+ extra information

+ classes
(sub-) classes
+ properties
(+ instances)

+ Swoop - error explanation - debug

- infer subclasses

- Pellet - description < restrictions / rules

- SWRL in Pellets

Protege 3 & SWRL

* install SWRL tab
* install JESS engine

→ Run Rules & write back

Protege 4 & SWRL

- Rules through: Window → Views → Ont. views → Rules
- text editor
- Run rules: Pellet
- no direct "method for "write back"

Protege API

- Java load
- SWRL reasoning
- SQWRL
- save back

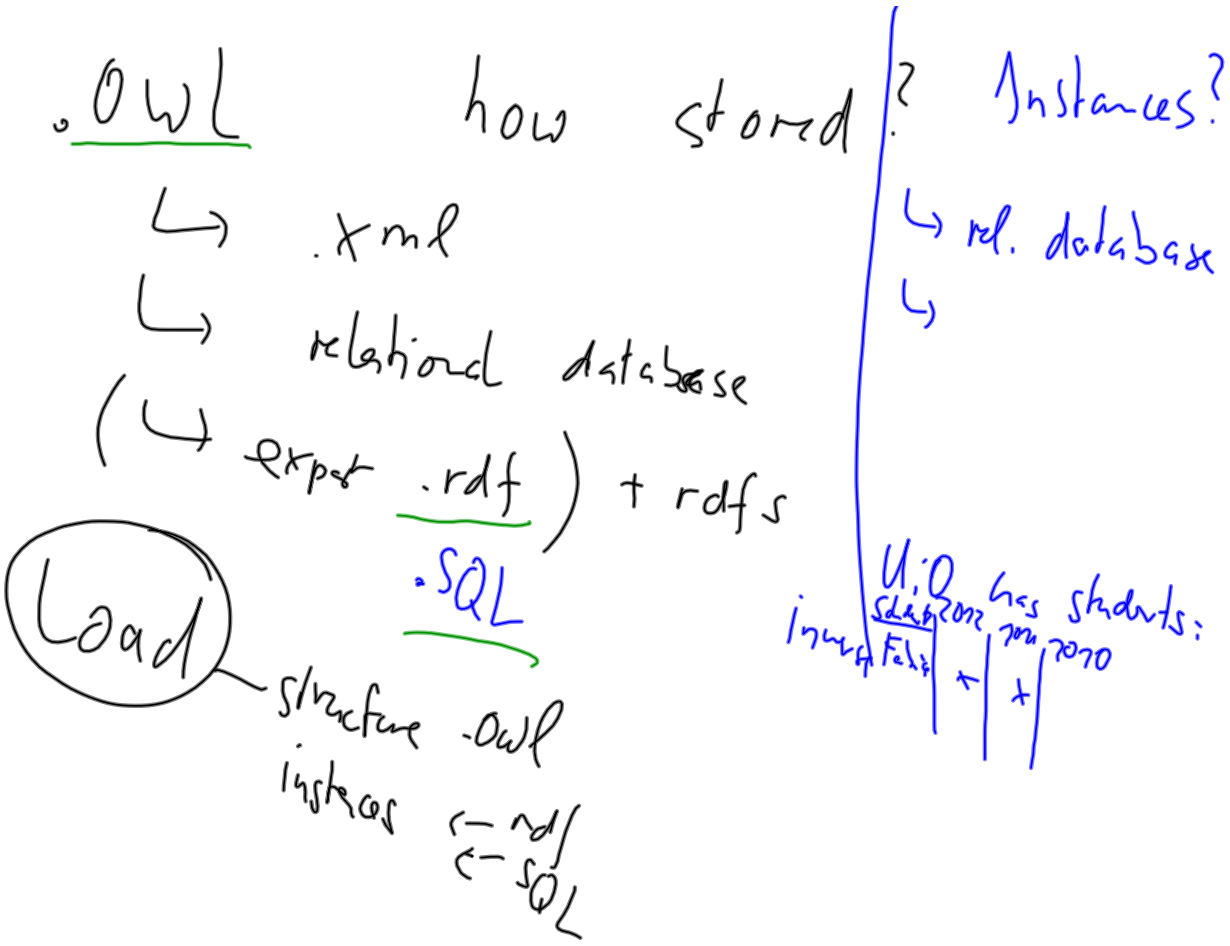
need DROOLS / JESS "default"

API (I check Jess for)

OWL API 3.2.4
Pellet 2.2

Compatible

OWL ⟷ OWL



.owl how stored?

↳ .xml

↳ relational database

(↳ export .rdf) + rdfs

.SQL

Load — structure .owl

instances ← rdf

← SQL

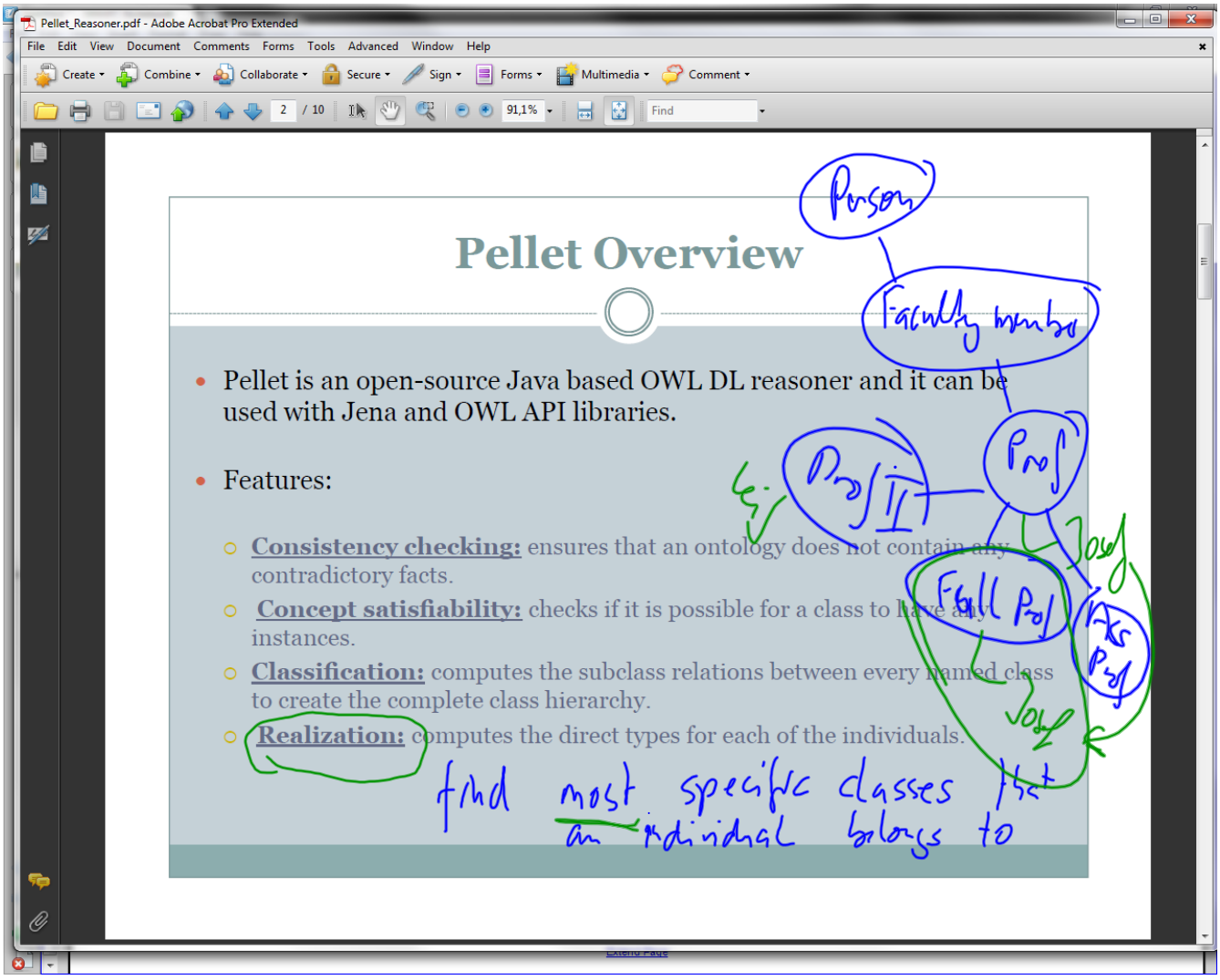
Instances?

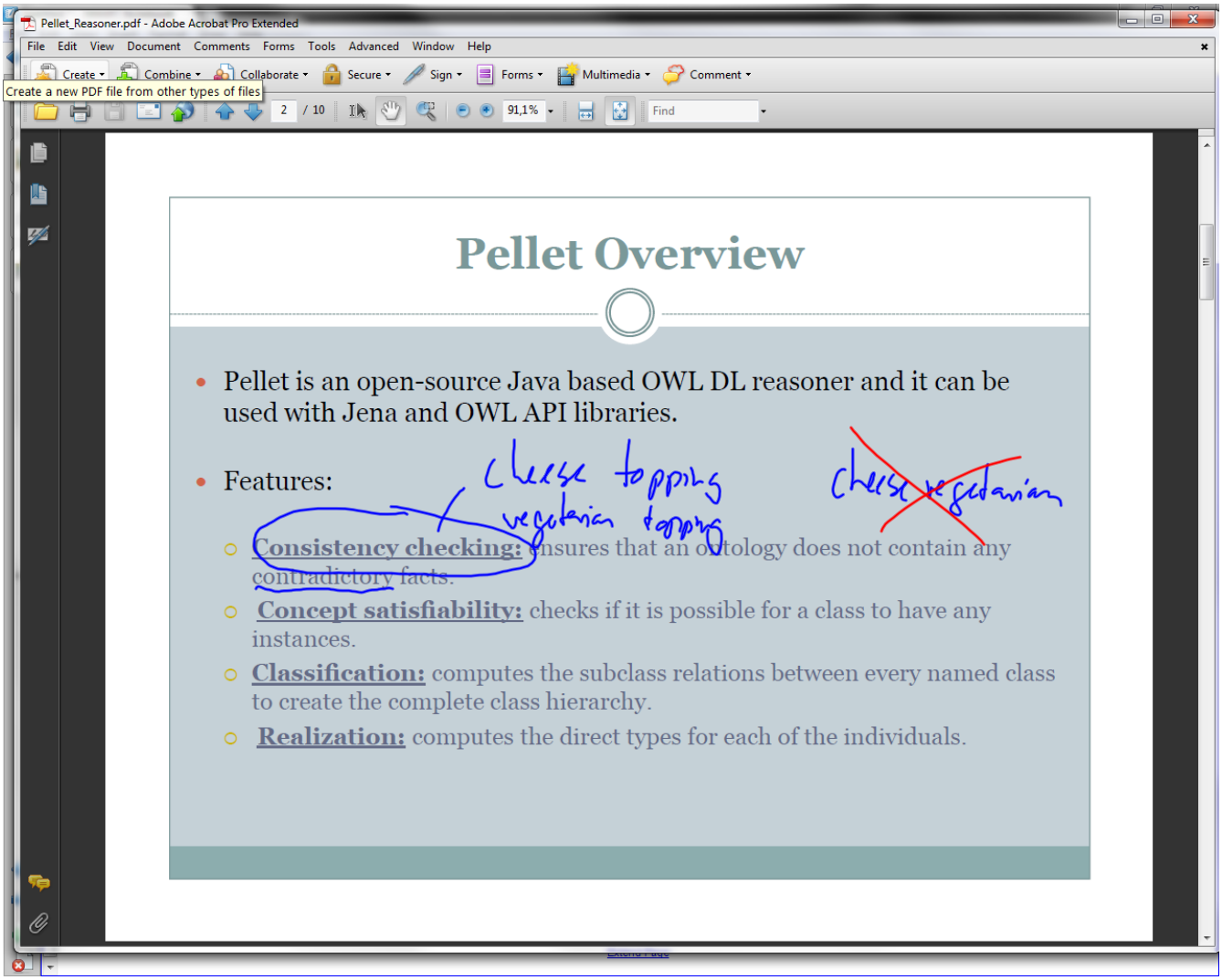
↳ rel. database

↳

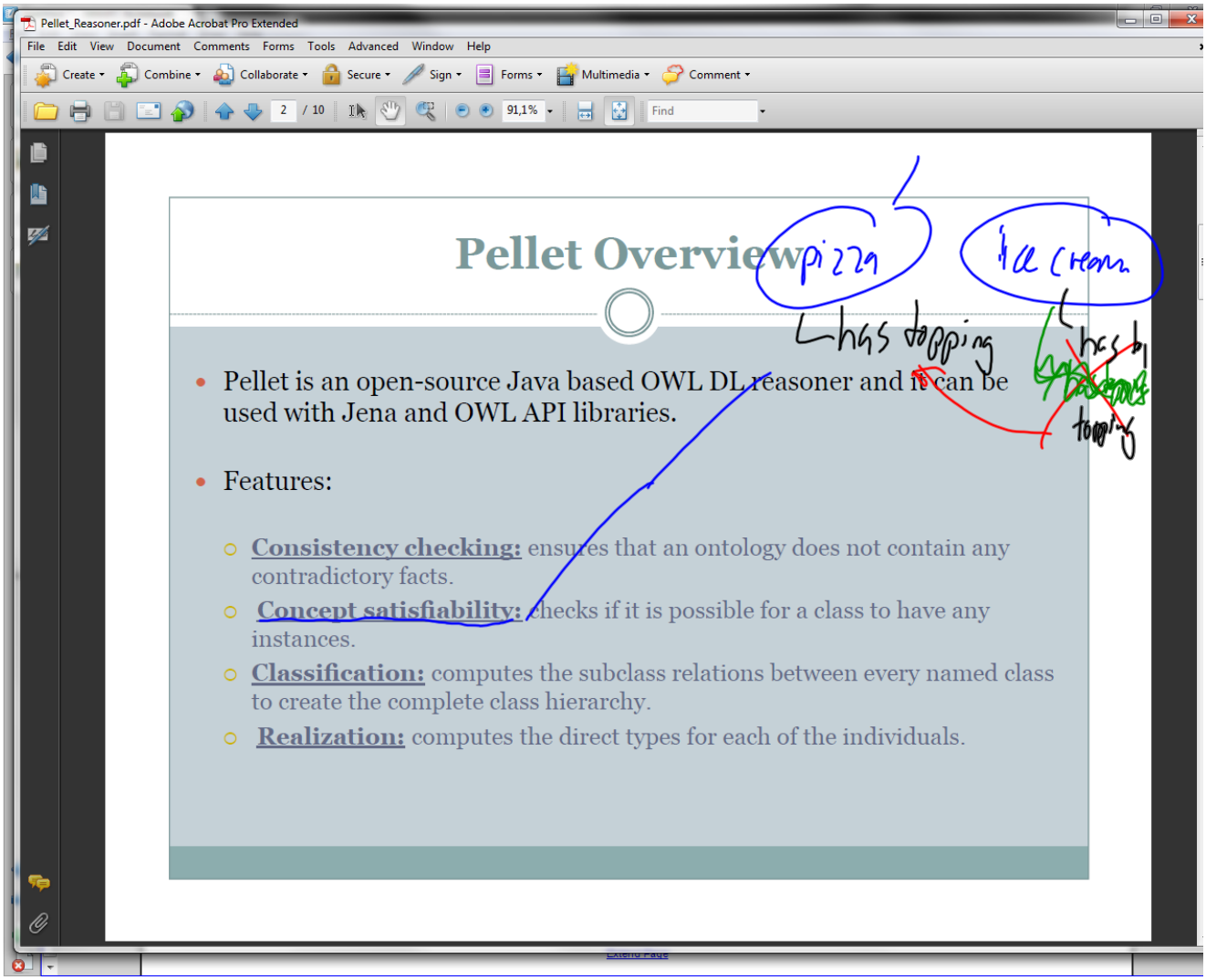
UiO has students:

Student	2012	2011	2010
inverse Fabio	x	x	


Pellet_Reasoner.pdf - Adobe Acrobat Pro Extended
File Edit View Document Comments Forms Tools Advanced Window Help
Create Combine Collaborate Secure Sign Forms Multimedia Comment
2 / 10
91,1%
Find
Pellet Overview
Pellet is an open-source Java based OWL DL reasoner and it can be used with Jena and OWL API libraries.
Features:
Consistency checking: ensures that an ontology does not contain any contradictory facts.
Concept satisfiability: checks if it is possible for a class to have any instances.
Classification: computes the subclass relations between every named class to create the complete class hierarchy.
Realization: computes the direct types for each of the individuals.
Person
Faculty member
Prof
Full Prof
find most specific classes that an individual belongs to


Pellet_Reasoner.pdf - Adobe Acrobat Pro Extended
File Edit View Document Comments Forms Tools Advanced Window Help
Create Combine Collaborate Secure Sign Forms Multimedia Comment
Create a new PDF file from other types of files
2 / 10
91,1%
Find
Pellet Overview
Pellet is an open-source Java based OWL DL reasoner and it can be used with Jena and OWL API libraries.
Features:
cheese topping
vegetarian topping
cheese vegetarian
Consistency checking: ensures that an ontology does not contain any contradictory facts.
Concept satisfiability: checks if it is possible for a class to have any instances.
Classification: computes the subclass relations between every named class to create the complete class hierarchy.
Realization: computes the direct types for each of the individuals.

Inconsistency ← two classes have contradiction
← class has no instances



Data property : is a value

has data type

has email address

jose@unik.no

Object property : connect classes

has student

Goto Meeting: 730-354-530

Arne

OWL	SMW
property exists	property
type	type
	⋮ Subject

Protégé 3x vs 4x

Feature	Protégé 3x	Protégé 4x
OWL	OWL 1.0 OWL and RDF(S) support	OWL 2.0 Pure OWL framework
API	Protege-OWL API	OWL API
SPARQL	Yes	Not yet
SWRL	SWRLTab	through a basic editor and pellet for reasoning
meta-modeling	allowing OWL Full	No OWL Full
Reasoner	through HTTP DIG interface or direct connection	Direct connection to FaCT++, Pellet and other DL reasoners
Multi-user Support	using the client-server version of Protege	None (Protege Client-Server has not been migrated yet)
Database Storage Model	Ability to store ontologies in a database provided by the JDBC database back-end	None (database back-end has not been migrated yet)

3

http://protegewiki.stanford.edu/wiki/Protege4Migration