Machine learning with biomedical ontologies

Presented by Sarah Alghamdi, Azza Althaqafi, Robert Hoehndorf, Maxat Kulmanov, Sumyyah Toonsi, Fernando Zhapa-Camacho

Learning Outcomes

- Introduce different methods that use ontologies in machine learning models
- Discuss unsupervised machine learning methods that can “embed” from one structure to another.
- Introduce mOWL, a software library for machine learning with ontologies
- Incorporate mOWL in Biomedical data analysis using different approaches

Preliminaries: ontologies

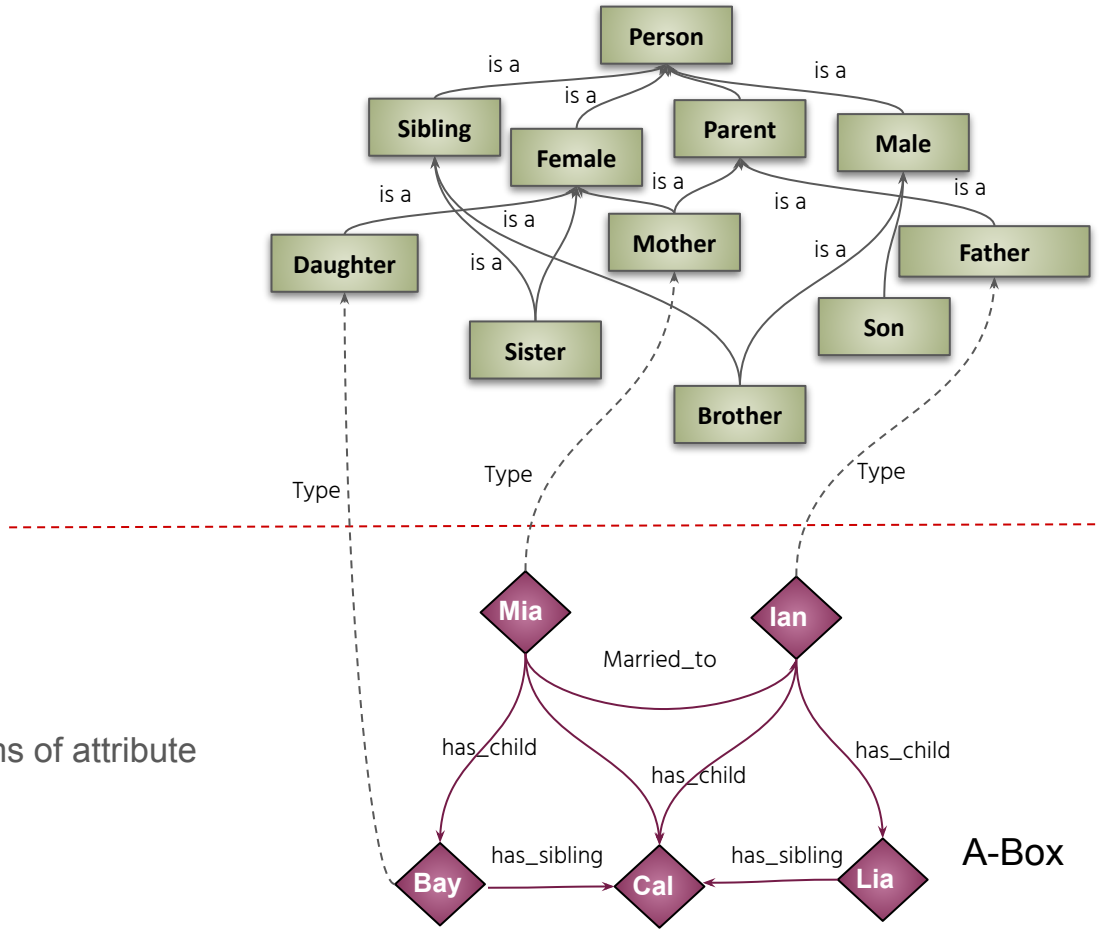
- “ An ontology is an explicit specification of a conceptualization” … Gruber 1993
- “ An ontology is an explicit formal specification of a shared conceptualization” … Borst 1997
- “ An ontology is a logical theory designed in order to capture the intended models corresponding to a certain conceptualization and to exclude the unintended ones” … Guarino 2009

Preliminaries: ontologies

- Ontology consist of :
- T-Box
  - Set of terminological Component

- A-Box
  - Set of assertions using T-Box
    - Concept assertions
    - Relation Assertion

- Metadata
  - representation of a resource in terms of attribute name-value pairs
    - Definition
    - Labels
    - …



Preliminaries: ontologies

- Description Logic (DL) is used to formally and exipisity represent ontologies

Name	DL syntax	Semantics
Top concept	$\top$	$\Delta^{\mathcal{I}}$
Bottom concept	$\bot$	$\emptyset$
Concept	C	$C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Concept disjunction	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
Concept conjunction	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
Concept negation	$\neg C$	$\Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$
Universal restriction	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}}((x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\}$
Existential restriction	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}((x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}})\}$
Subclass of	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
Subproperty of	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$
Equivalent class	$C_1 \equiv C_2$	$C_1^{\mathcal{I}} = C_2^{\mathcal{I}}$
Equivalent property	$R_1 \equiv R_2$	$R_1^{\mathcal{I}} = R_2^{\mathcal{I}}$

Parent $\equiv \exists$ ***has_child. Peron***
Son $\sqsubseteq$ ***Male*** $\sqcap \exists$ ***child_of. Person***
Mother $\sqsubseteq$ ***Female*** $\sqcap$ ***Parent***
Sibling $\sqsubseteq \exists$ ***has_sibling. Person***

Preliminaries: ontologies

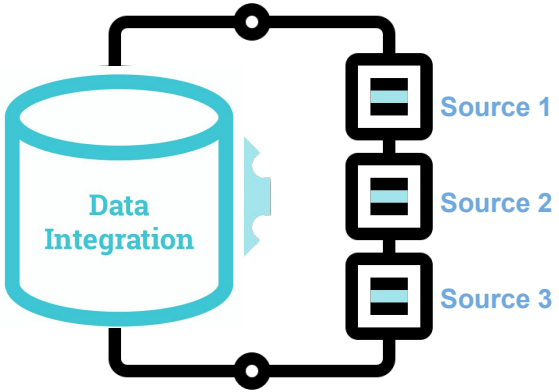
- Description Logic (DL) is used to formally and exipisity represent ontologies

DL Syntax	Manchester Syntax
$C \sqcap D$	C and D
$C \sqcup D$	C or D
$\neg C$	not C
$\exists R.C$	R some C
$\forall R.C$	R only C
$(\geq nR.C)$	R min n C
$(\leq nR.C)$	R max n C
$(= nR.C)$	R exactly n C
$\{a\} \sqcup \{b\} \sqcup \ldots$	{a b ...}

$Parent \equiv \exists has\_child.\ Peron$
$Son \sqsubseteq Male \sqcap \exists child\_of.\ Person$
$Mother \sqsubseteq Female \sqcap Parent$
$Sibling \sqsubseteq \exists\ has\_sibling.\ Person$

How Ontologies are used in Databases

- Annotations and data integration
  - Ontologies play a crucial role in facilitating data integration across databases due to their usage of standard identifiers for classes and relations

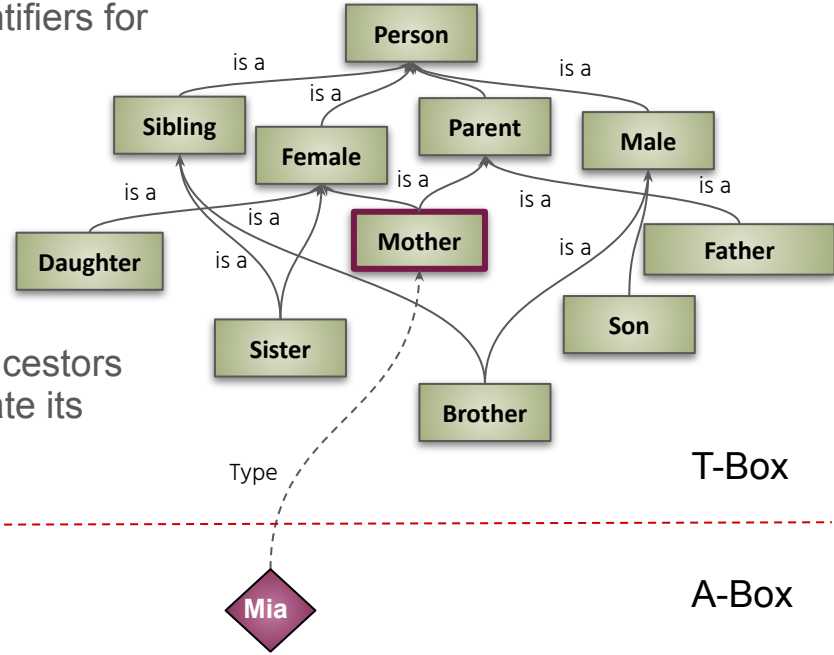


- True path rule:
  - Annotation for a class is immediately passed to its ancestors
  - Unannotated entities for a class is not used to annotate its descendants

How Ontologies are used in Databases

- Annotations and data integration
  - Ontologies play a crucial role in facilitating data integration across databases due to their usage of standard identifiers for classes and relations

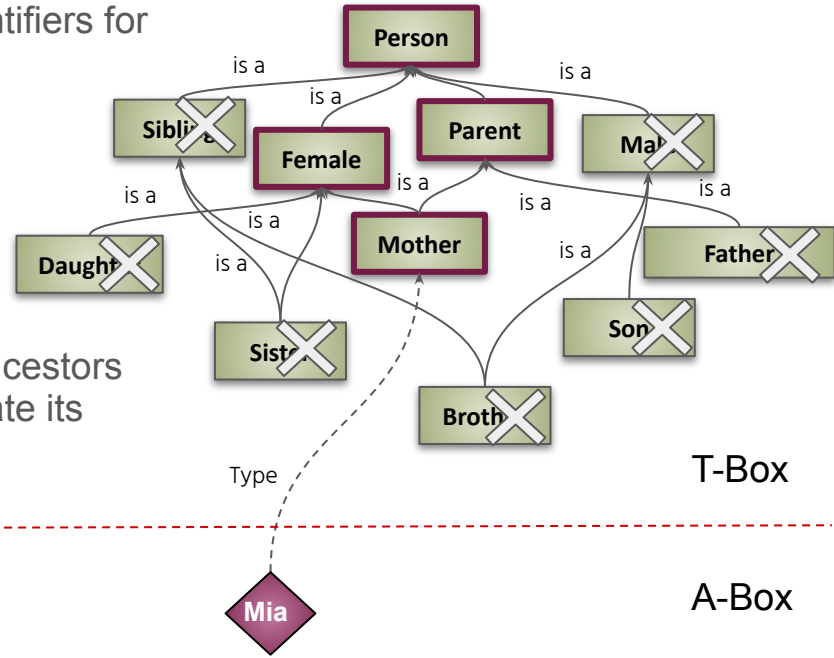
- True path rule:
  - Annotation for a class is immediately passed to its ancestors
  - Unannotated entities for a class is not used to annotate its descendants



How Ontologies are used in Databases

- Annotations and data integration
  - Ontologies play a crucial role in facilitating data integration across databases due to their usage of standard identifiers for classes and relations

- True path rule:
  - Annotation for a class is immediately passed to its ancestors
  - Unannotated entities for a class is not used to annotate its descendants



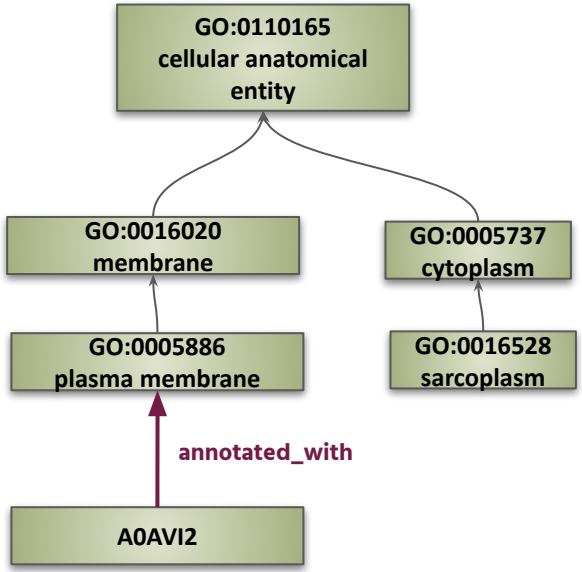
T-Box

Annotations to OWL axioms

Example:

Annotating protein **A0AVI2** To Gene Ontology

- Annotations to T-Box
  - $\textbf{\textit{A0AVI2}} \sqsubseteq \exists\ \textit{annotated\_with.}\ \textbf{\textit{GO:0005886}}$



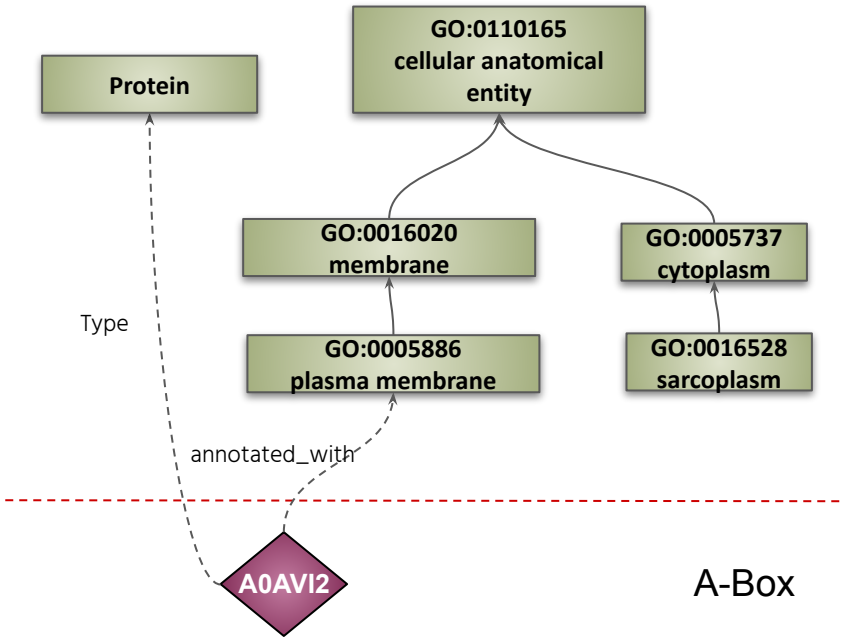
Annotations to OWL axioms

Example:

Annotating protein **A0AVI2** To Gene Ontology

- Annotations to T-Box
  - ***A0AVI2*** $\sqsubseteq \exists$ *annotated_with.* ***GO:0005886***

- Annotations to A-Box
  - Protein(***A0AVI2***)
  - annotated_with(***A0AVI2, GO:0005886***)



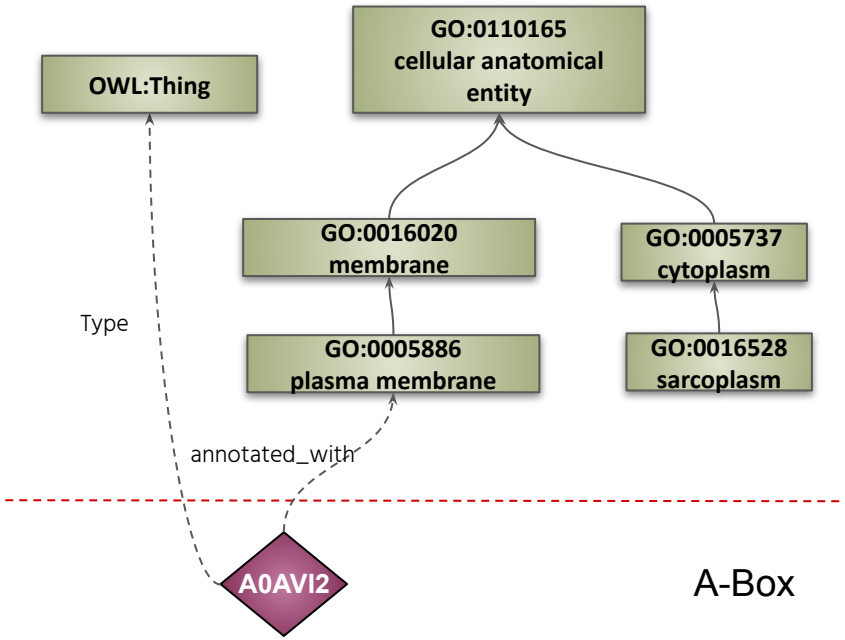
Annotations to OWL axioms

Example:

Annotating protein **A0AVI2** To Gene Ontology

- Annotations to T-Box
  - ***A0AVI2*** $\sqsubseteq \exists$ *annotated_with.* ***GO:0005886***

- Annotations to A-Box
  - OWL:Thing(***A0AVI2***)
  - annotated_with(***A0AVI2, GO:0005886***)

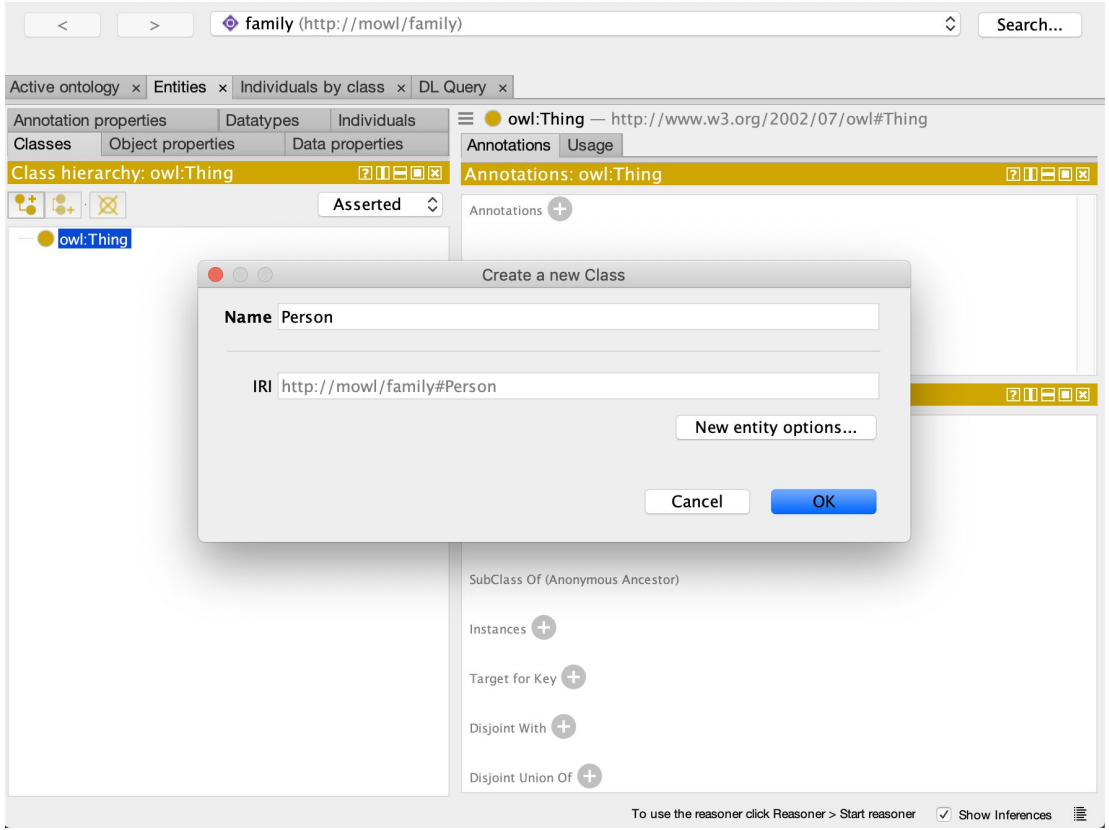


Hands on (1):

- Creating the family ontology
  - Using Protégé
    - Download from: https://protege.stanford.edu/

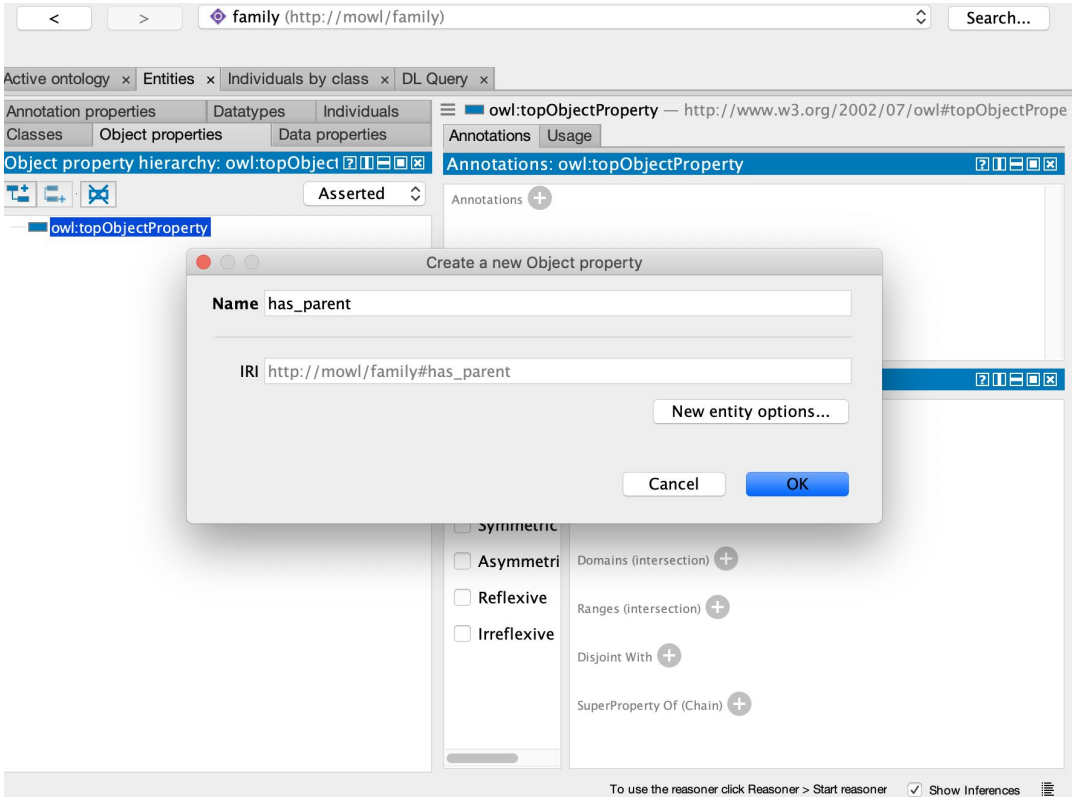
Hands on (1):

- Protégé:
  - Add new classes



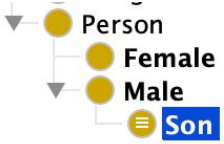
Hands on (1):

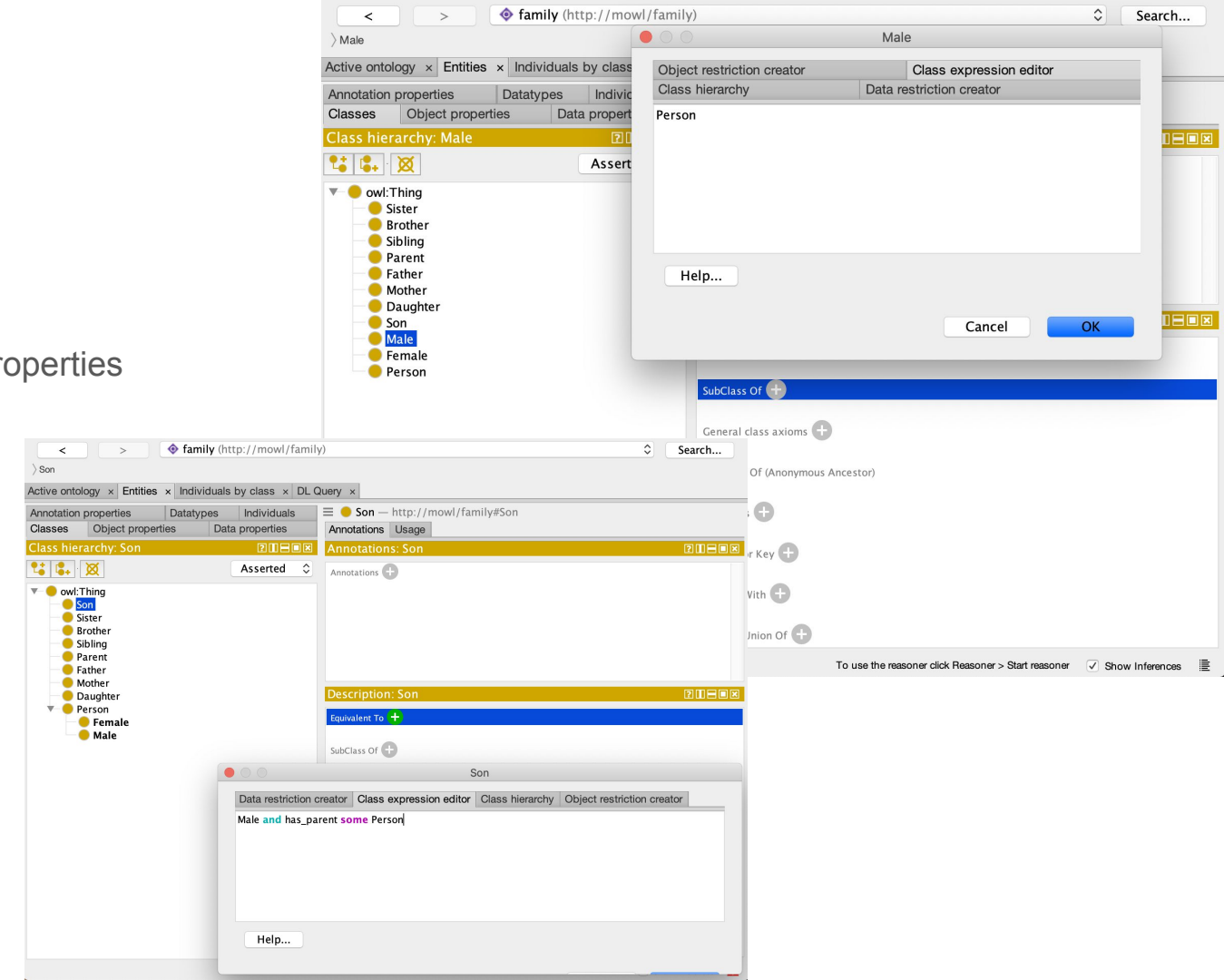
- Protégé:
  - Add new classes
  - Add new object properties



Hands on (1):

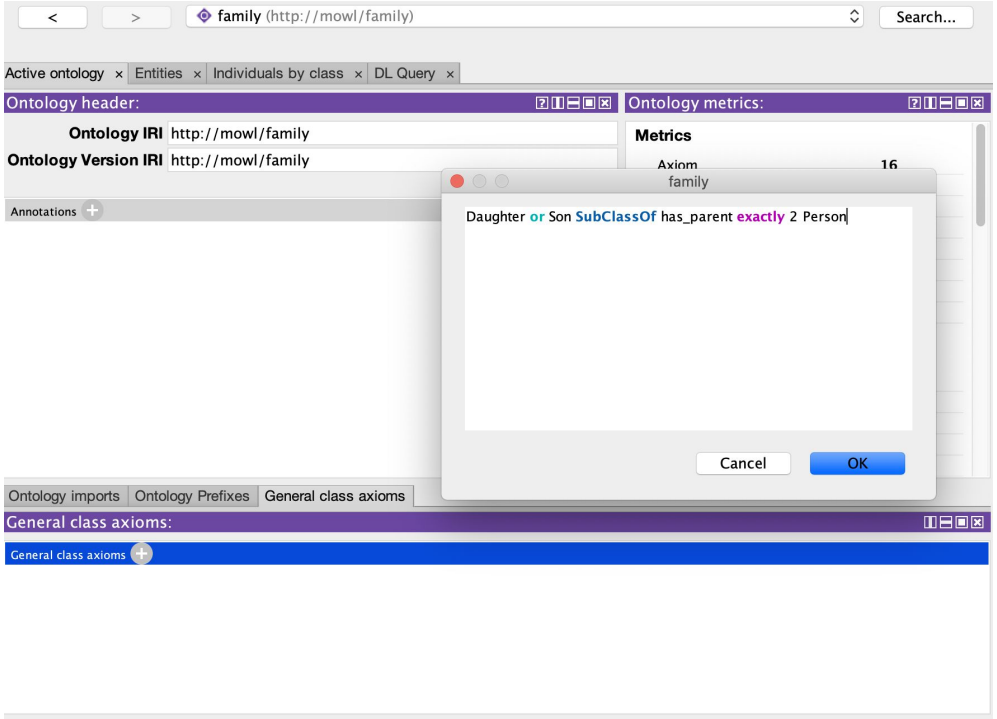
- Protégé:
  - Add new classes
  - Add new object properties
  - Adding axioms
    - SubclassOf
    - Equivalent
    - ….





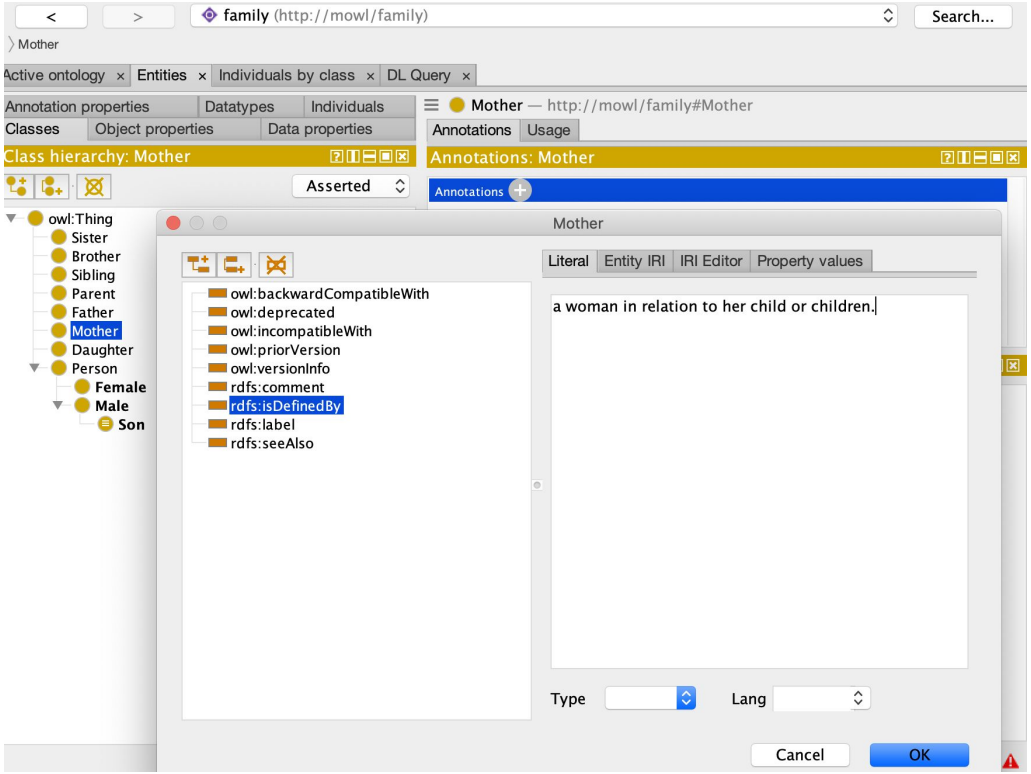
Hands on (1):

- Protégé:
  - Add new classes
  - Add new object properties
  - Adding axioms
  - Adding GCIs



Hands on (1):

- Protégé:
  - Add new classes
  - Add new object properties
  - Adding axioms
  - Adding GCIs
  - Adding definitions, synonyms



Hands on (1):

- Protégé:
  - Add new classes
  - Add new object properties
  - Adding axioms
  - Adding GCIs
  - Adding definitions, synonyms
  - Using reasoners