

Jose: Ontology ✓

java export : export ✓

Start Java project: ↪ .java

- Create classes & structure {
- same project folder as Protege
 - add .jar files into project
 - add .java files (ontology)
 - start OWL-API to fetch indiv. (to perform reasoning)
- :

Implementation Semantic x cvi.unik.no/images/UNIK x

cvi.unik.no/images/UNIK4710-L16-v13.pdf

Zoom: 200 %

Facts

- code example
- communicate with owl
- thinks OWL



Criteria

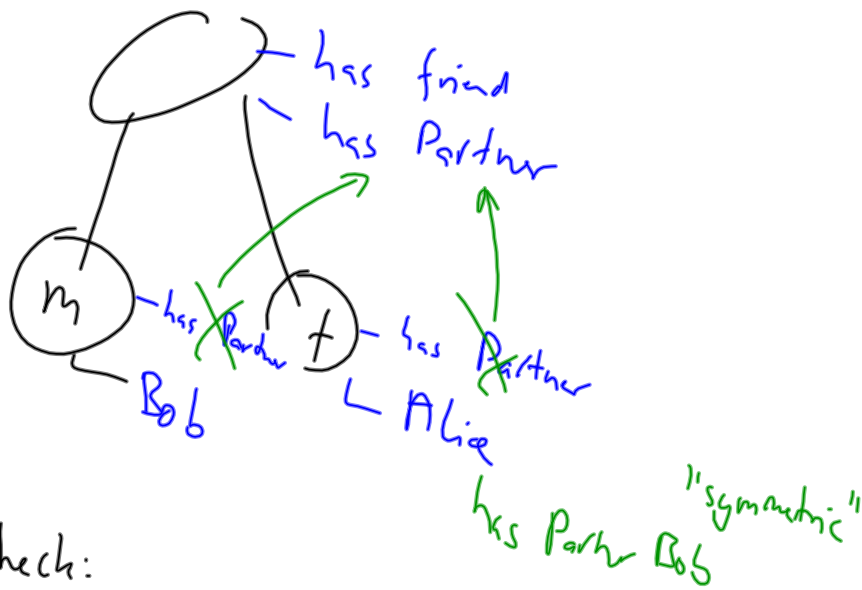
- easy expert // user friendly
- creates structure for java prog
- allows to "think" java
- just classes and relations
- don't need to think OWL
- implicit knowledge
- programming, object oriented

Evaluation

Recommendation

- more logic // business process oriented
- does not need ~~OWL~~ ^{Protege} knowledge through OWL API
- "get FirstName" = how to write back knowledge into .owl



check:

IMDB

Linked open data

linked imdb.org

"Final Presentation"

App Scenario "normal user"

Specific focus on "outcome" Reasoner

↳ vocal expl.

Implementation

- how (generic)

- examples of code and outcome

- Demo #1

- Evaluation (Open issues, limitations, critics, ...)

- Conclusion

*1 Emulator, AirDroid

Final lectures

10.5. Q&A

(17.5. Public holidays)

⇒ 24.5. Presentations

37.5. /
7.6 Grades Exam