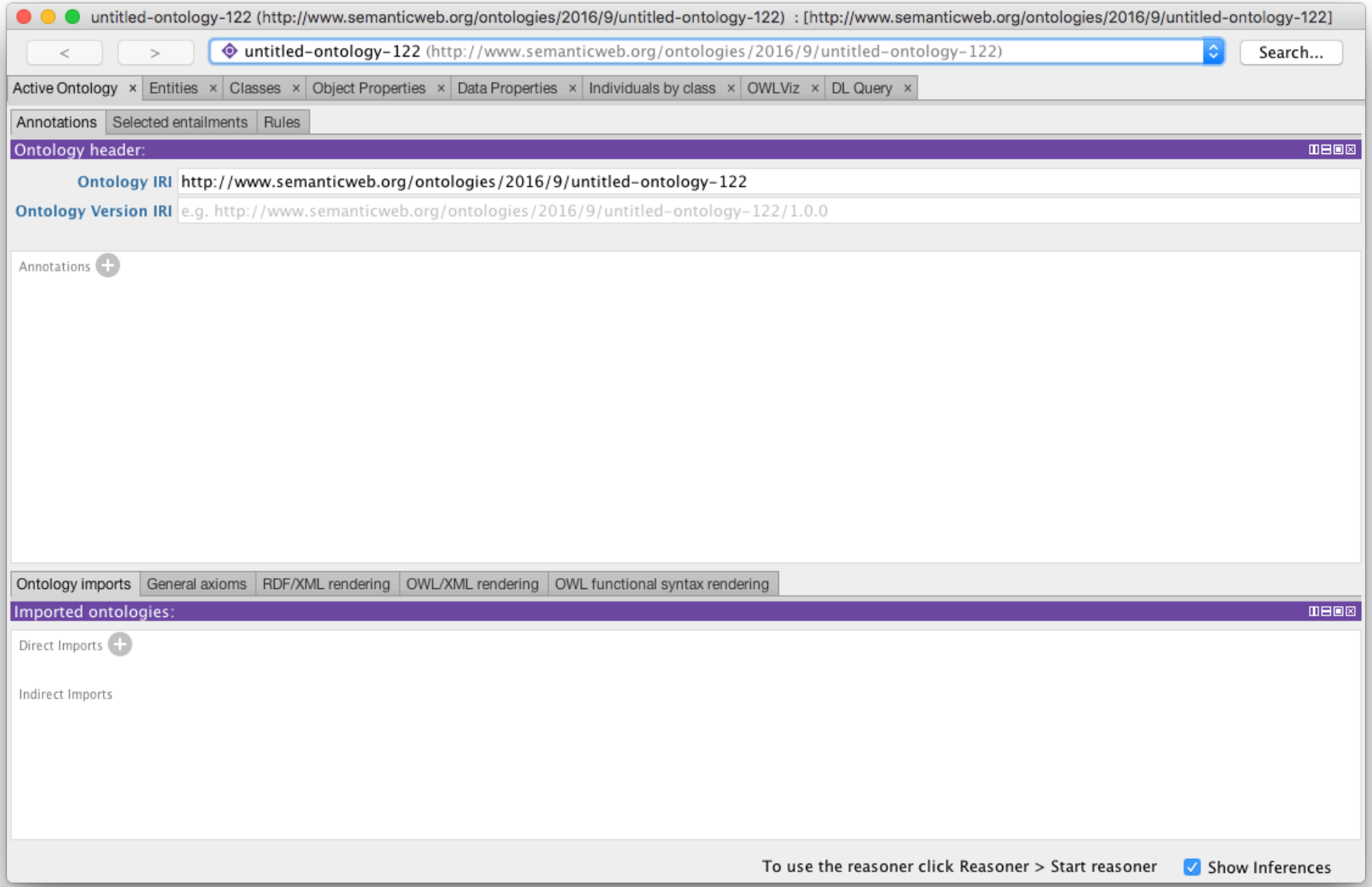
Ontology Editors

IDEs for Ontologies

- Some people use simple text editors
  - Working with XML serialization will drive you crazy
  - Using Turtle or an abstract syntax works well
- Others prefer an IDE
  - Good IDEs include support for reasoning, visualization, and more
- Protégé is a very popular IDE
  - From Stanford, free, lots of plugins
- TopQuadrant Composer is also good
  - Feature rich but expensive ($600 for a single license)

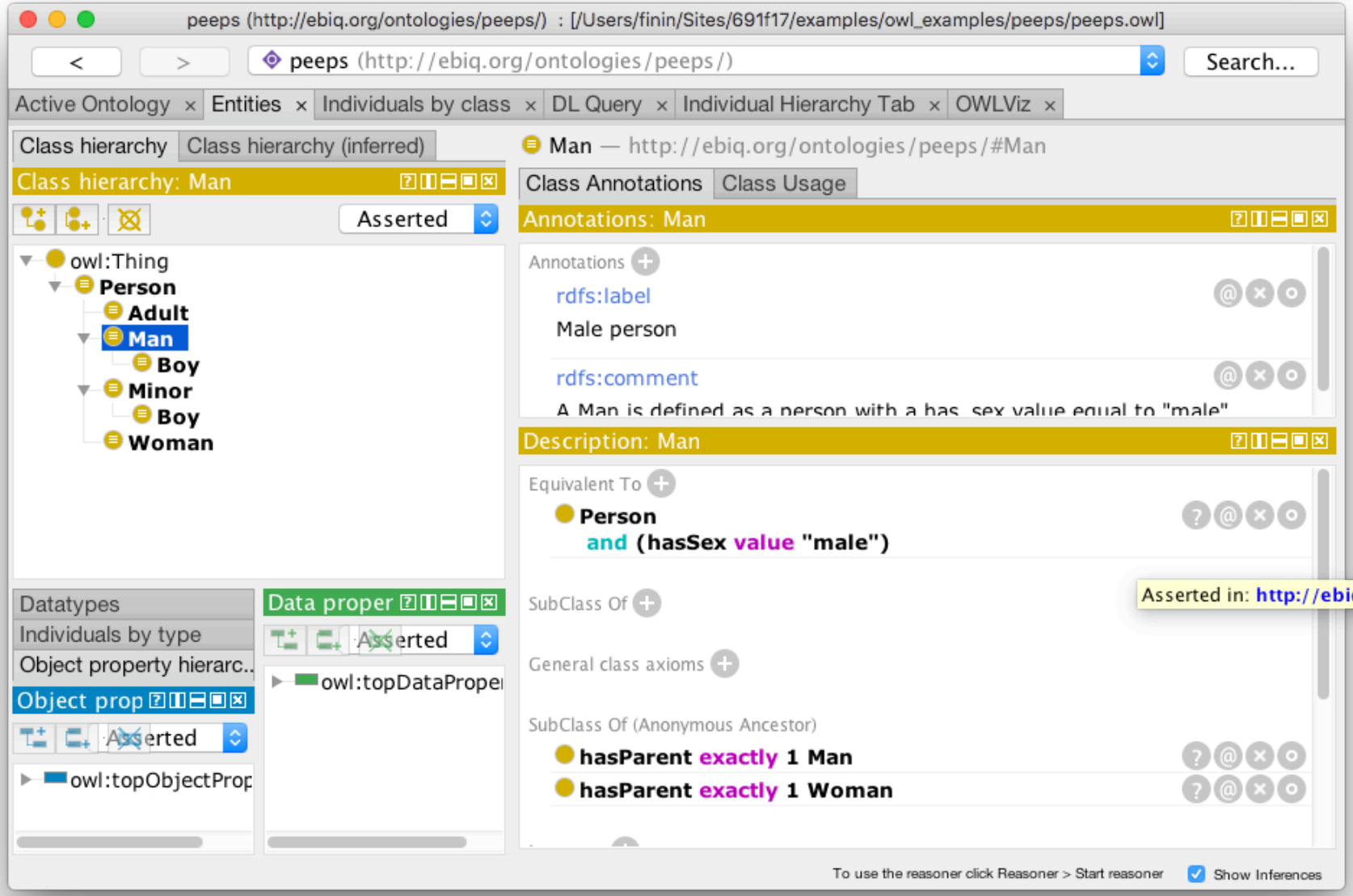
Protégé 5.5


untitled-ontology-122 (http://www.semanticweb.org/ontologies/2016/9/untitled-ontology-122) : [http://www.semanticweb.org/ontologies/2016/9/untitled-ontology-122]
untitled-ontology-122 (http://www.semanticweb.org/ontologies/2016/9/untitled-ontology-122)
Search...
Active Ontology
Entities
Classes
Object Properties
Data Properties
Individuals by class
OWLViz
DL Query
Annotations
Selected entailments
Rules
Ontology header:
Ontology IRI
http://www.semanticweb.org/ontologies/2016/9/untitled-ontology-122
Ontology Version IRI
e.g. http://www.semanticweb.org/ontologies/2016/9/untitled-ontology-122/1.0.0
Annotations
Ontology imports
General axioms
RDF/XML rendering
OWL/XML rendering
OWL functional syntax rendering
Imported ontologies:
Direct Imports
Indirect Imports
To use the reasoner click Reasoner > Start reasoner
Show Inferences

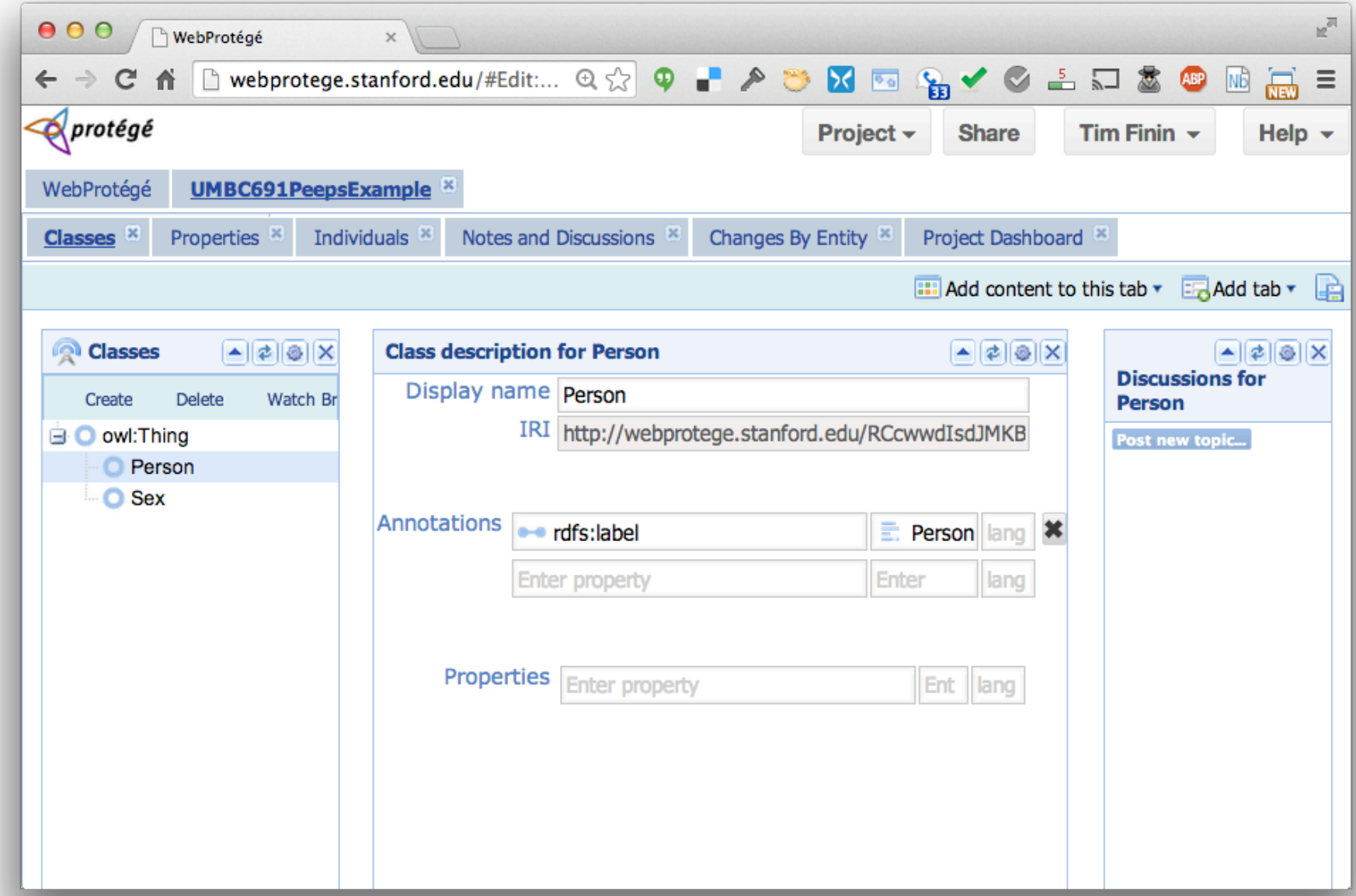
Protégé 5.5

- http://protege.stanford.edu/
- Free, open source ontology editor and KB framework
- Predates OWL, still supports earlier Frames representation
- In Java, extensible, large community of users
  - Requires Java Runtime Environment
- Desktop and Web versions
  - Works will under Linux, Mac OS X and Windows

Desktop Protégé



Web Protégé


WebProtégé
webprotege.stanford.edu/#Edit:...
protégé
Project
Share
Tim Finin
Help
WebProtégé
UMBC691PeepsExample
Classes
Properties
Individuals
Notes and Discussions
Changes By Entity
Project Dashboard
Add content to this tab
Add tab
Classes
Create
Delete
Watch Br
owl:Thing
Person
Sex
Class description for Person
Display name
Person
IRI
http://webprotege.stanford.edu/RCcwwdIsdJMKB
Annotations
rdfs:label
Person
lang
Enter property
Enter
lang
Properties
Enter property
Ent
lang
Discussions for Person
Post new topic...

YAS: Yet Another Syntax

- Neither OWL's official abstract syntax nor XML serialization is easy to read or use
- Protégé uses the Manchester syntax
- Simpler and more compact: “some” and “only”, not “someValuesFrom” and “allValuesFrom”
- A W3C recommendation (http://bit.ly/manSyn), used in the OWL 2 Primer (http://bit.ly/OWL2Pri)

```
Class: man

    Annotations: rdfs:label "man"

    EquivalentTo: adult and male and person
```

Manchester OWL syntax

OWL	DL Symbol	Manchester OWL Syntax Keyword	Example
someValuesFrom	∃	some	hasChild **some** Man
allValuesFrom	∀	only	hasSibling **only** Woman
hasValue	∋	value	hasCountryOfOrigin **value** England
minCardinality	≥	min	hasChild **min** 3
cardinality	=	exactly	hasChild **exactly** 3
maxCardinality	≤	max	hasChild **max** 3

Manchester OWL syntax

OWL	DL Symbol	Manchester OWL Syntax Keyword	Example
intersectionOf	⊓	and	`Doctor and Female`
unionOf	⊔	or	`Man or Woman`
complementOf	¬	not	`not Child`

Example 1

How can we define a class that is people who have children and all of them are male?

Example: People with just boys

How can we define a class that is people who have children and all of them are male?

Define as the union of three classes

1. Person
2. Things that have children
3. Things where all of their children are mail

Example: People with just boys

How can we define a class that is people who have children and all of them are male?

Define as the union of three classes

1. Person
2. Things that have children
3. Things where all of their children are male

An **owl**:someValuesFrom restriction on hasChild property

An **owl**:allValuesFrom restriction on hasChild property

Example: People with just boys

How can we define a class that is people who have children and all of them are male?

```
Person and
   (hasChild only Man) and
   (hasChild some Person)
```

Example 2

```
Person and
   hasChild some
      (Person and
         (hasChild only Man) and
           (hasChild some Person))
```

The set of people who have at least one child that has some children that are only men (i.e., grandparents that only have a Pjb)

Data values and datatypes

- Data values typed or untyped (e.g., int, boolean, float)
- Constants w/ or w/o type, e.g.: hasAge value "21"^^long
- Use datatype names as classes: hasAge some int
- XSD facets, e.g.: Person and hasAge some int[>= 65]
- Ranges: Person and hasAge some int[>= 18, <= 30]

XSD facet	Meaning
< x, <= x	less than, less than or equal to x (more info)
> x, >= x	greater than, greater than or equal to x (more info)
length x	For strings, the number of characters must be equal to x (more info)
maxLength x	For strings, the number of characters must be less than or equal to x (more info)
minLength x	For strings, the number of characters must be greater than or equal to x (more info)
pattern regexp	The lexical representation of the value must match the regular expression, regexp (more info)
totalDigits x	Number can be expressed in x characters (more info)
fractionDigits x	Part of the number to the right of the decimal place can be expressed in x characters (more info)

Demonstration

- We'll use Protégé OWL v5.5 to implement a tiny ontology for people
- Start by downloading and installing Protégé 5.5 (You will need the JRE installed)
- You may want to install Graphviz
- Configure Protégé
  - E.g., select a reasoner to use (e.g., HermiT)

A basic workflow

- Think about usecases
- Preliminaries
  - Choose namespace URL, import other ontologies used
- Identify and define classes
  - Place in hierarchy, add **axioms** and run reasoner to check for errors or omissions
- Identify and define properties
  - Place in hierarchy, add **axioms**, run reasoner
- Add individuals & reasoner to check for problems
- Add comments and labels
- Export in desired formats, maybe upload to Web

More workflow steps

- Use OOPS to find common ontology pitfalls

  OntOlogy Pitfall Scanner detect many common pitfalls introduced when developing ontologies

- Link concepts (and individuals) to common ontologies (e.g., DBpedia, Freebase, foaf)

  Use owl:sameAs

- Generate visualizations
- Produce documentation
- Develop examples with your use case(s)
- Encode data, describe in VoID (Vocabulary of Interlinked Datasets), add to LOD cloud

Demonstration/HW4

Use Protégé OWL (v5.5) to build a simple ontology for people based on the following

- People have just one sex that's either *male* or *female*, an integer age, and two parents, one male, one female
- A person's grandparent is the parent of their parent
- Every person is either a man or a woman but not both
- A man is defined as any person whose sex is male and a woman as any person whose sex is female
- A boy is defined as a person whose sex is male and whose age is less than 18, a girl is …
- A person is either an adult or (age >18), minor (age <18), …

Test cases

AllDifferent people

Alice F

Bob M

Carol F

Don M

Edith F

Pat ?

Other people

Frank M

Gwen F

Some possible test cases

- Alice parent Bob . Bob parent Carol
  - Alice grandparent Carol
- Alice parent Bob . Alice parent Don.
  - Contradiction
- Alice parent Bob . Pat parent Bob
  - Pat a female
- Alice parent Bob . Gwen parent Bob .
  - Alice owl:sameAs Gwen