

Reasoning and querying examples using Protégé tool with transformed AVCL missions

Curt Blais and Don Brutzman
Naval Postgraduate School (NPS)

1 March 2020

PiratesSeizingMerchantDefense mission: Turtle syntax read into the Protégé ontology tool

The screenshot shows the Protégé ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The main window is titled 'missions (https://www.nps.edu/ontologies/MissionExecutionOntology/missions)' and shows the 'Ontology header' tab. The 'Ontology IRI' is 'https://www.nps.edu/ontologies/MissionExecutionOntology/missions' and the 'Ontology Version IRI' is 'e.g. https://www.nps.edu/ontologies/MissionExecutionOntology/missions/1.0.0'. The 'Ontology metrics' panel on the right shows the following data:

Metrics	
Axiom	403
Logical axiom count	245
Declaration axioms count	79
Class count	11
Object property count	19
Data property count	0
Individual count	50
Annotation Property count	2

The 'Class axioms' section shows:

Class axioms	
SubClassOf	21
EquivalentClasses	1
DisjointClasses	2
GCI count	0
Hidden GCI Count	0

The 'Object property axioms' section shows:

Object property axioms	
SubObjectPropertyOf	5
EquivalentObjectProperties	0
InverseObjectProperties	1
DisjointObjectProperties	0

The 'Imported ontologies' panel shows a direct import of 'MissionExecutionOntology' from 'https://www.nps.edu/ontologies/MissionExecutionOntology' with the location 'C:\Users\curt\Documents\EthicalControl\ontologies\MissionExecutionOntology.rdf'. The bottom status bar indicates 'Git: master' and 'To use the reasoner click Reasoner > Start reasoner'.

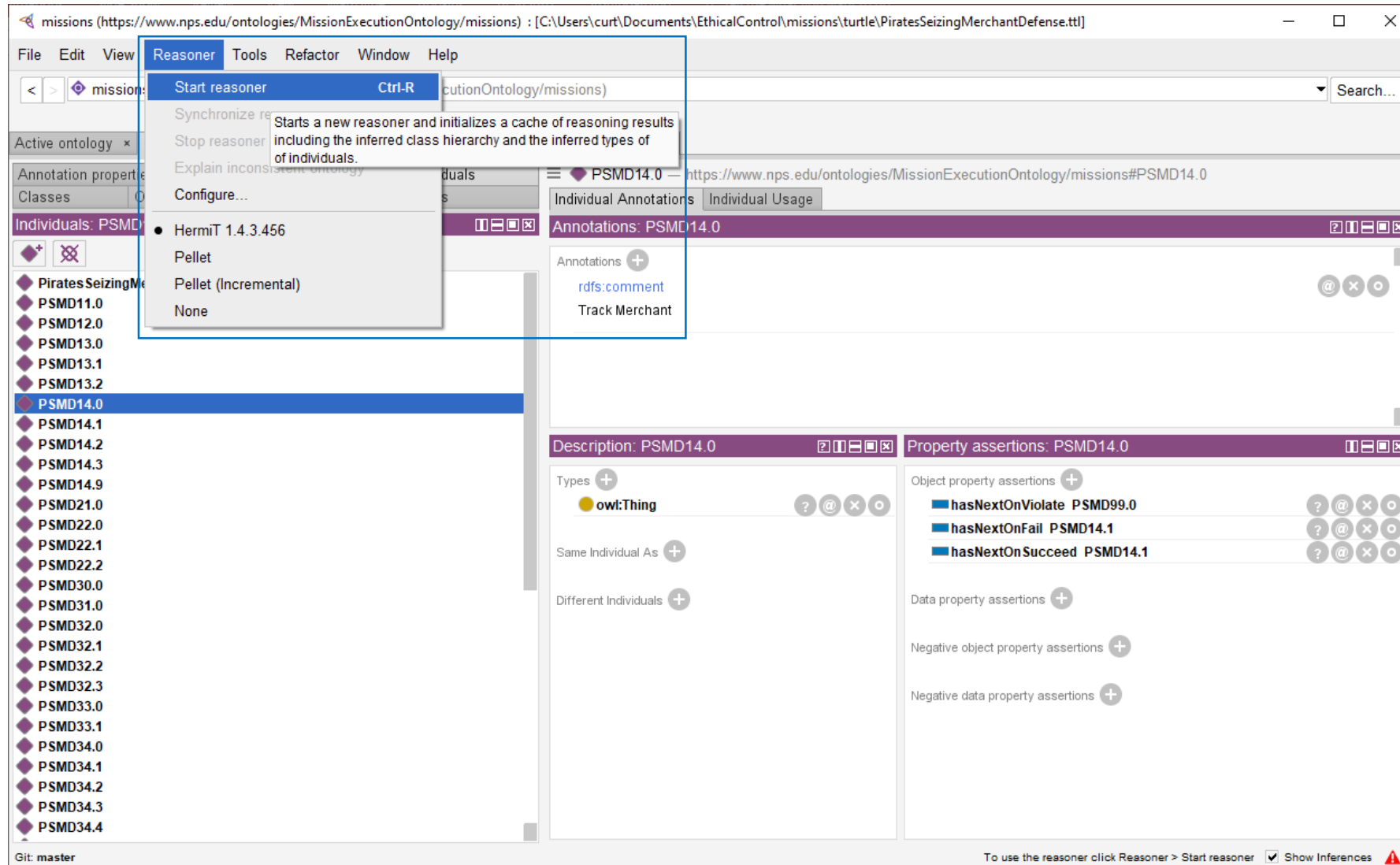
Note: The base relationships ontology (MissionExecutionOntology) has 133 axioms, so the other 270 are from the PiratesSeizingMerchantDefense mission expressed in the ontology.

PiratesSeizingMerchantDefense mission: Turtle syntax read using Protégé ontology tool

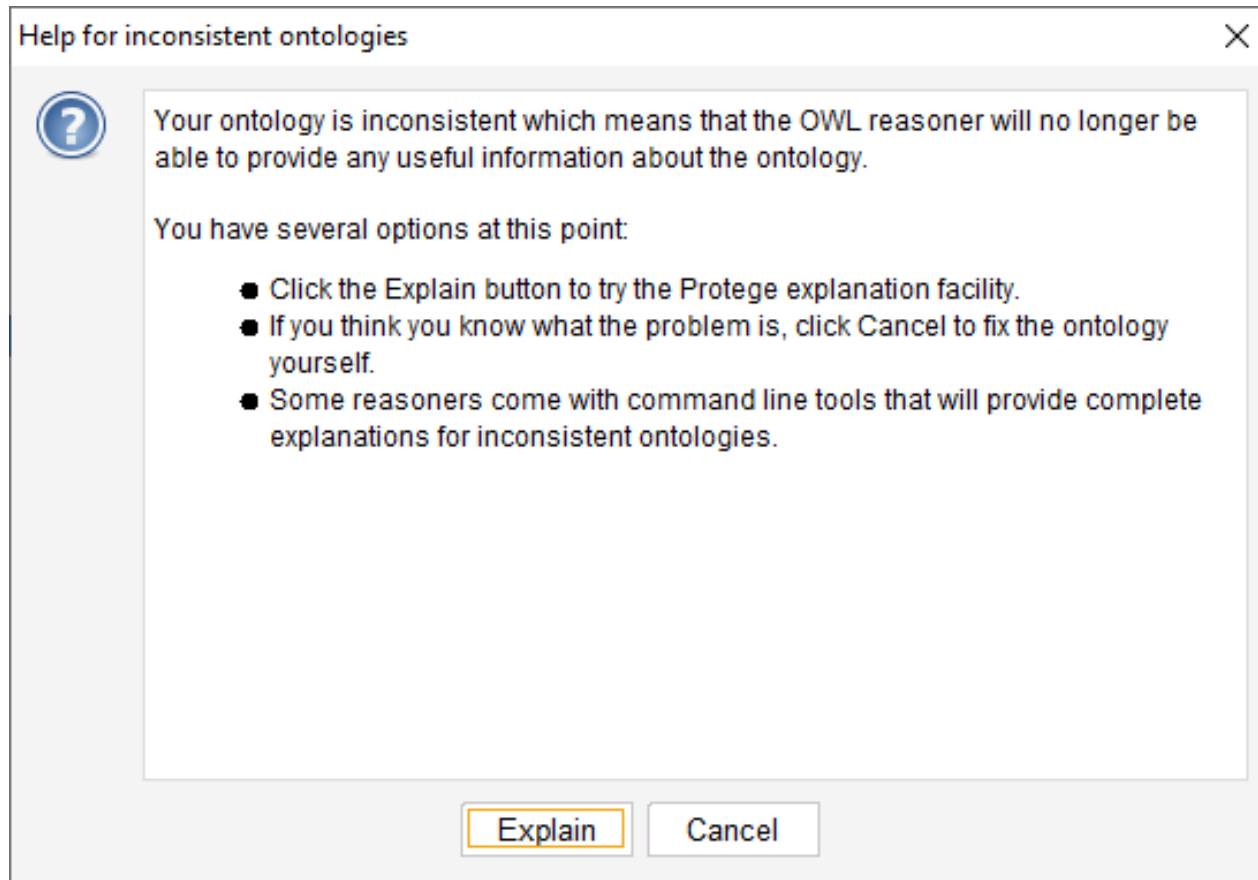
The screenshot displays the Protégé ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The address bar shows the URL: [missions \(https://www.nps.edu/ontologies/MissionExecutionOntology/missions\)](https://www.nps.edu/ontologies/MissionExecutionOntology/missions). The left pane, titled "Individuals: PSMD14.0", lists various individuals, including PiratesSeizingMerchantDefense, PSMD11.0, PSMD12.0, PSMD13.0, PSMD13.1, PSMD13.2, PSMD14.0 (selected), PSMD14.1, PSMD14.2, PSMD14.3, PSMD14.9, PSMD21.0, PSMD22.0, PSMD22.1, PSMD22.2, PSMD30.0, PSMD31.0, PSMD32.0, PSMD32.1, PSMD32.2, PSMD32.3, PSMD33.0, PSMD33.1, PSMD34.0, PSMD34.1, PSMD34.2, PSMD34.3, and PSMD34.4. A blue box highlights the selected individual, PSMD14.0, with the text: "Protégé lists the individuals (e.g., goals, mission) and associated axioms asserted in the mission ontology." The right pane, titled "PSMD14.0 — https://www.nps.edu/ontologies/MissionExecutionOntology/missions#PSMD14.0", shows the description and property assertions for the selected individual. The description pane shows the type "owl:Thing". The property assertions pane shows three assertions: "hasNextOnViolate PSMD99.0", "hasNextOnFail PSMD14.1", and "hasNextOnSucceed PSMD14.1".

Protégé lists the individuals (e.g., goals, mission) and associated axioms asserted in the mission ontology.

Start a reasoner (here, HermiT) to check validity of the *Mission* against the MissionExecutionOntology base ontology



Reasoner indicates that ontology has inconsistencies, provides path to check explanation of those findings.



The next slide shows even further diagnostic detail provided by Protégé.

Not receiving any warnings is good!

Corollary:
the worst-case error is an undiscovered error.

Even worse: later repeating that previously corrected error, without detecting recurrence.

TODO: figure out why offline ARQ validation isn't identifying every problem as expected.

Reasoner explanations indicate that two goals (id PMSD35.2 and PSMD90.0) have properties that violated the irreflexive condition (a goal cannot immediately follow itself)

The screenshot shows a window titled "Inconsistent ontology explanation" with a close button (X) in the top right corner. It contains two sections, "Explanation 1" and "Explanation 2", each with a checkbox for "Display laconic explanation".

Explanation 1: owl:Thing SubClassOf owl:Nothing

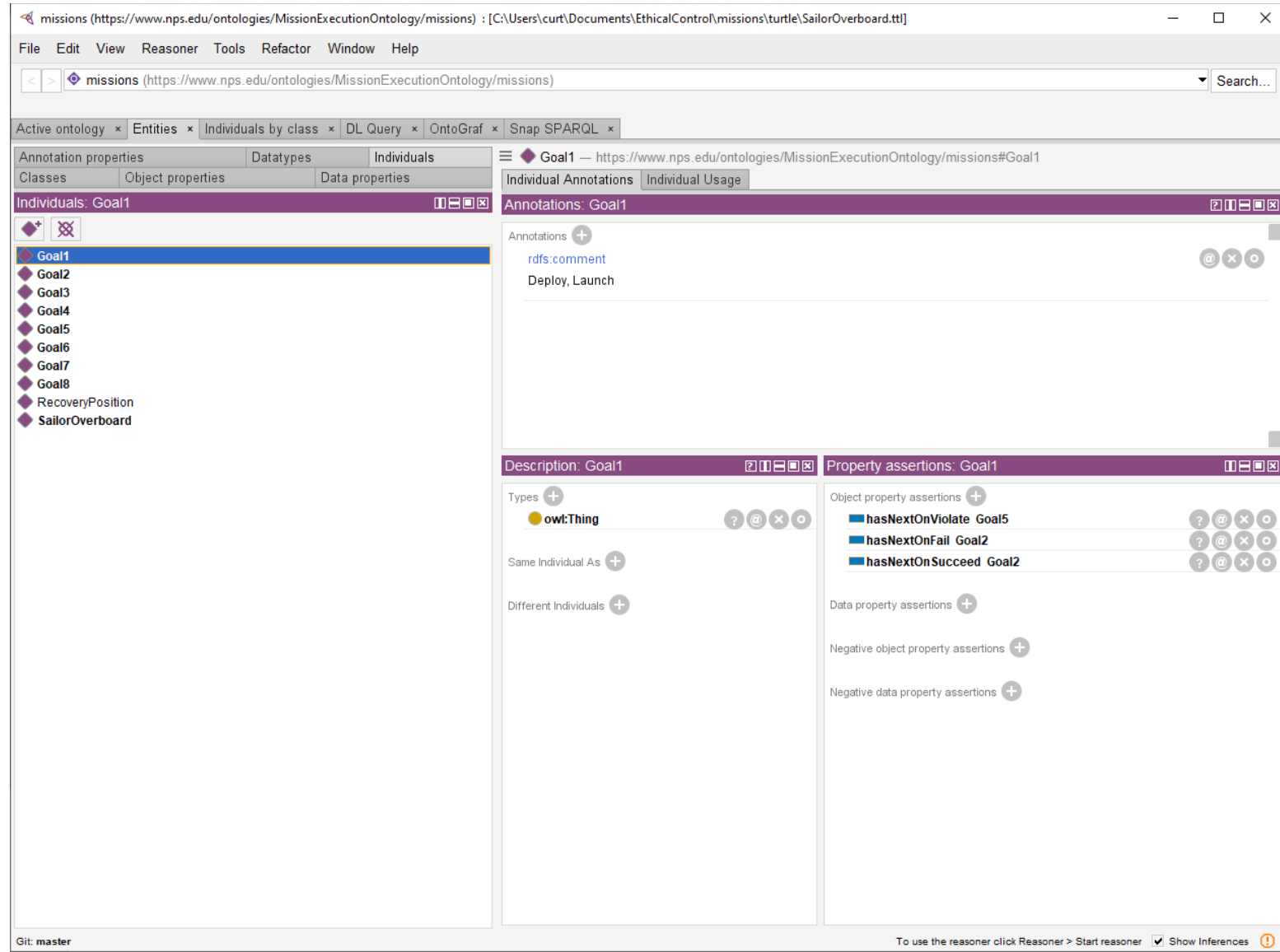
- 1) **PMSD35.2** **hasNextOnFail** PMSD35.2 In NO other justifications ?
- 2) **Irreflexive: hasNext** In ALL other justifications ?
- 3) **hasNextOnFail** **SubPropertyOf: hasNext** In NO other justifications ?

Explanation 2: owl:Thing SubClassOf owl:Nothing

- 1) **hasNextOnViolate** **SubPropertyOf: hasNext** In NO other justifications ?
- 2) **PSMD90.0** **hasNextOnViolate** PSMD90.0 In NO other justifications ?
- 3) **Irreflexive: hasNext** In ALL other justifications ?

The reasoner identified two errors for correction in the original AVCL mission that were not diagnosed otherwise! This is important milestone of progress. The original mission mistakes causing these errors have since been corrected.

Consider the SailorOverboard *Mission*: here it has been loaded into Protégé



... now run Reasoner to see what can be inferred from the formal *Mission* definition

The screenshot shows the Protégé ontology editor with the 'Reasoner' tab active. The ontology is 'missions' (https://www.nps.edu/ontologies/MissionExecutionOntology/missions). The selected individual is 'Goal1' (https://www.nps.edu/ontologies/MissionExecutionOntology/missions#Goal1). The 'Property assertions: Goal1' panel shows several assertions, with inferred ones highlighted in yellow:

- hasNextOnViolate Goal5
- hasNextOnFail Goal2
- hasNextOnSucceed Goal2
- hasNext Goal2
- hasNext Goal5
- isFollowedBy Goal3
- isFollowedBy Goal2
- isFollowedBy RecoveryPosition
- isFollowedBy Goal5
- isFollowedBy Goal4
- isFollowedBy Goal6

The 'Types' panel shows 'owl:Thing' and 'Goal' as types for 'Goal1'. The 'Inferred axioms are highlighted' callout points to the highlighted assertions in the 'Property assertions' panel.

Inferred axioms
are highlighted

With the reasoner running, can use DLQuery to interrogate the ontology

The screenshot shows the Protégé DLQuery interface. The left pane displays the class hierarchy for 'Goal', with 'Goal' selected. The main pane shows a DL query: 'hasEndCondition some EndCondition'. Below the query, the 'Query results' section shows 'Instances (0 of 0)'. Two blue callout boxes provide context: one points to the query expression stating 'i.e., find any individuals in the mission that have an EndCondition', and the other points to the 'Instances (0 of 0)' result stating 'no individuals (i.e. “instances”) were found meeting this condition'. The bottom status bar indicates 'Reasoner active' and 'Show Inferences' is checked.

missions (https://www.nps.edu/ontologies/MissionExecutionOntology/missions) : [C:\Users\curt\Documents\EthicalControl\missions\turtle\SailorOverboard.ttl]

File Edit View Reasoner Tools Refactor Window Help

missions (https://www.nps.edu/ontologies/MissionExecutionOntology/missions) Search...

Active ontology x Entities x Individuals by class x DL Query x OntoGraf x Snap SPARQL x

Class hierarchy: Goal

- owl:Thing
 - Constraint
 - EndCondition
 - Goal
 - GoalRequirement
 - Mission
 - Vehicle
 - VehicleFeature

DL query:

Query (class expression)

hasEndCondition some EndCondition

Execute Add to ontology

Query results

Instances (0 of 0)

Query for

Result filters

Name contains

☒ Instances

☒ Display owl:Thing (in superclass results)

☒ Display owl:Nothing (in subclass results)

Git: master Reasoner active Show Inferences

Here, none of the Goal individuals in the mission ontology satisfy the property constraint “hasEndCondition some EndCondition”. The mission definition does not satisfy some of the initial design ideas encoded into the base ontology rules.

TODO: review, fix!

Can also use SPARQL for such queries, which can be executed inside or outside a tool like Protégé.

SPARQL query: check *Mission* to find initial *Goal*

missions (https://www.nps.edu/ontologies/MissionExecutionOntology/missions) : [C:\Users\curt\Documents\EthicalControl\missions\turtle\SailorOverboard.ttl]

File Edit View Reasoner Tools Refactor Window Help

< > missions (https://www.nps.edu/ontologies/MissionExecutionOntology/missions) Search...

requires

Active ontology × Entities × Individuals by class × DL Query × OntoGraf × Snap SPARQL ×

Snap SPARQL Query:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX meo: <https://www.nps.edu/ontologies/MissionExecutionOntology#>

# Query to find any individuals in the ontology file that are related to some goal through the startsWith property.

SELECT ?Mission ?Goal WHERE {
    ?mission meo:startsWith ?goal .
    BIND (strafter(xsd:string(?mission), "#") AS ?Mission)
    BIND (strafter(xsd:string(?goal), "#") AS ?Goal)
}
```

Execute

?Mission	?Goal
SailorOverboard ^{^^} xsd:string	Goal1 ^{^^} xsd:string

1 results

Git: master Reasoner active ☒ Show Inferences

SPARQL query: Find *Goal* individuals linked to other *Goal* individuals through *hasNextOnSucceed* property

missions (https://www.nps.edu/ontologies/MissionExecutionOntology/missions) : [C:\Users\curt\Documents\EthicalControl\missions\turtle\SailorOverboard.ttl]

File Edit View Reasoner Tools Refactor Window Help

< > missions (https://www.nps.edu/ontologies/MissionExecutionOntology/missions) Search...

> requires

Active ontology x Entities x Individuals by class x DL Query x OntoGraf x Snap SPARQL x

Snap SPARQL Query:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX meo: <https://www.nps.edu/ontologies/MissionExecutionOntology#>

# Query to find any goals in the ontology file that are related to some other goal(s) through the hasNextOnSucceed property.

SELECT ?Goal ?HasNextOnSucceed WHERE {
  ?goal a meo:Goal .
  ?linkedToGoal a meo:Goal .
  ?goal meo:hasNextOnSucceed ?linkedToGoal .
  BIND (strafter(xsd:string(?goal), "#") AS ?Goal)
  BIND (strafter(xsd:string(?linkedToGoal), "#") AS ?HasNextOnSucceed)
}
```

Execute

?Goal	?HasNextOnSucceed
Goal1^^xsd:string	Goal2^^xsd:string
Goal3^^xsd:string	Goal4^^xsd:string
Goal2^^xsd:string	Goal4^^xsd:string
Goal5^^xsd:string	RecoveryPosition^^xsd:string
Goal4^^xsd:string	Goal5^^xsd:string

5 results


Git: master Reasoner active ☒ Show Inferences

missions (https://www.nps.edu/ontologies/MissionExecutionOntology/missions) : [C:\Users\curt\Documents\EthicalControl\missions\turtle\SailorOverboard.ttl]

File Edit View Reasoner Tools Refactor Window Help

< > missions (https://www.nps.edu/ontologies/MissionExecutionOntology/missions) Search...

Active ontology x Entities x Individuals by class x DL Query x OntoGraf x Snap SPARQL x

Snap SPARQL Query: 

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX meo: <https://www.nps.edu/ontologies/MissionExecutionOntology#>


# Count the number of goal individuals a mission starts with (anything other than 1 is an error condition).

SELECT (COUNT(?startGoal) as ?numberOfStartingGoals) WHERE {
    ?mission meo:startsWith ?startGoal .
    OPTIONAL {?x rdfs:label ?lab}
}
ORDER BY ?startGoal
```

Execute

?numberOfStartingGoals
1

1 results

Git: master Reasoner active ☒ Show Inferences 

As discussed with Rich Markeloff, proper construction of this query can be difficult.

Alternatively, and perhaps better, if we make the startsWith property functional (a Mission can only relate to a single Goal individual through the startsWith property), then a mission with two distinct individuals related to the mission by the startsWith property causes the reasoner to infer the ontology is inconsistent.

missions (https://www.nps.edu/ontologies/MissionExecutionOntology/missions) : [C:\Users\curt\Documents\EthicalControl\missions\turtle\SailorOverboard.ttl]

File Edit View Reasoner Tools Refactor Window Help

< > missions (https://www.nps.edu/ontologies/MissionExecutionOntology/missions) Search...

Active ontology x Entities x Individuals by class x DL Query x OntoGraf x Snap SPARQL x

Snap SPARQL Query: ⏏ ⏏ ⏏ ⏏

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX meo: <https://www.nps.edu/ontologies/MissionExecutionOntology#>
PREFIX : <https://www.nps.edu/ontologies/MissionExecutionOntology/missions#>

# Find goal individuals in the sequence of goals in the mission that have chains leading back to themselves.

SELECT ?loopGoal
WHERE {
    ?loopGoal meo:isFollowedBy ?loopGoal .
}
ORDER BY ?loopGoal
```

Execute

?loopGoal
:Goal2
:Goal3
:Goal4
:Goal5

4 results

Git: master Reasoner active ☒ Show Inferences ⏏

Same objective... but here is an alternate formulation that finds follow-on Goals.

We are working through ontology design to ensure that all constructs are testable, unambiguous and widely implementable. Unit testing of missions is essential.

TODO

Many additional queries are possible and planned as work continues. Future updates to this slideset will continue to explore them.

All results remain repeatable and online at

- <https://savage.nps.edu/EthicalControl/#Queries>
- <https://gitlab.nps.edu/Savage/EthicalControl/tree/master/queries>