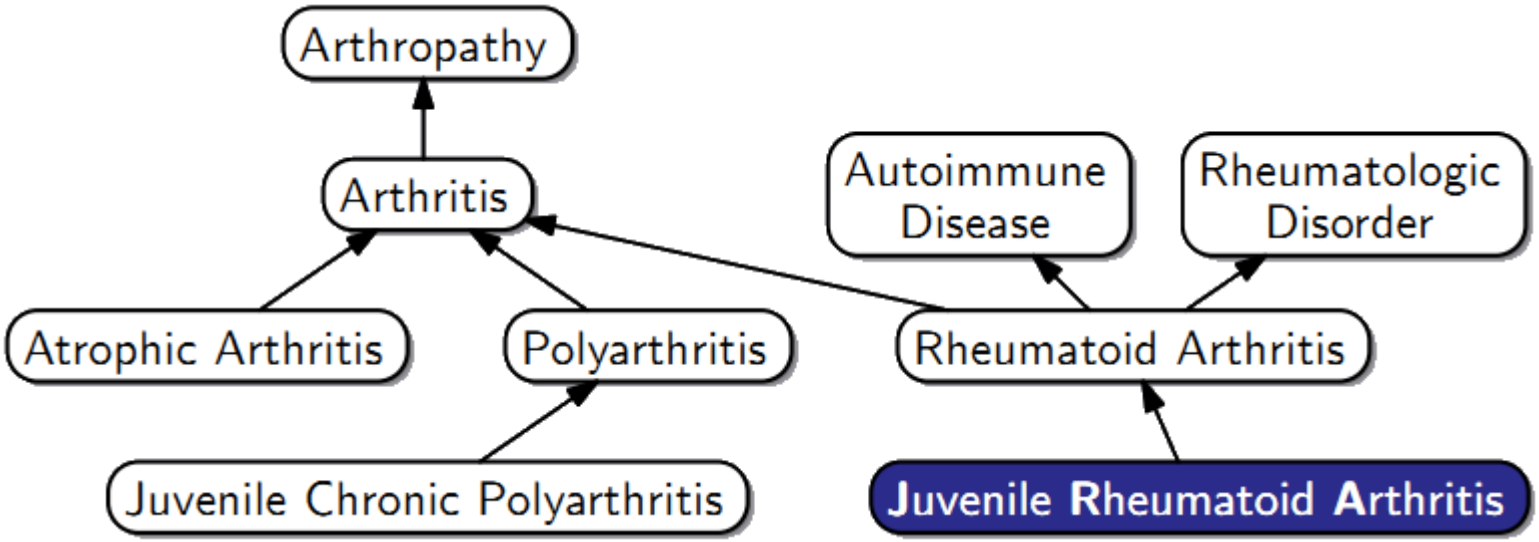
  (Example: Breast Cancer Risk Calculator)



A very quick note on ontology reuse and modularity in OWL

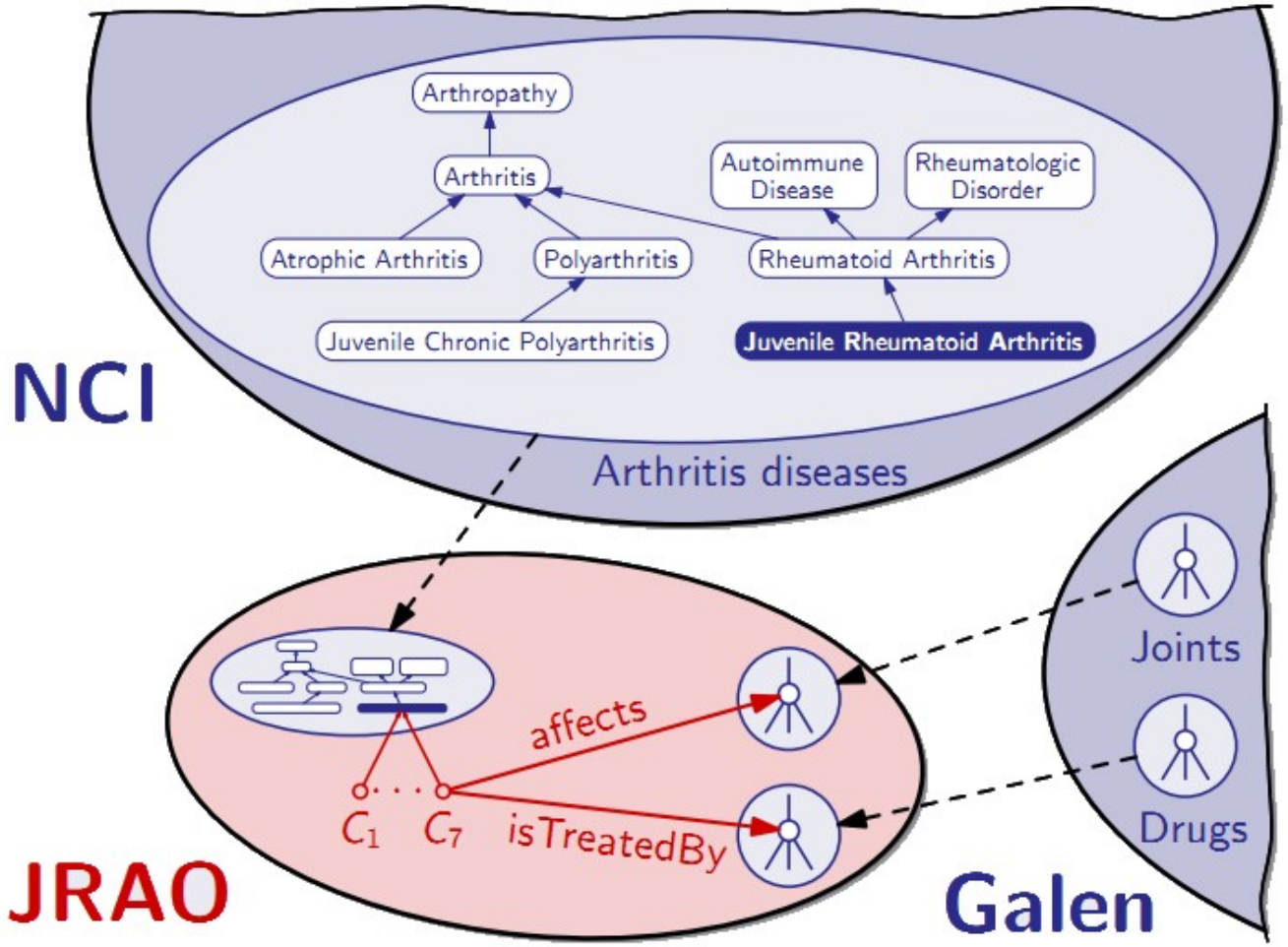
Ontology Re-use

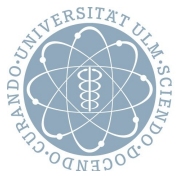
Want to use terms defined in existing domain ontologies, like the NCI Thesaurus:



Ontology Re-use

Want to use terms defined in existing domain ontologies, like the NCI Thesaurus:

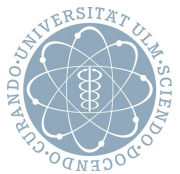




A Note on Module Extraction in OWL

But you want to be:

- Economic: owl:importing NCI will kill your editor, your reasoner, and, possibly, other apps
- Complete: losing relevant knowledge is never good



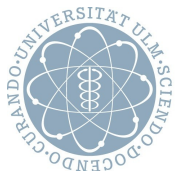
A Note on Module Extraction in OWL

But you want to be:

- Economic: owl:importing NCI will kill your editor, your reasoner, and/or other apps
- Complete: losing relevant knowledge is never good

Your choices:

- Be ad hoc. Almost guaranteed to violate the above
- Use logic-based modularity
  - Guarantees completeness and approximates economy
  - via a Web-based tool: owl.cs.manchester.ac.uk/modularity/
  - Inside your app (it's a part of the OWL API)



Questions!