Logics and Symbolic AI

Ontologies Assignment Report

Students:
Thomas Wimmer
Ümit Bora Gökbakan
Jules Soria
Haileleul Haile
Henon Lamboro
Jana Sweidan

May 23, 2023

Contents

1 Introduction

The aim of this project is to develop an ontology describing a topic of our choosing. Ontologies are a means to formally model concepts from a particular domain into a detailed specification of entities with properties and relations [1]. For this project, we are tasked with identifying a problem we want to represent as an ontology, designing the ontology, and simulating it using Protege. Protege is a free, open-source ontology editor and framework for building intelligent systems [2]. This report presents the topic we have chosen to work on, the narration of the story which led to our design of the ontology, our design of the ontology, and the final result.

2 Problem Statement

The Trojan war is one of the most legendary conflicts in history. The actuality of the conflict remains to be debated but there are archeological recoveries in present-day Turkey that suggest that it might be true. Nevertheless, it is accounted for by many great authors such as the Greek historian Homer and the roman poet Virgil. The events are described in detail in Homer's poem *The Iliad*.

The Trojan war was a legendary conflict between the Greeks and the Trojans dated by later Greek authors to the 12th or 13th century BCE. Trojan War was caused by Paris, son of the Trojan king, and Helen, wife of the Greek king Menelaus when they went off together to Troy. To get her back, Menelaus sought help from his brother Agamemnon, who assembled a Greek army to defeat Troy. Another myth attributes the origin of the Trojan War to a quarrel between the goddesses Athena, Aphrodite, and Hera over who among them was the fairest. After Paris chose Aphrodite, Athena and Hera plotted against Troy. That was followed by 10 years of war involving at least 100,000 men in each army as well as 1,184 Greek ships.

Homer presents the story in magnificence as a conflict even the mythological Greek gods took part. In his description, the conflict is so great that the Gods sided with either of the two conflicting sides. It is said that Apollo, Artemis, Ares, and Aphrodite sided with the Trojans, while Hera, Athena, Poseidon, Hermes, and Hephaestus aided the Greeks.

The Trojan war was also famous for introducing heroes such as Achilles, Odysseus, Nestor, and Ajax on the side of the Greeks and Hector, Sarpedon, Memnon, and Aeneas on the side of the Trojans. The legend of these heroes and the war still lives on in the form of everyday English idioms such as Achilles Heel and Trojan Horse.

The interaction between these many characters in the story is a rich source to develop an ontology. There are many forms of interactions within the same allegiance and with adversaries. In this project, we have looked through the many participants in the conflict, listed them, identified relationships they had with each other, and generated inference rules.

3 Narration

We have added individuals and relations based on how the story is told. Multiple chapters tell the story of *The Iliad*, and each addition is a way for the ontology to describe the events foretold. This is an attempt to follow the first chapters of the story alongside the ontology we created.

Plan of Zeus

According to Greek mythology, **Zeus** had become king of the gods by overthrowing his father **Cronus**; Cronus in turn had overthrown his father **Uranus**. Zeus was not faithful to his wife and sister **Hera**, and had many relationships from which many children were born. Zeus was mad at mankind, and firmly believed that Earth was overpopulated. Furthermore, Zeus came to learn that, like his father Cronus, he would be overthrown by one of his sons. The god-king meddled in human affairs and arranged a major war between the **Greeks** and the **Trojans** to depopulate the Earth, especially of demi-gods.

```
isChildOf(Cronus, Uranus)
isChildOf(Cronus, Gaia)
kills(Cronus, Uranus)

isChildOf(Zeus, Cronus)
isChildOf(Zeus, Rhea)
kills(Zeus, Cronus)

isSiblingOf(Zeus, Hera)
isMarriedTo(Zeus, Hera)

Trojan(x) ⇔ ¬Greek(x)
```

The Prophecy

```
rulesOver(Priam, Troy)
isMarriedTo(Priam, Hecuba)

isChildOf(Hector, Hecuba)
isChildOf(Hector, Priam)

isChildOf(Alexander, Hecuba)
isChildOf(Alexander, Priam)

isChildOf(Cassandra, Hecuba)
isChildOf(Cassandra, Priam)

isSameAs(Alexander, Paris)
```

Alexander was the second-born son of the **Trojan King Priam** and **Queen Hecuba**. During her pregnancy with her second son, Hecuba had an ominous dream of birthing a huge, burning torch that was covered in writhing serpents. She sought out local prophets who warned the queen that her second son would cause the downfall of Troy.

The couple concluded that Alexander had to die. Priam left the death of the infant Alexander in the hands of one of his shepherds, who intended to leave the prince in the wilderness to die of exposure. When the shepherd returned and found Alexander in good health, he viewed it as divine intervention and brought the infant home with him, raising him under the name **Paris**.

The Wedding of Peleus and Thetis

The nymph **Thetis** was courted by both **Zeus** and **Poseidon**, but neither of them married her, out of fear of a prophecy that said Thetis’ son would surpass his father in glory. The gods arranged for Thetis to get married to an aging **Phthian king** and former Greek hero, **Peleus**. Together, they had a son named **Achilles**. Thetis bathed Achilles in the Styx, the river that runs to the underworld, making him invulnerable wherever he was touched by the water. Because she had held him by the heel, it was not immersed during the bathing and thus the heel remained mortal and vulnerable to injury (the famous "Achilles’ heel"). At a crucial point in the war, she assists her son by providing weapons divinely forged by **Hephaestus**.

```
loves(Zeus, Thetis)
loves(Poseidon, Thetis)
isMarriedTo(Peleus, Thetis)
isChildOf(Achilles, Peleus)
isChildOf(Achilles, Thetis)

rulesOver(Achilles, Phthia)

blesses(Thetis, Achilles)
blesses(Hephaestus, Achilles)
```

We wanted to annotate that Peleus and Thetis are rulers of Phthia, but the way we designed the ontology does not allow for a Deity to also be Ruler (subclass of Mortals). As such, we only indicate that Achilles is the ruler, since he is of age at the time of the Trojan War.

The Judgement of Paris

```
hates(Eris, Hermes)

blesses(Aphrodite, Paris)
curses(Athena, Paris)
curses(Hera, Paris)

isMarriedTo(Helen, Menelaus)
rulesOver(Menelaus, Sparta)
```

All the gods and goddesses are invited to the wedding of Peleus and Thetis except one – **Eris**, the goddess of discord, stopped at the door by **Hermes**. Angered, she throws a golden apple into the party, bearing the inscription ’to the most beautiful’. The three goddesses **Hera, Athena, and Aphrodite**, all claim it for themselves, and the king of the gods, Zeus, not willing to get involved himself, picks the Trojan prince **Paris** as the judge. The goddess of love, Aphrodite, wins the competition as she has promised Paris possession of the most beautiful women on earth, **Helen**. There’s just one problem. Helen is already married to **Menelaus**, king of the Greek city of **Sparta**.

Elopement of Paris and Helen

Menelaus had promised Aphrodite a hecatomb, a sacrifice of 100 oxen, if he won Helen, daughter of Zeus, but forgot about it and earned her wrath. Paris, under the guise of a supposed diplomatic mission, went to Sparta to get Helen and bring her back to Troy. Helen fell in love with Paris when she saw him, as promised by Aphrodite. The two of them eloped to Troy. To bring Helen back and restore his honour, the deceived husband, King Menelaus, assembles a huge army of Greek heroes. Its leader is Menelaus' brother Agamemnon, king of the powerful Greek city of Mycenae.

```
isChildOf(Helen, Zeus)
curses(Aphrodite, Menelaus)

loves(Helen, Paris)
loves(Paris, Helen)
loves(Menelaus, Helen)

isSiblingOf(Helen, Clytemnestra)
isSiblingOf(Menelaus, Agamemnon)
isMarriedTo(Agamemnon, Clytemnestra)
rulesOver(Agamemnon, Mycenae)
```

Odysseus and Achilles

```
 rulesOver(Odysseus, Ithaca)
isMarriedTo(Penelope, Odysseus)
isFatherOf(Odysseus, Telemachus)
isFatherOf(Achilles, Neoptolemus)
isMotherOf(Deidamia, Neoptolemus)
```

Since Menelaus's wedding, **Odysseus** had married **Penelope** and fathered a son, **Telemachus**. At Skyros, **Achilles** had an affair with the king's daughter **Deidamia**, resulting in a child, **Neoptolemus**.

The Trojan War

The main protagonists of *The Iliad* are now introduced, the main setting and plot is laid out. Of course, the Trojan War lasted 10 years, with many events, pitting the Greeks (also called Achaeans) against the Trojans. All the key points of this mythical story are included in our ontology, from the siege of Troy, the military campaign of surrounding cities, the wrath of Achilles caused by Agamemnon, the death of Patroclus, the battle between Hector and Achilles, to the infamous trick of the Trojan horse by Odysseus, up to the final moments of Achilles, shot by Paris.

4 Ontology

We modeled the Trojan War on Protégé 5.5.0, creating classes and individuals (i.e., entity) [1]. We had classes to model different Characters depending on their roles, such as kings, heroes, deities, and also places like cities, regions and islands. The relations between different individuals were modeled using object property relations, while literal data properties were used to model relations such as kings and their number of ships, mentioned in the famous Catalogue of Ships in Book II of the

[1] https://github.com/boragokbakan/greeks-vs-trojans

Iliad. Protégé runs under the Open World Assumption, and we, by no means, make a claim of completeness.

In total, our ontology comprises 1386 axioms. We added 25 classes to model characters, places, alliances, and events. 281 individuals were created to instantiate these classes.

Using only-binary relations is restrictive, so we implemented a `DivineIntervention` class to normalize gods' interventions (i.e., events) into binary relations (Table 1).

As the Iliad contained many characters only named in the context of a fight who have no roles other than to kill or get killed, it was impossible to find and enter them all manually. Instead, a list of deaths was found online[2], from which individuals and kills relations were extracted into the OWL XML file with a Python script written for this purpose. 213 of the 281 individuals were created automatically with this script.

4.1 Overview

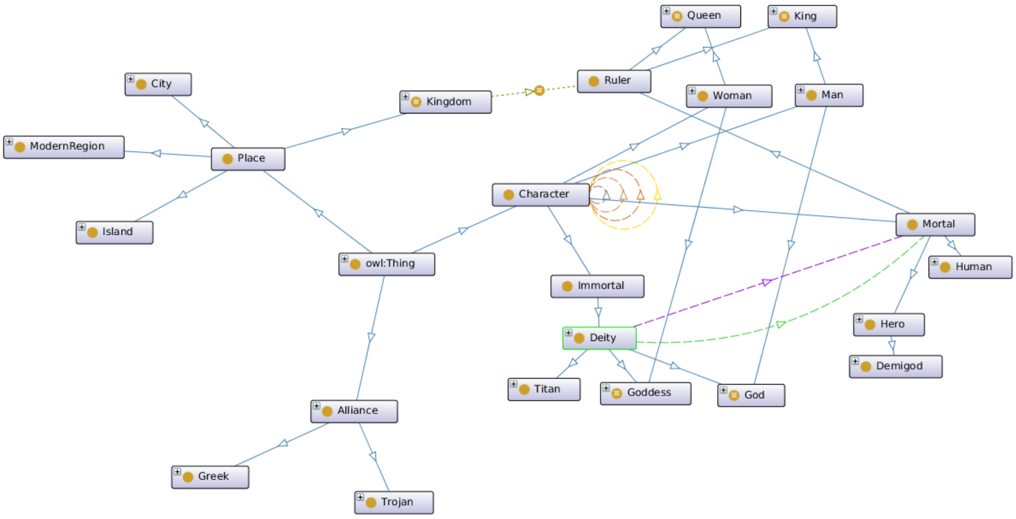
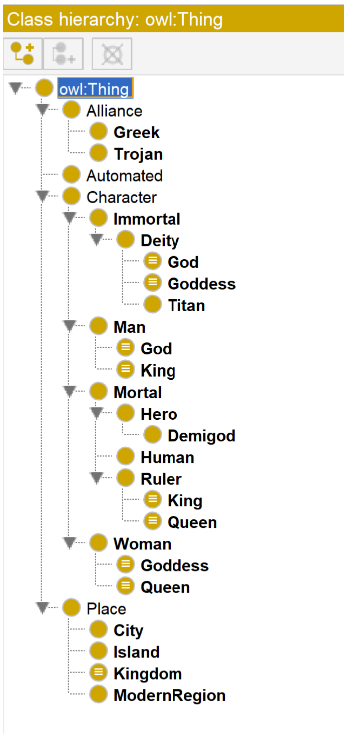


Figure 1: Schema of the Trojan War Ontology

[2] http://johnstoniatexts.x10host.com/homer/iliaddeaths.htm

4.2 Class Hierarchy



Class hierarchy: owl:Thing

- owl:Thing
  - Alliance
    - Greek
    - Trojan
  - Automated
  - Character
    - Immortal
      - Deity
        - God
        - Goddess
        - Titan
    - Man
      - God
      - King
    - Mortal
      - Hero
        - Demigod
      - Human
      - Ruler
        - King
        - Queen
    - Woman
      - Goddess
      - Queen
  - Place
    - City
    - Island
    - Kingdom
    - ModernRegion

Figure 2: Class Hierarchy

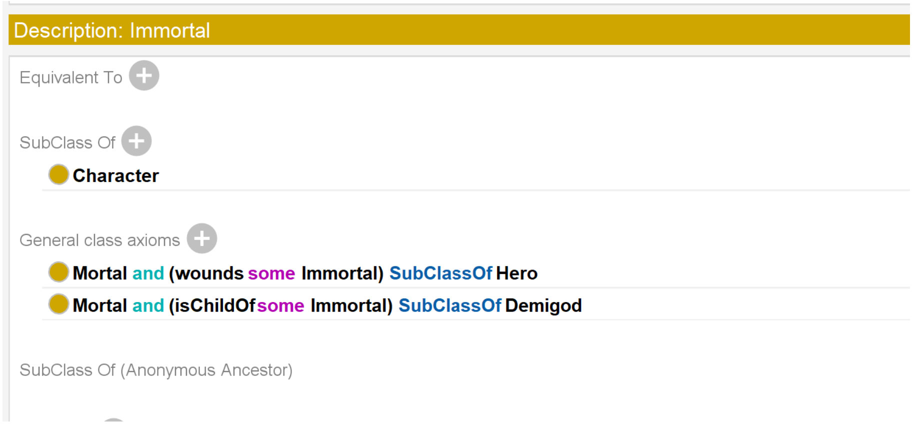
Fig 2 shows the hierarchy of our classes. Some classes are repeated as they subclass more than one parent class, such as `Queen` which is a child class of both `Ruler` and `Woman`. `Automated` just shows individuals that were extracted automatically with our custom scripts, and its serves no purpose other than being a flag.

We also put some contraints on our classes, albeit very few, to ensure that some classes are disjoint:

- `Man` and `Woman` are disjoint
- `Deity` and `Mortal` are disjoint

Finally, it is possible to define classes as equivalents to other classes. We used this property to e.g., model `King` as equivalent to `Ruler and Man`.

4.3 General Class Axioms



Description: Immortal

Equivalent To

SubClass Of

- Character

General class axioms

- Mortal and (wounds some Immortal) SubClassOf Hero
- Mortal and (isChildOfsome Immortal) SubClassOf Demigod

SubClass Of (Anonymous Ancestor)

Figure 3: Class Axioms for Immortal

In addition to constraints, we defined some General Class Axioms to give a deeper semantic description of some classes. Using only class hierarchy is not enough to model certain relationships between classes. For instance, any `Mortal` who can `wound` an `Immortal` must be a `Hero` (Fig 3), as this is indeed a super-human feat.

Similarly, `King` is a subclass of `Man`, but it can also be modeled as a man who rules a `Kingdom`, as seen in the examples below. The use of `SubClassOf` instead of `EquivalentTo:` should be noted, as harming an immortal not the only reason someone is defined a hero.

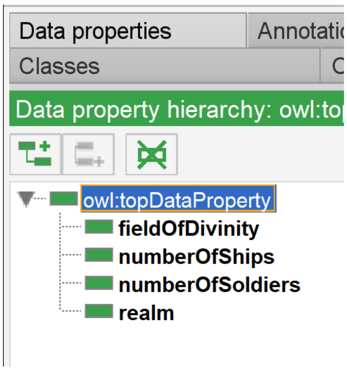
- `isRuledBy some Greek SubClassOf: Greek`
- `isRuledBy some Trojan SubClassOf: Trojan`
- `Mortal and (wounds some Immortal) SubClassOf: Hero`
- `Man and (rulesOver some Kingdom) EquivalentTo: King`
- `Mortal and (isChildOf some Immortal) SubClassOf: Demigod`

4.4 Data Properties

In order to model the Trojan War, using object relations allowed us to define object-object relations. However, some objects also need literals as their properties, like the number of ships each king commands in the Catalogue of Ships.

We created the following data properties:

- `fieldOfDivinity` with domain `Deity` and range `xsd:int`
- `numberOfShips` with domain `Kingdom` and range `xsd:int`
- `numberOfSoldiers` with domain `Kingdom` and range `xsd:int`
- `realm` with domain `Kingdom` and range `xsd:int`



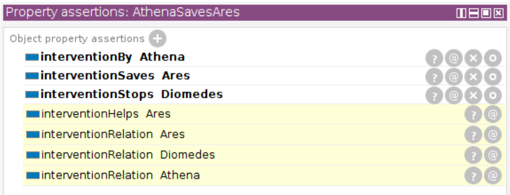
Data property hierarchy: owl:to

- owl:topDataProperty
  - fieldOfDivinity
  - numberOfShips
  - numberOfSoldiers
  - realm

Figure 4: Data Properties Hierarchy

4.5 Object Properties

The table 1 shows all object properties we defined, together with their respective domains, ranges, inverses and an example instantiation for each. There are some additional constraints that are not listed in the table, namly that `blesses` and `curses` are disjoint, as well as `hates` and `loves`.



Property assertions: AthenaSavesAres

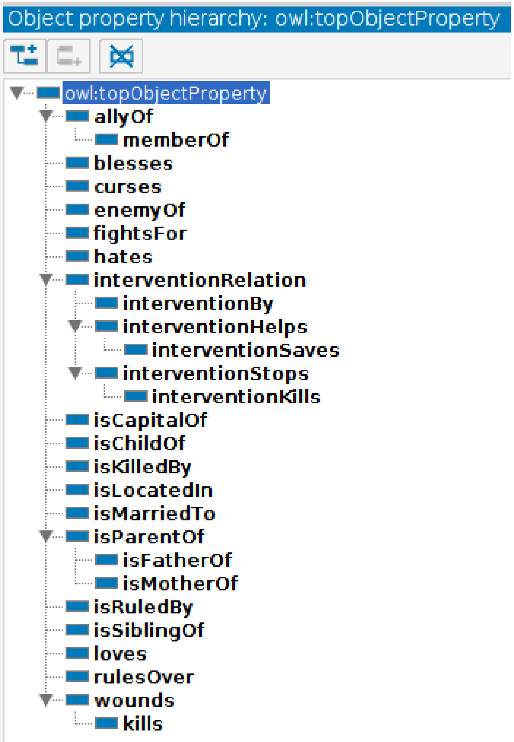
Object property assertions

interventionBy Athena
interventionSaves Ares
interventionStops Diomedes
interventionHelps Ares
interventionRelation Ares
interventionRelation Diomedes
interventionRelation Athena

Figure 5: `AthenaSavesAres` instantiates `DivineIntervention`

Object Property	**Domain**	**Range**	**Inverse of**	**Characteristics**	**Subproperty of**	**Example**
allyOf	Character	Alliance or Kingdom				(Apollo,Trojans)
memberOf	Mortal	Alliance or Kingdom			allyOf	(Achilles,Greeks)
blesses	Deity	Mortal				(Apollo,Paris)
curses	Deity	Mortal				(Athena,Paris)
enemyOf	Character	Character		symmetric		
fightsFor	Character	Character		symmetric		
hates	Character					
loves						(Helen,Paris)
isCapitalOf	City	Alliance or Kingdom				
isChildOf			isParentOf			(Artemis,Zeus)
isKilledBy	Character	Character	kills	functional		(Abas,Diomedes)
isLocatedIn	Place and (not(ModernRegion))	ModernRegion				(Sparta,Greece)
isMarriedTo				symmetric		(Helen,Menelaus)
isParentOf			isChildOf			
isFatherOf	Man				isParentOf	(Priam,Hector)
isMotherOf	Woman				isParentOf	(Rhea,Hera)
isRuledBy	Kingdom and Alliance	Ruler	rulesOver			(Sparta,Menelaus)
isSiblingOf						(Zeus,Hera)
rulesOver	Ruler	Kingdom and Alliance	isRuledBy			(Priam,Troy)
wounds	Character	Character				(Diomedes,Ares)
kills	Character	Character	isKilledBy	inverse functional	wounds	(Paris,Achilles)
interventionRelation	DivineIntervention	Character				(AthenaSavesAres,Athena)
interventionBy	DivineIntervention	Deity			interventionRelation	(AthenaSavesAres,Athena)
interventionHelps	DivineIntervention	Deity			interventionRelation	(AthenaSavesAres,Ares)
interventionSaves	DivineIntervention	Deity			interventionHelps	(AthenaSavesAres,Ares)
interventionStops	DivineIntervention	Deity			interventionRelation	(AthenaSavesAres,Diomedes)
interventionKills	DivineIntervention	Deity			interventionStops	
kills	Character	Character	isKilledBy	inverse functional	wounds	(Paris,Achilles)
kills	Character	Character	isKilledBy	inverse functional	wounds	(Paris,Achilles)
kills	Character	Character	isKilledBy	inverse functional	wounds	(Paris,Achilles)

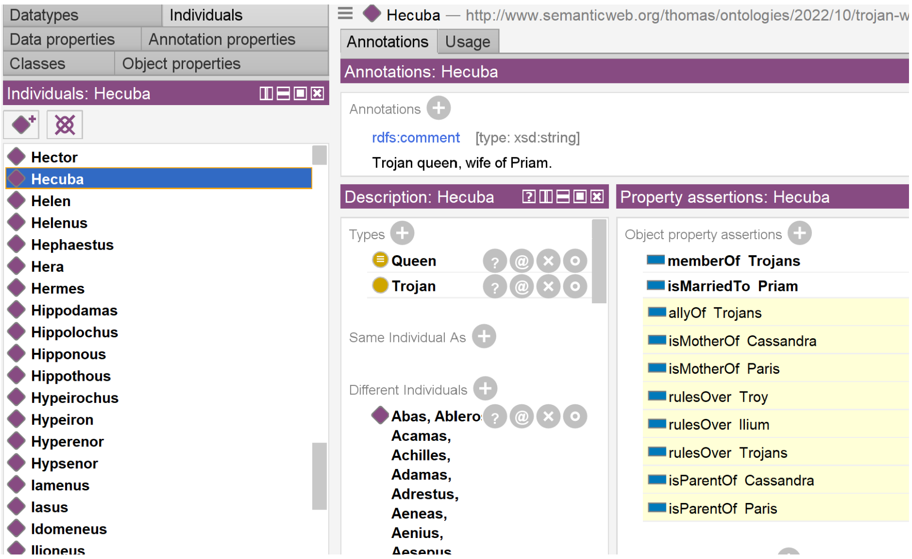
Table 1: Object Properties



Object property hierarchy: owl:topObjectProperty

```
owl:topObjectProperty
    allyOf
        memberOf
    blesses
    curses
    enemyOf
    fightsFor
    hates
    interventionRelation
        interventionBy
        interventionHelps
            interventionSaves
        interventionStops
            interventionKills
    isCapitalOf
    isChildOf
    isKilledBy
    isLocatedIn
    isMarriedTo
    isParentOf
        isFatherOf
        isMotherOf
    isRuledBy
    isSiblingOf
    loves
    rulesOver
    wounds
        kills
```

Figure 6: Object Properties Hierarchy



Hector
Hecuba
Helen
Helenus
Hephaestus
Hera
Hermes
Hippodamas
Hippolochus
Hipponous
Hippothous
Hypeirochus
Hypeiron
Hyperenor
Hypsenor
Iamenus
Iasus
Idomeneus
Ilioneus

rdfs:comment [type: xsd:string]
Trojan queen, wife of Priam.

Types: Queen, Trojan

Different Individuals: Abas, Ablero, Acamas, Achilles, Adamas, Adrestus, Aeneas, Aenius, Aesepus

memberOf Trojans
isMarriedTo Priam
allyOf Trojans
isMotherOf Cassandra
isMotherOf Paris
rulesOver Troy
rulesOver Ilium
rulesOver Trojans
isParentOf Cassandra
isParentOf Paris

Figure 7: Inferenced relations from HermiT for Hecuba

5 Inferences and Queries

Having the ontology set up and populated with meaningful instances and relations, we can run the built-in reasoner to infer knowledge from the given information and the **inference rules**. Simple examples are given in the next Figure.

$$\begin{aligned}\texttt{isKilledBy(x,y)} &\implies \texttt{kills(y,x)}\\ \texttt{isSiblingOf(x,y)} &\implies \texttt{isSiblingOf(y,x)}\\ \texttt{kills(x,y)} &\implies \texttt{wounds(x,y)}\end{aligned}$$

Figure 8: Simple inferences that can directly be inferred from the previously defined object properties (e.g., inverse or symmetric properties) or the defined property structure (e.g., sub-properties imply the super-properties).

From the defined inference rules, we can assert that `isSiblingOf` is a symmetric, transitive and euclidean relationship. We also have examples of anti-symmetric relationships (such as `rulesOver` and `isRuledBy`), and examples of sub-properties (such as `kills` being a sub-property of `wounds`).

Apart from these fairly easy derivable inferences, we can also make the reasoner infer more advanced information by defining our own inference rules. This can be achieved by using the Semantic Web Rule Language (SWRL[3]) that is built into Protegé. We can thus define rules of the structure shown in Figure 9.

The inferred information becomes especially powerful when the reasoner can apply several inference rules in a sequence and thus acquire even more knowledge about the instances. An example is shown in Figure 10.

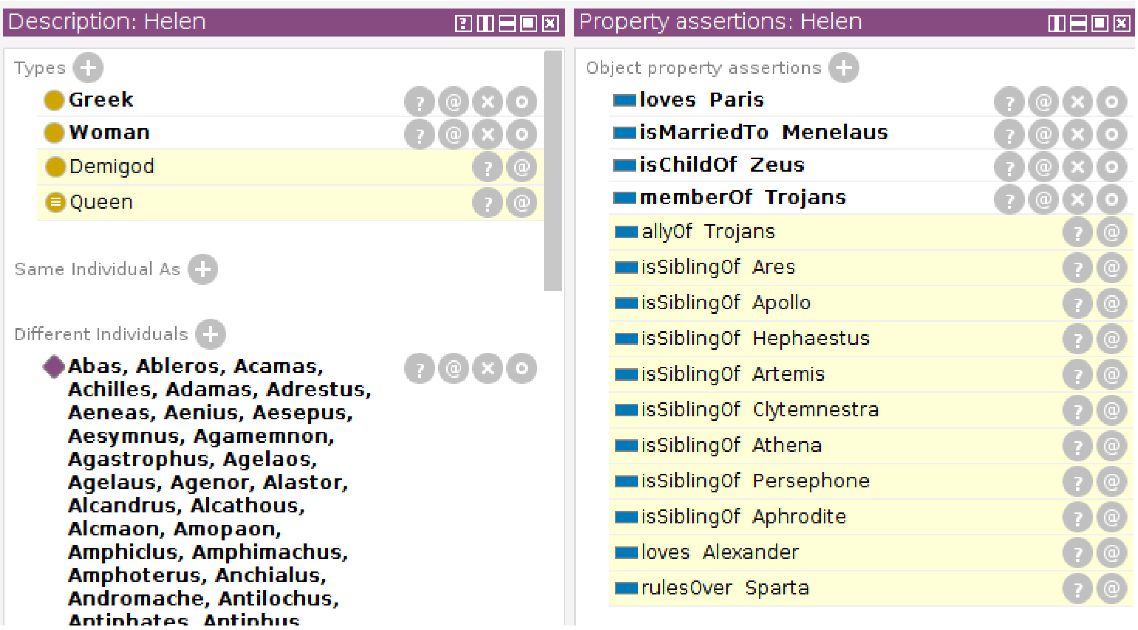
[3] https://www.w3.org/Submission/SWRL/

If x and y are both children of z, we can infer that x and y are siblings.
`isChildOf(x,z)` $\wedge$ `isChildOf(y,z)` $\wedge$ `(x≠y)` $\implies$ `isSiblingOf(x,y)`

If x is married to y who rules over z, we can infer that x also rules over z.
`isMarriedTo(x,y)` $\wedge$ `rulesOver(y,z)` $\implies$ `rulesOver(x,z)`

If x is the parent of y and is a Woman, we can infer that x is the mother of y.
`isParentOf(x,y)` $\wedge$ `Woman(x)` $\implies$ `isMotherOf(x,y)`

Figure 9: More complex inference rules that can use classes of instances can be defined using the SWRL.



Description: Helen

Types
- ● Greek
- ● Woman
- ● Demigod
- ● Queen

Same Individual As

Different Individuals
- ◆ Abas, Ableros, Acamas, Achilles, Adamas, Adrestus, Aeneas, Aenius, Aesepus, Aesymnus, Agamemnon, Agastrophus, Agelaos, Agelaus, Agenor, Alastor, Alcandrus, Alcathous, Alcmaon, Amopaon, Amphiclus, Amphimachus, Amphoterus, Anchialus, Andromache, Antilochus, Antiphates, Antiphus

Property assertions: Helen

Object property assertions
- loves Paris
- isMarriedTo Menelaus
- isChildOf Zeus
- memberOf Trojans
- allyOf Trojans
- isSiblingOf Ares
- isSiblingOf Apollo
- isSiblingOf Hephaestus
- isSiblingOf Artemis
- isSiblingOf Clytemnestra
- isSiblingOf Athena
- isSiblingOf Persephone
- isSiblingOf Aphrodite
- loves Alexander
- rulesOver Sparta

Figure 10: For Helen, we can infer that she is the daughter of Zeus. That is why she has the `isSiblingOf` relation to all of Zeus' children in the ontology. However, as her class is Greek which is a subclass of Mortal, she gets assigned to the Demigod class (as shown in the General Class Axiom above). Another interesting inference is the `rulesOver` attribute (and thus also the class association Queen) that is inferred since she is the wife of Menelaus.

Another powerful tool that can be used by the reasoner to infer knowledge from the given instances are the **General Class Axioms**. These axioms can be used to assign specific instances to classes when they meet the defined criteria. The General Class Axioms defined in our ontology are listed in Figure 11.

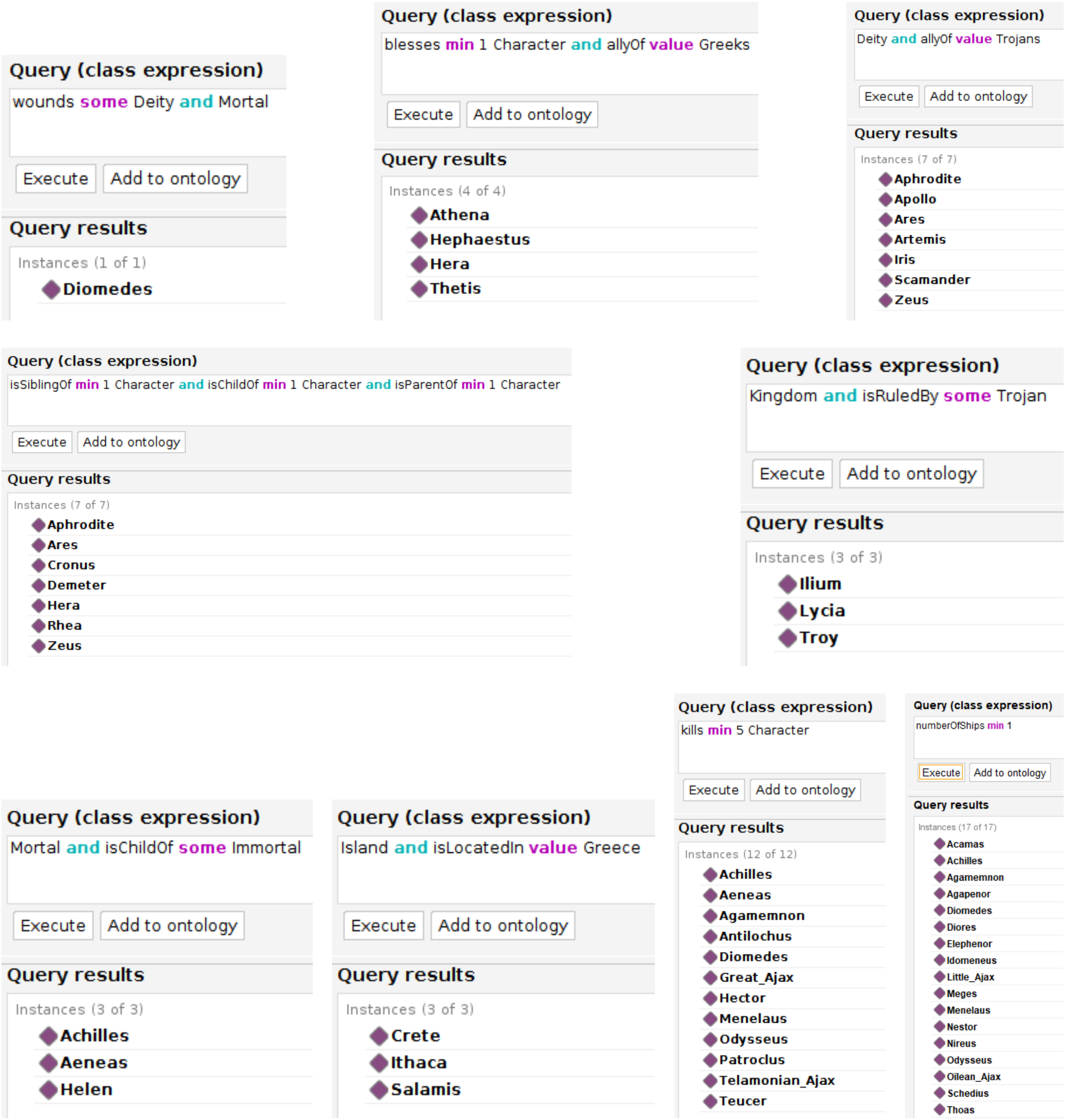
`Mortal(x)` $\wedge$ `Immortal(y)` $\wedge$ `wounds(x,y)` $\implies$ `Hero(x)`
`Mortal(x)` $\wedge$ `Immortal(y)` $\wedge$ `isChildOf(x,y)` $\implies$ `Demigod(x)`
`isRuledBy(x,y)` $\wedge$ `Greek(y)` $\implies$ `Greek(x)`
`isRuledBy(x,y)` $\wedge$ `Trojan(y)` $\implies$ `Trojan(x)`

Figure 11: General Class Axioms help the reasoner infer more information about class associations of instances.

DL Queries

The developed ontology database and the inferred information by the reasoner itself already provide valuable information but by running queries on the ontology database, we can get interesting new insights into the data.

Some examples of interesting queries and their results are shown in the following Figure 12. We can observe that there is only one character in the story who became a hero through wounding a god, namely Diomedes. There exist a few characters who are demigods because they are mortal but they have an immortal parent and we can compute all characters that killed at least 5 characters. This query could now also be transformed in a General Class Axiom to recognize their achievments and associate them with the Hero class. Based on the instances that we inserted, we can also compute deities that are allies of the Greeks and blessed at least one character, characters with much information on their family or different information on kingdoms and their modern day locations.



Query (class expression)

wounds some Deity and Mortal

Query results

Instances (1 of 1)

- Diomedes

Query (class expression)

blesses min 1 Character and allyOf value Greeks

Query results

Instances (4 of 4)

- Athena
- Hephaestus
- Hera
- Thetis

Query (class expression)

Deity and allyOf value Trojans

Query results

Instances (7 of 7)

- Aphrodite
- Apollo
- Ares
- Artemis
- Iris
- Scamander
- Zeus

Query (class expression)

isSiblingOf min 1 Character and isChildOf min 1 Character and isParentOf min 1 Character

Query results

Instances (7 of 7)

- Aphrodite
- Ares
- Cronus
- Demeter
- Hera
- Rhea
- Zeus

Query (class expression)

Kingdom and isRuledBy some Trojan

Query results

Instances (3 of 3)

- Ilium
- Lycia
- Troy

Query (class expression)

Mortal and isChildOf some Immortal

Query results

Instances (3 of 3)

- Achilles
- Aeneas
- Helen

Query (class expression)

Island and isLocatedIn value Greece

Query results

Instances (3 of 3)

- Crete
- Ithaca
- Salamis

Query (class expression)

kills min 5 Character

Query results

Instances (12 of 12)

- Achilles
- Aeneas
- Agamemnon
- Antilochus
- Diomedes
- Great_Ajax
- Hector
- Menelaus
- Odysseus
- Patroclus
- Telamonian_Ajax
- Teucer

Query (class expression)

numberOfShips min 1

Query results

Instances (17 of 17)

- Acamas
- Achilles
- Agamemnon
- Agapenor
- Diomedes
- Diores
- Elephenor
- Idomeneus
- Little_Ajax
- Meges
- Menelaus
- Nestor
- Nireus
- Odysseus
- Oilean_Ajax
- Schedius
- Thoas

Figure 12: Different queries on the created ontology database.

References

[1] Angela Carrera-Rivera. Ux- for smart-pss: Towards a context-aware framework. 10 2022. doi: 10.5220/0011379700003323.

[2] Protege. https://protege.stanford.edu/. Accessed: 2022-11-11.