http://www.unik.no/drift/undervisning.html

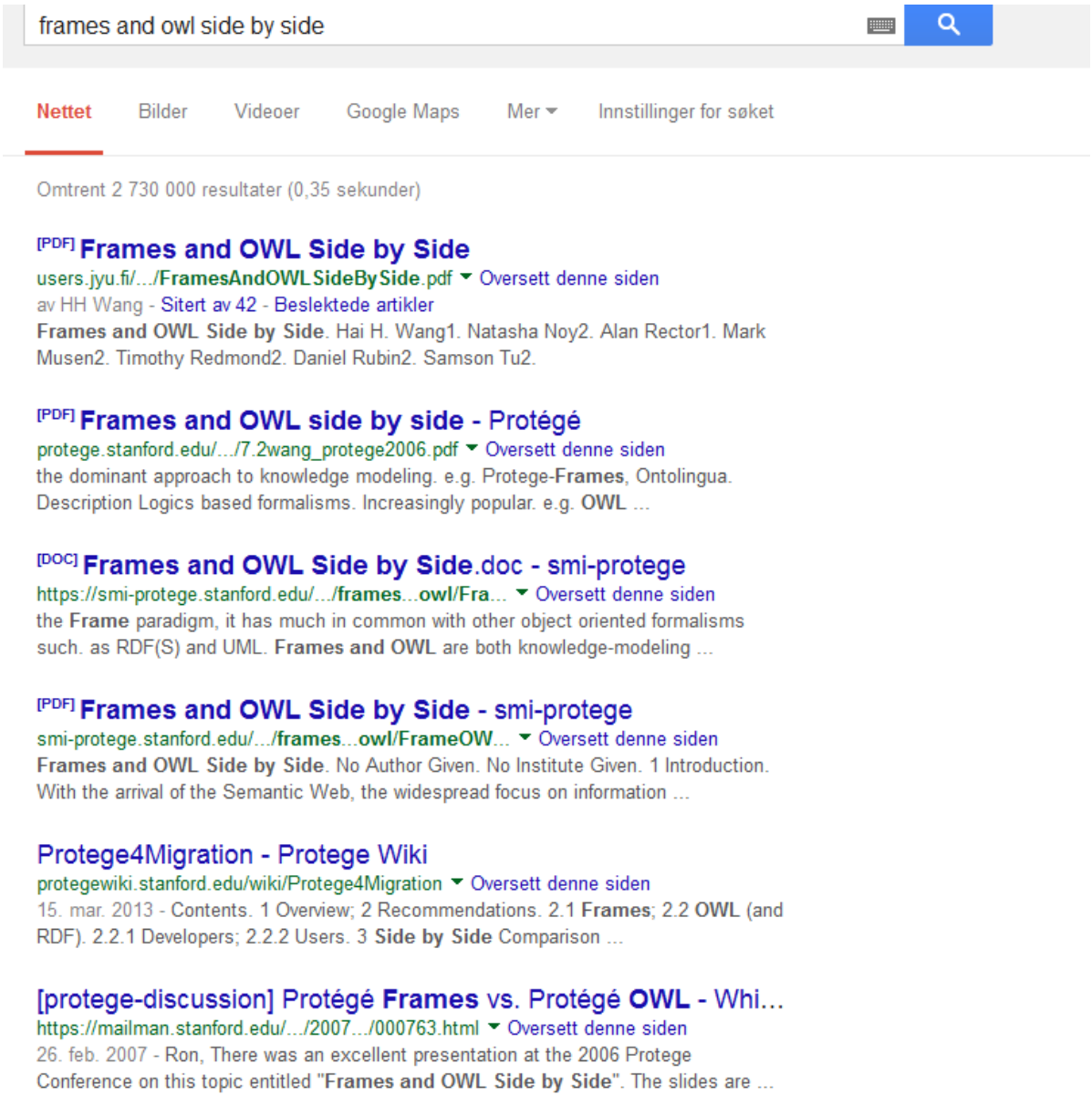
- Comparison Protégé 3.x ⟷ Protégé 4.x

http://cwi.unik.no/index.php?title=Policies_and_Semantic_Web_Rules_in_Practise&action=slide

- Internal vs external rules

- rules

- API : Programming interface



http://protege.stanford.edu/conference/2006/submissions/slides/7.2wang_protege2006.pdf

2.1 Semantics and implication

The following represents some of the major differences between Frames and OWL:

Unique name assumption In Frames, if two objects have different names, they are assumed to be different, unless explicitly stated otherwise. In OWL, no such assumption is made.

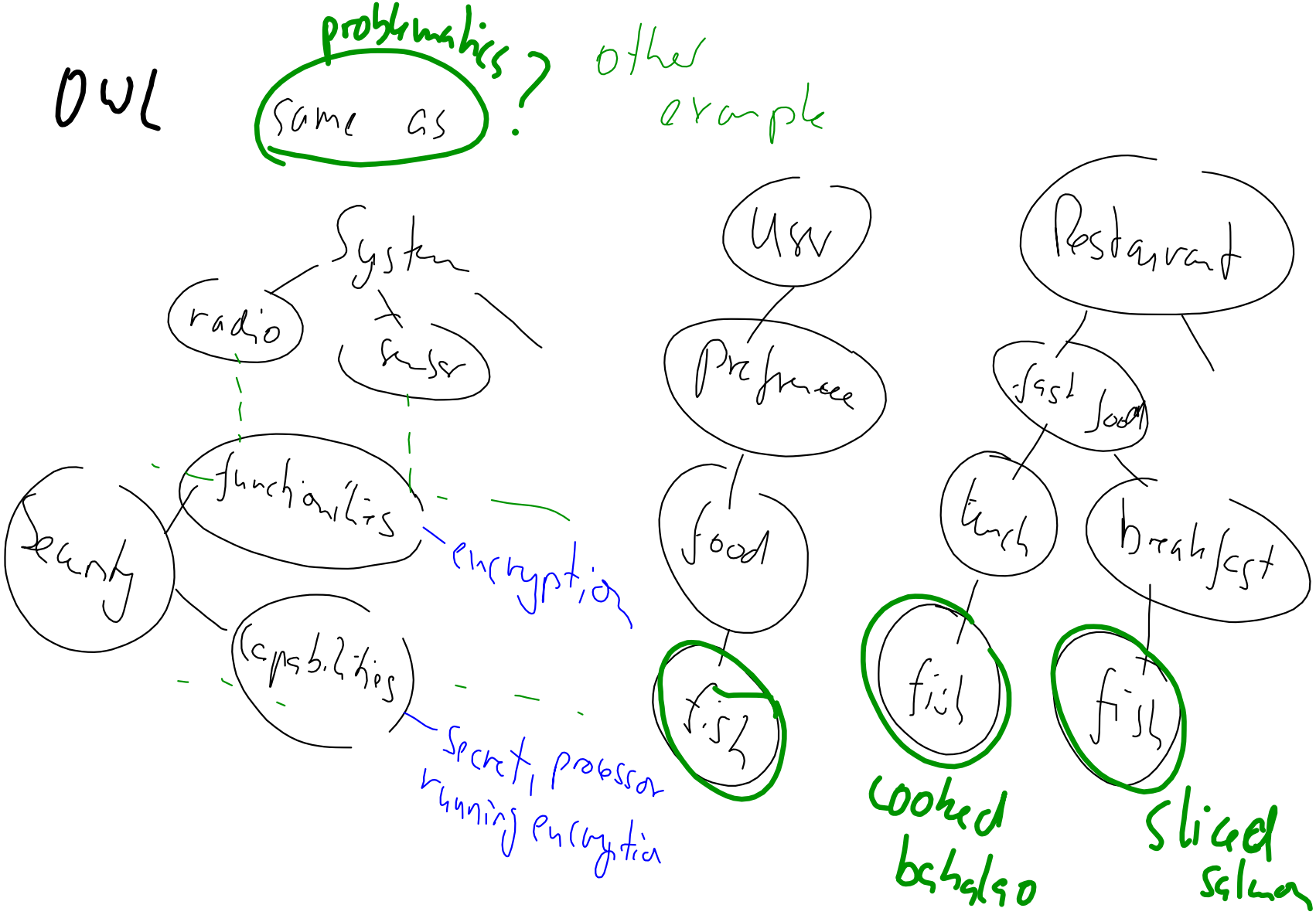
Closed World Assumption vs Open World Assumption In Frames, everything is prohibited until it is permitted; in OWL, everything is permitted until it is prohibited. Nothing can be entered into a Frames KB until there is a place for it in the corresponding template. Anything can be entered into an OWL KB unless it violates one of the constraints.

Single vs Multiple models A Frames ontology only has one model which is the minimal model that satisfies each of the assertions of the Frames ontology. This means that models for a Frames ontology can only contain instances that are explicitly specified. In general an OWL ontology will have many models consisting of all possible interpretations that satisfy each of the assertions in the OWL ontology.

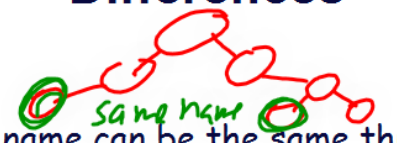
These differences have direct implications on how inference is performed, and therefore, implications for the use of modeling constructs in the two paradigms. The foremost of these differences are:

Assertion vs Classification In Frames, defining facets on a slot at a class, or defining a constraint on a slot at the top level, makes a statement about all instances of that class (except for possible exceptions provided by default values), describing *necessary* conditions for instances of that class. In OWL, there are effectively two kinds of statements about classes: a) those that, as in Frames are true of all individuals in a class, and b) those that are collectively *necessary and sufficient* to recognize members of a class. A OWL classifier can use the sufficient conditions to infer which classes are subclasses of the defined class. There is no equivalent feature in Frames.

Constraint checking vs Consistency checking The same reasoner that checks the classification also checks that an OWL KB is consistent. The classifier tries to build a model that satisfies all the axioms in the ontology. If no such model can be built, the ontology is inconsistent. When building a model that satisfies all the assertions, a classifier may assign new types to ontology instances, in addition to the types explicitly asserted by a modeler. By contrast, a Frames reasoner checks if the constraints are satified by the property values on instances; if they are not, the instance is non-conformant. In Frames, the inference cannot assign a new type to an instance.


OWL
problematics?
same as
other example
System
radio
functionalities
security
capabilities
encryption
secret, processor
running encryption
User
Preferences
food
fish
Restaurant
fast food
lunch
breakfast
fish
fish
cooked
sliced

Differences



OWL 1.0 (Protege 3.x)

- Objects with same name can be the same thing
- Open World Assumption (everything is ALLOWED until otherwise specified)
- Can contain instances of all kinds of interpretations
- Statements about classes can be both must and necessary
- Different reasoner
- SWRL Query language

Frames, OWL 2.0 (Protege 4.x)

- Objects with same name are assumed to be different
- Closed World Assumption (everything is PROHIBITED unitil otherwise specified)
- Can only contain instances which are explicitly specified
- Statements about classes goes for all children (only must)
- Logical query language
- More property characteristics

Policies and Semantic Web Rules in Practise

Policies and Semantic Web Rules in Practise

UNIK4710/UNIK9710 5 / 9

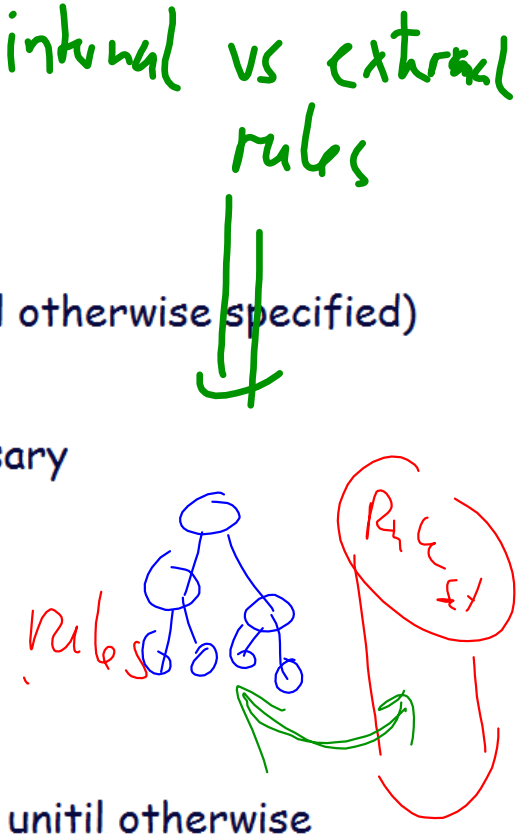
Differences

OWL 1.0 (Protege 3.x)

- Objects with same name can be the same thing
- Open World Assumption (everything is ALLOWED until otherwise specified)
- Can contain instances of all kinds of interpretations
- Statements about classes can be both must and necessary
- Different reasoner
- SWRL Query language

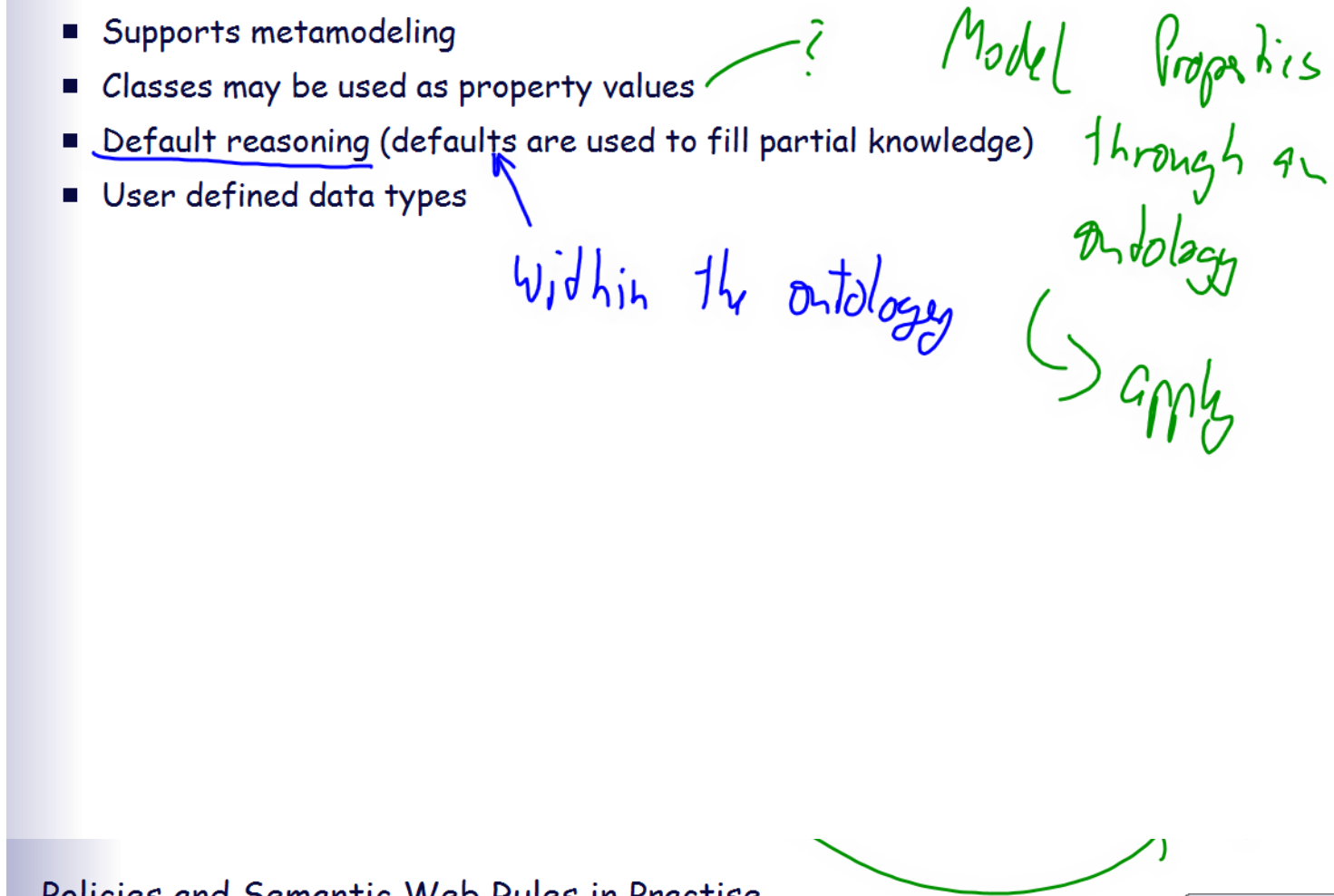
Frames, OWL 2.0 (Protege 4.x)

- Objects with same name are assumed to be different
- Closed World Assumption (everything is PROHIBITED unitil otherwise specified)
- Can only contain instances which are explicitly specified
- Statements about classes goes for all children (only must)
- Logical query language
- More property characteristics

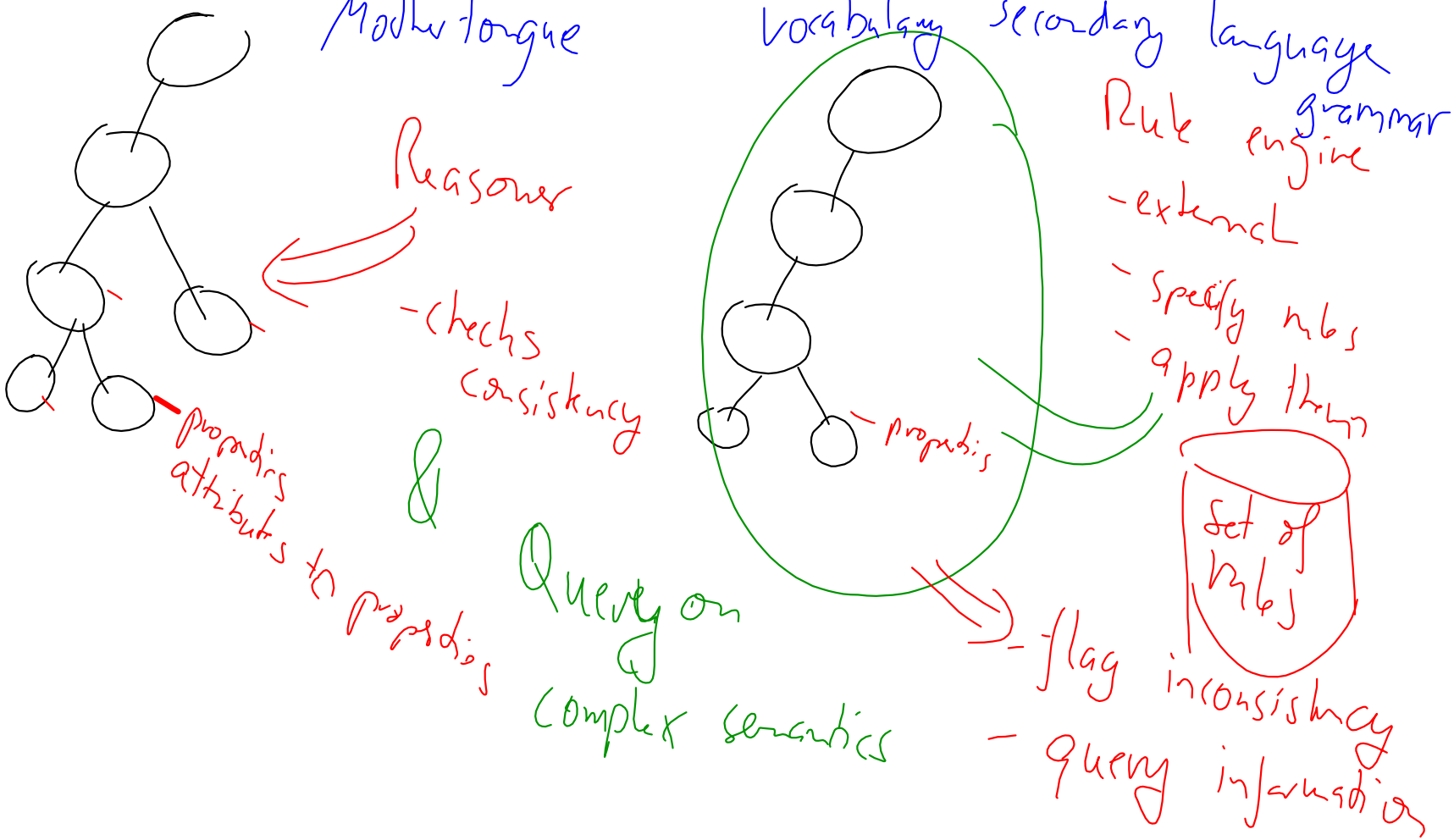


Strenghts Of OWL 2.0/Protége 4.x

- Supports metamodeling
- Classes may be used as property values
- Default reasoning (defaults are used to fill partial knowledge)
- User defined data types



Internal vs External rules



Advantages / Disadvantages

internal

+ consistency

josef@unik ?

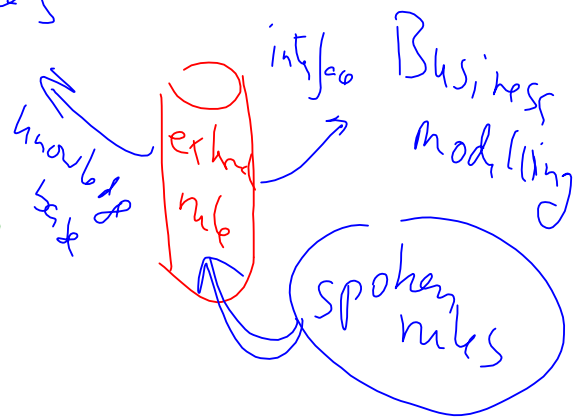
+ any change of knowledge base is reflected immediately

knowledge
- properties
- attributes

external rules

+ expressivity

+ interoperability



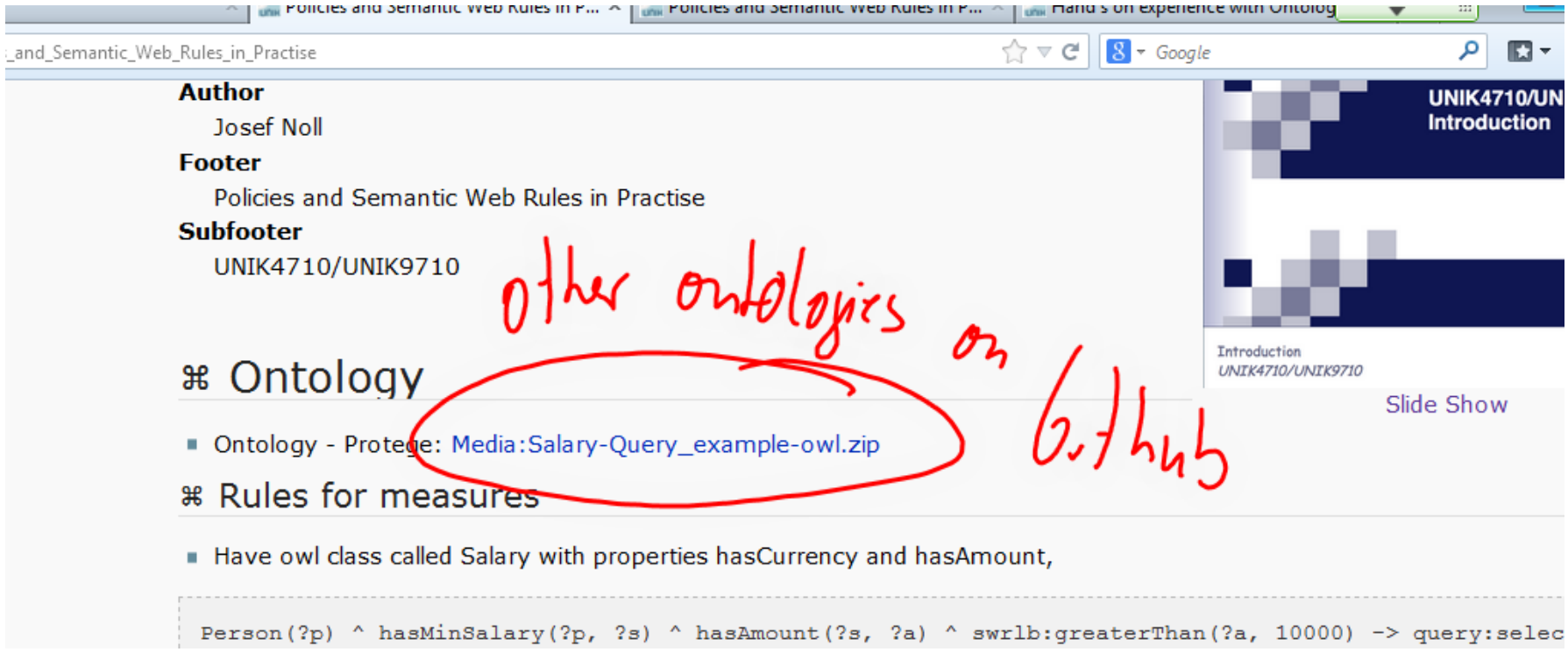
an email address has a "@" in it

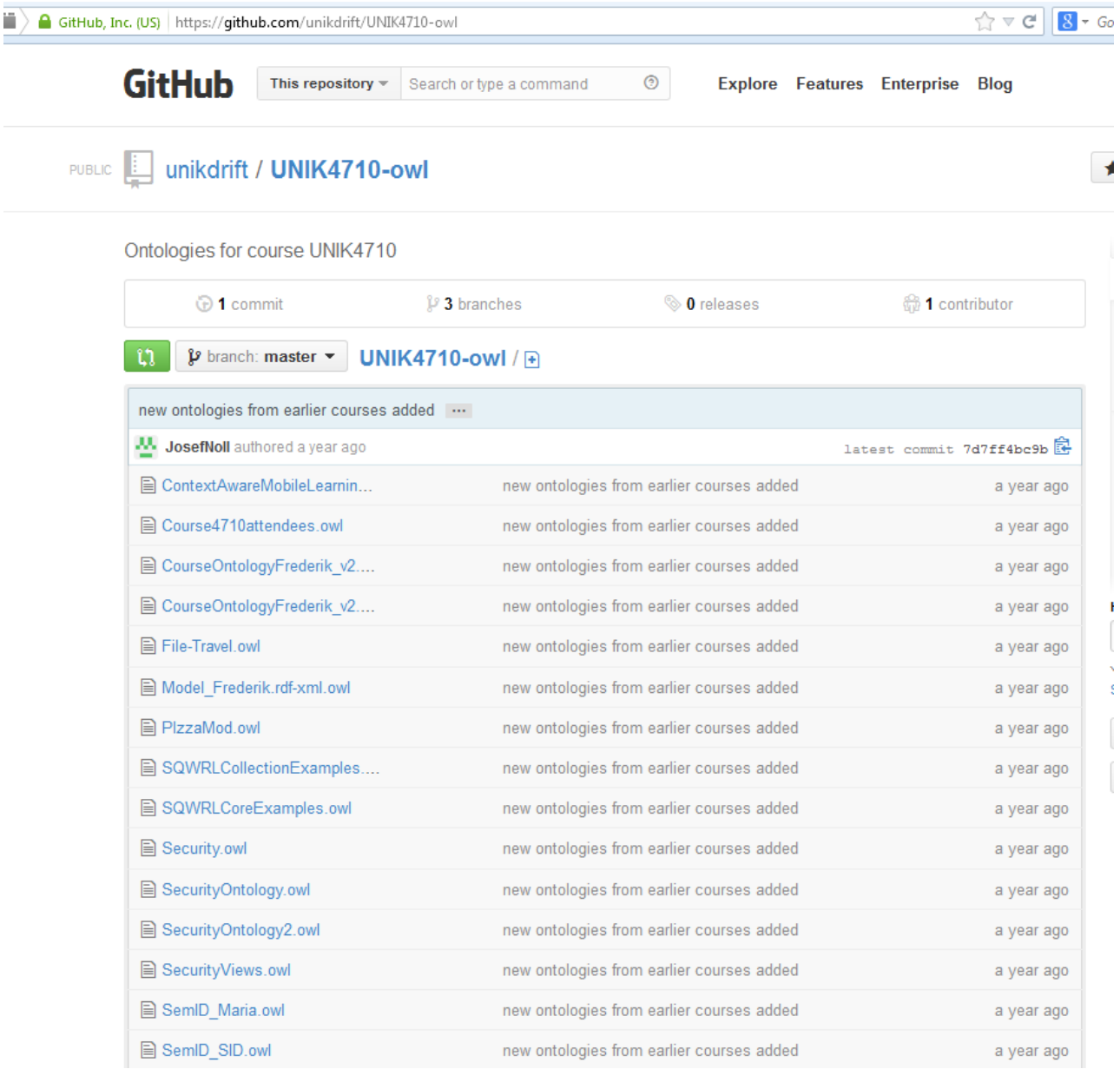
or external Query

Questions

ref to Jim Hendler book
google plus link

- when to use classes or properties?
- Graphviz to visualize individuals
- Graphviz installation
  dot for Windows
- how many properties should I define?
- extend standard properties by own ones
  data & object properties


_and_Semantic_Web_Rules_in_Practise
Google
Author
Josef Noll
Footer
Policies and Semantic Web Rules in Practise
Subfooter
UNIK4710/UNIK9710
other ontologies on github
UNIK4710/UN
Introduction
Introduction
UNIK4710/UNIK9710
Slide Show
⌘ Ontology
Ontology - Protege: Media:Salary-Query_example-owl.zip
⌘ Rules for measures
Have owl class called Salary with properties hasCurrency and hasAmount,
Person(?p) ^ hasMinSalary(?p, ?s) ^ hasAmount(?s, ?a) ^ swrlb:greaterThan(?a, 10000) -> query:selec


GitHub, Inc. (US) https://github.com/unikdrift/UNIK4710-owl
GitHub
This repository
Search or type a command
Explore
Features
Enterprise
Blog
PUBLIC
unikdrift / UNIK4710-owl
Ontologies for course UNIK4710
1 commit
3 branches
0 releases
1 contributor
branch: master
UNIK4710-owl /
new ontologies from earlier courses added
JosefNoll authored a year ago
latest commit 7d7ff4bc9b
ContextAwareMobileLearnin... new ontologies from earlier courses added a year ago
Course4710attendees.owl new ontologies from earlier courses added a year ago
CourseOntologyFrederik_v2.... new ontologies from earlier courses added a year ago
CourseOntologyFrederik_v2.... new ontologies from earlier courses added a year ago
File-Travel.owl new ontologies from earlier courses added a year ago
Model_Frederik.rdf-xml.owl new ontologies from earlier courses added a year ago
PIzzaMod.owl new ontologies from earlier courses added a year ago
SQWRLCollectionExamples.... new ontologies from earlier courses added a year ago
SQWRLCoreExamples.owl new ontologies from earlier courses added a year ago
Security.owl new ontologies from earlier courses added a year ago
SecurityOntology.owl new ontologies from earlier courses added a year ago
SecurityOntology2.owl new ontologies from earlier courses added a year ago
SecurityViews.owl new ontologies from earlier courses added a year ago
SemID_Maria.owl new ontologies from earlier courses added a year ago
SemID_SID.owl new ontologies from earlier courses added a year ago

Example of a rule

list of Adults

⌘ Example

```
Person(?p) ^ hasAge(?p, ?age) ^ swrlb:greaterThan(?age, 17) -> Adult(?p)
```

Can be easily turned in to a SQWRL query by replacing its consequent:

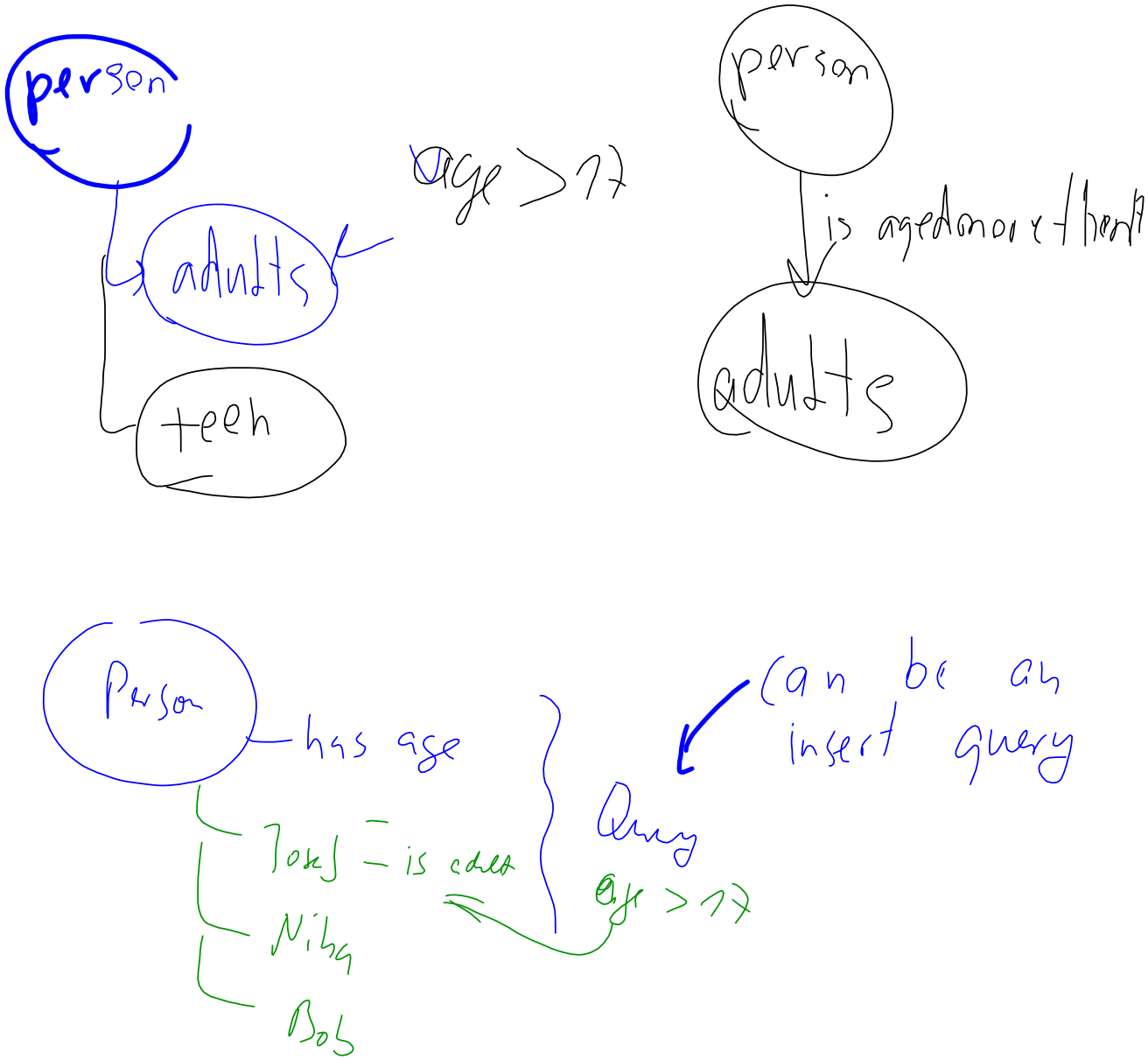
```
Person(?p) ^ hasAge(?p, ?age) ^ swrlb:greaterThan(?age, 17) -> sqwrl:select(?p, ?age)
```

Another alternative is to simply add a sqwrl:select clause at the end of a rule:

```
Person(?p) ^ hasAge(?p, ?age) ^ swrlb:greaterThan(?age, 17) -> Adult(?p) ^ sqwrl:select(?p, ?age)
```

representation in a query ⇝ returns tuples

construct { ?p a Adult }
where { ?p Age ?a .
filter(?a > 17) }


person
adults
teen
age > 17
person
is age dynamic threshold
adults
Person
has age
Tom - is adult
Nils
Bob
Query
age > 17
can be an insert query

a) Protégé 4.x
↳ enhance ontology
→ work with Queries
↳ query examples by Andreas

b) Protege 3.+
↳ simple ontology
→ extend SWRL rules to query and enhance

Reasoning in OWL 2 (UNIT 5?10)