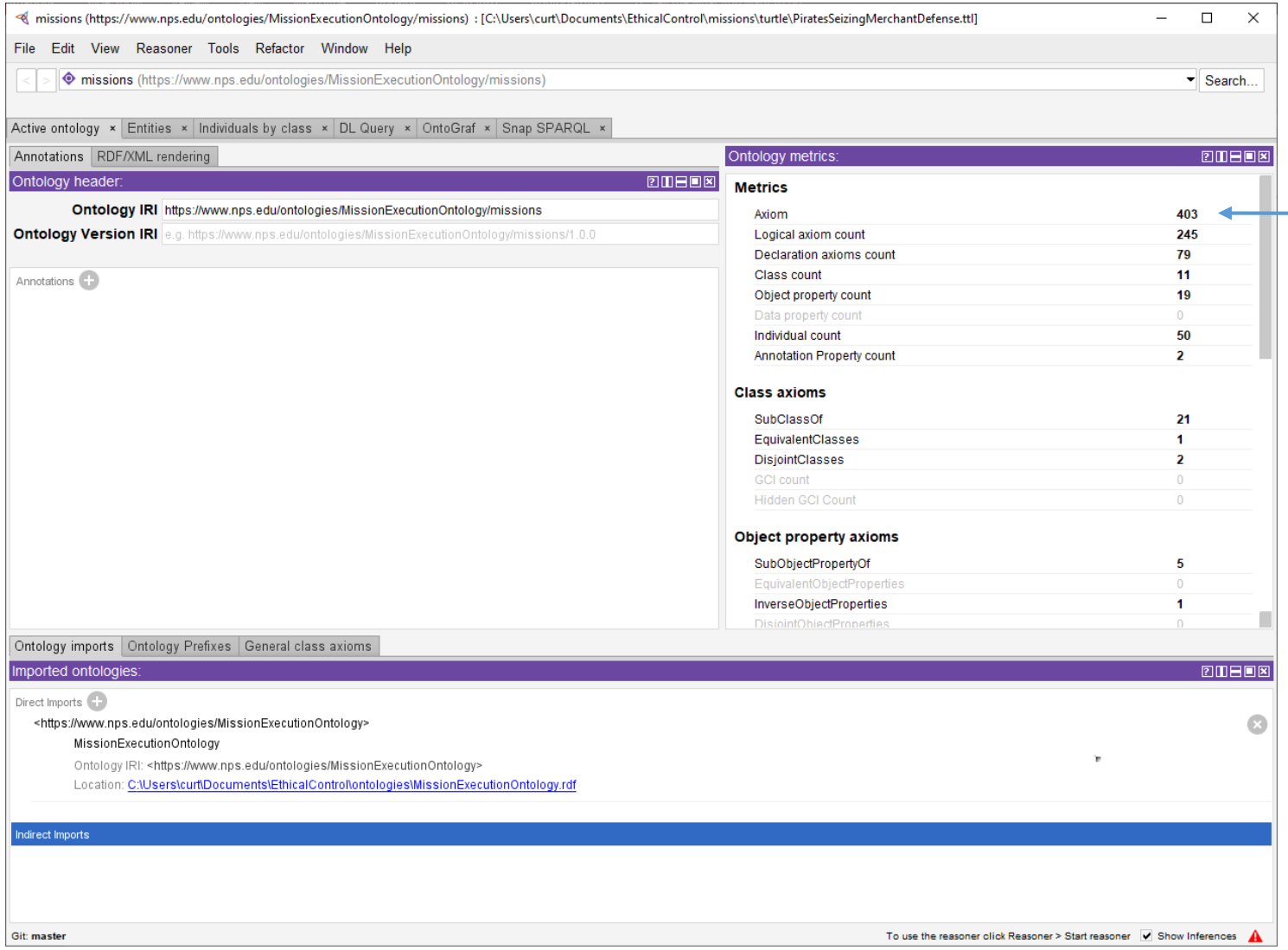
Reasoning and querying examples using Protégé tool with transformed AVCL missions

Curt Blais and Don Brutzman

Naval Postgraduate School (NPS)

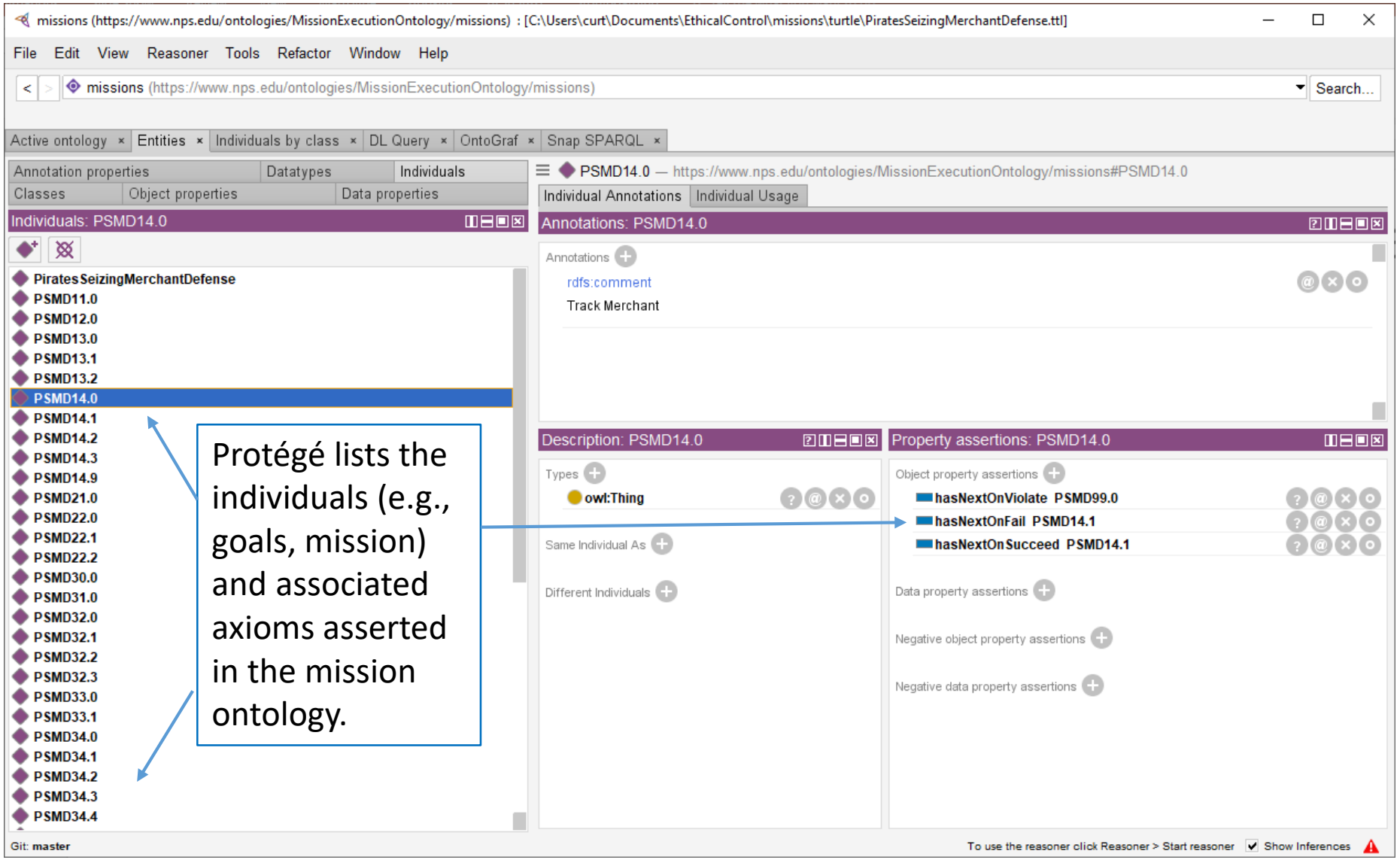
1 March 2020

PiratesSeizingMerchantDefense mission: Turtle syntax read into the Protégé ontology tool

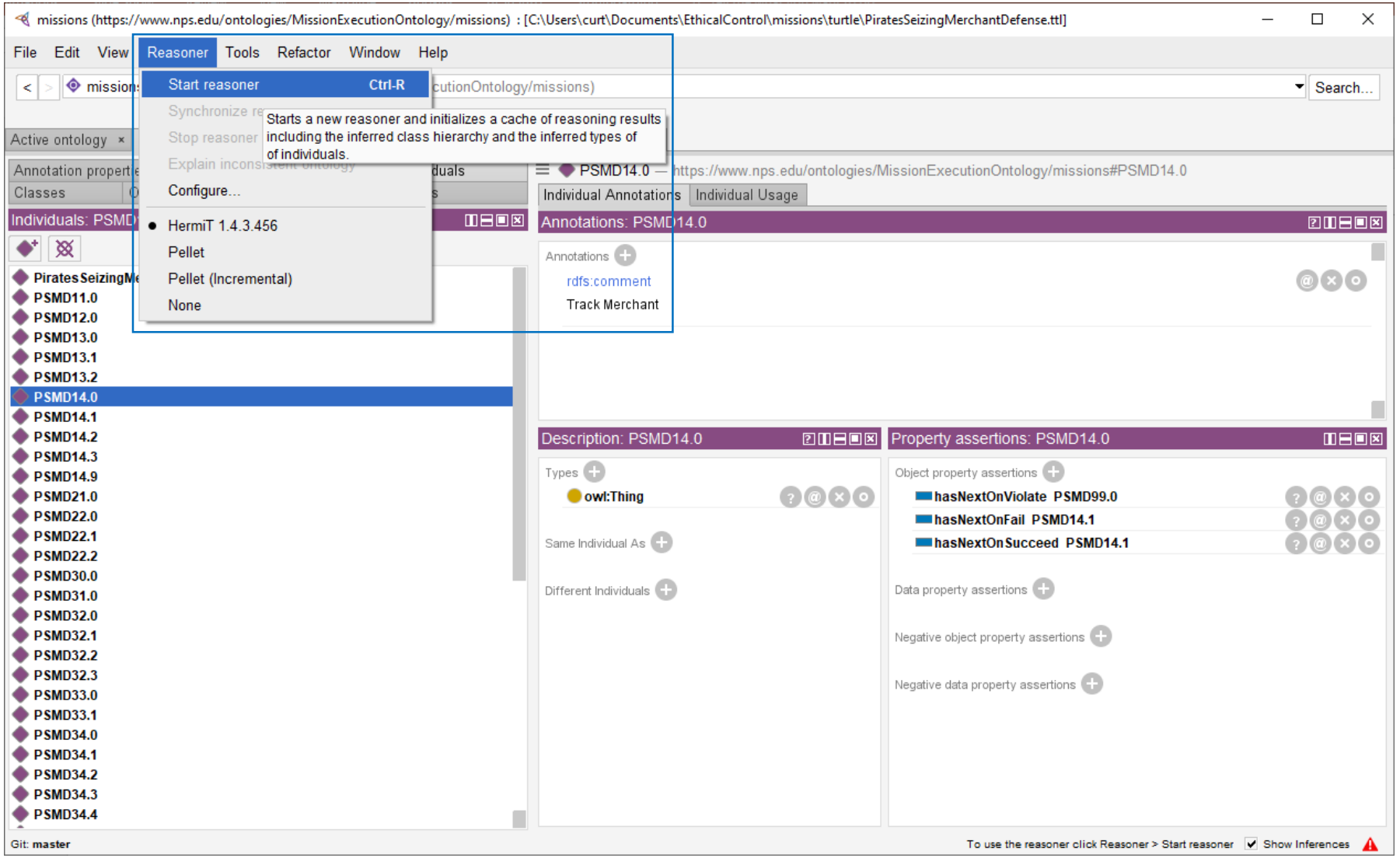


Note: The base relationships ontology (MissionExecutionOntology) has 133 axioms, so the other 270 are from the PiratesSeizingMerchantDefense mission expressed in the ontology.

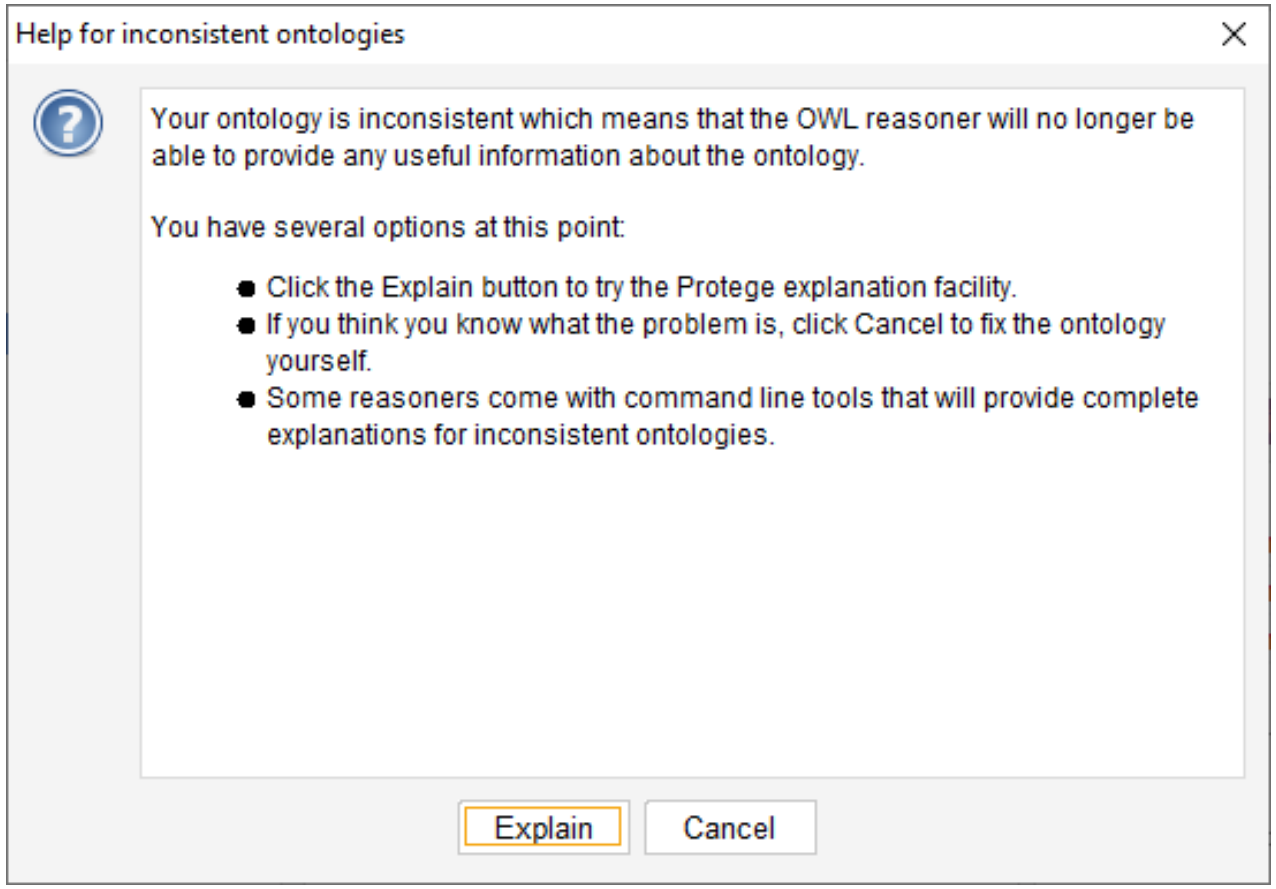
PiratesSeizingMerchantDefense mission: Turtle syntax read using Protégé ontology tool



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.

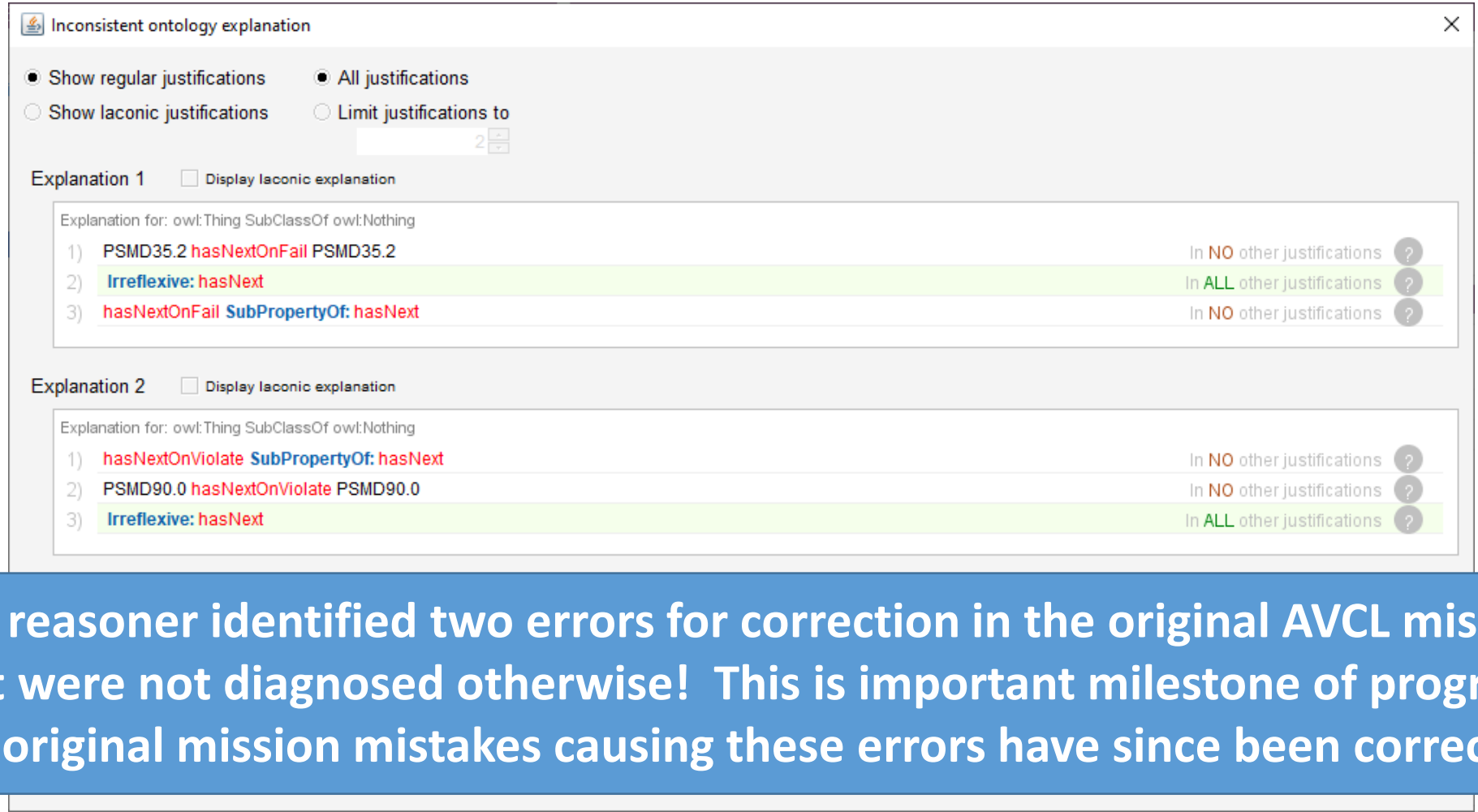
Not receiving any warnings is good!

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

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

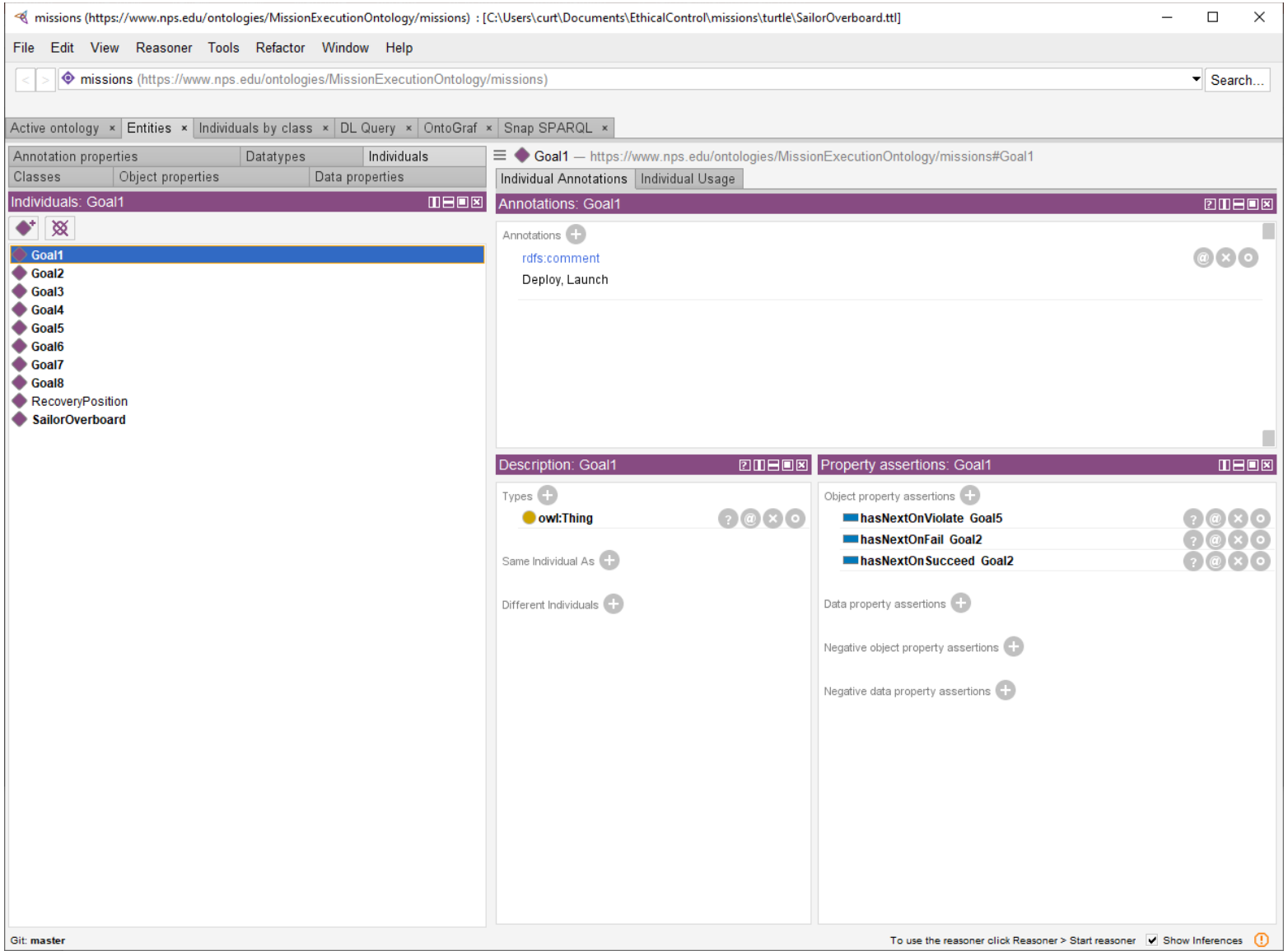
TODO: figure out why offline ARQ validation isn't identifying every problem as expected.

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)

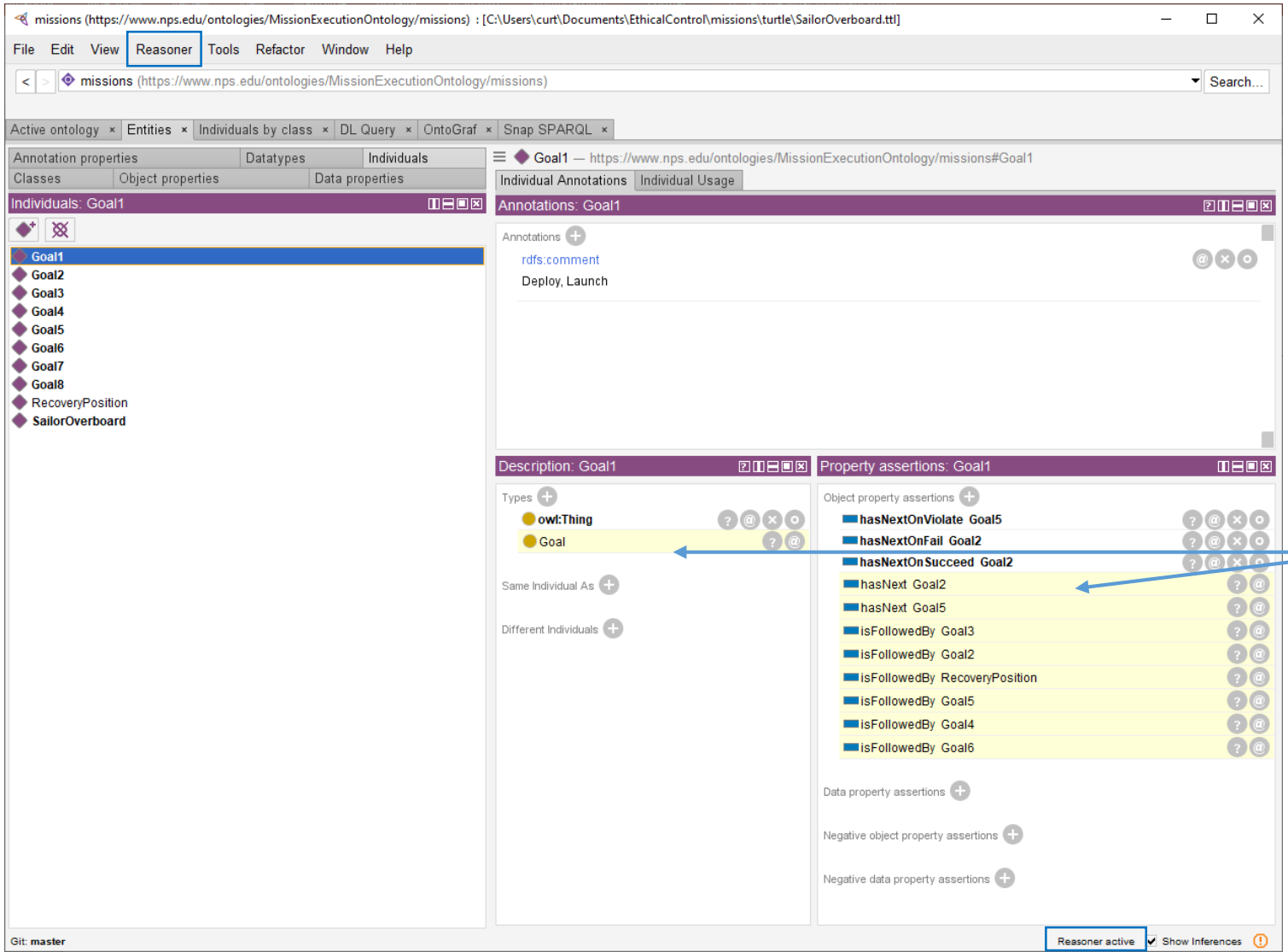


The reasoner identified two errors for correction in the original AVCL mission that were not diagnosed otherwise! This is important milestone of progress. The original mission mistakes causing these errors have since been corrected.

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

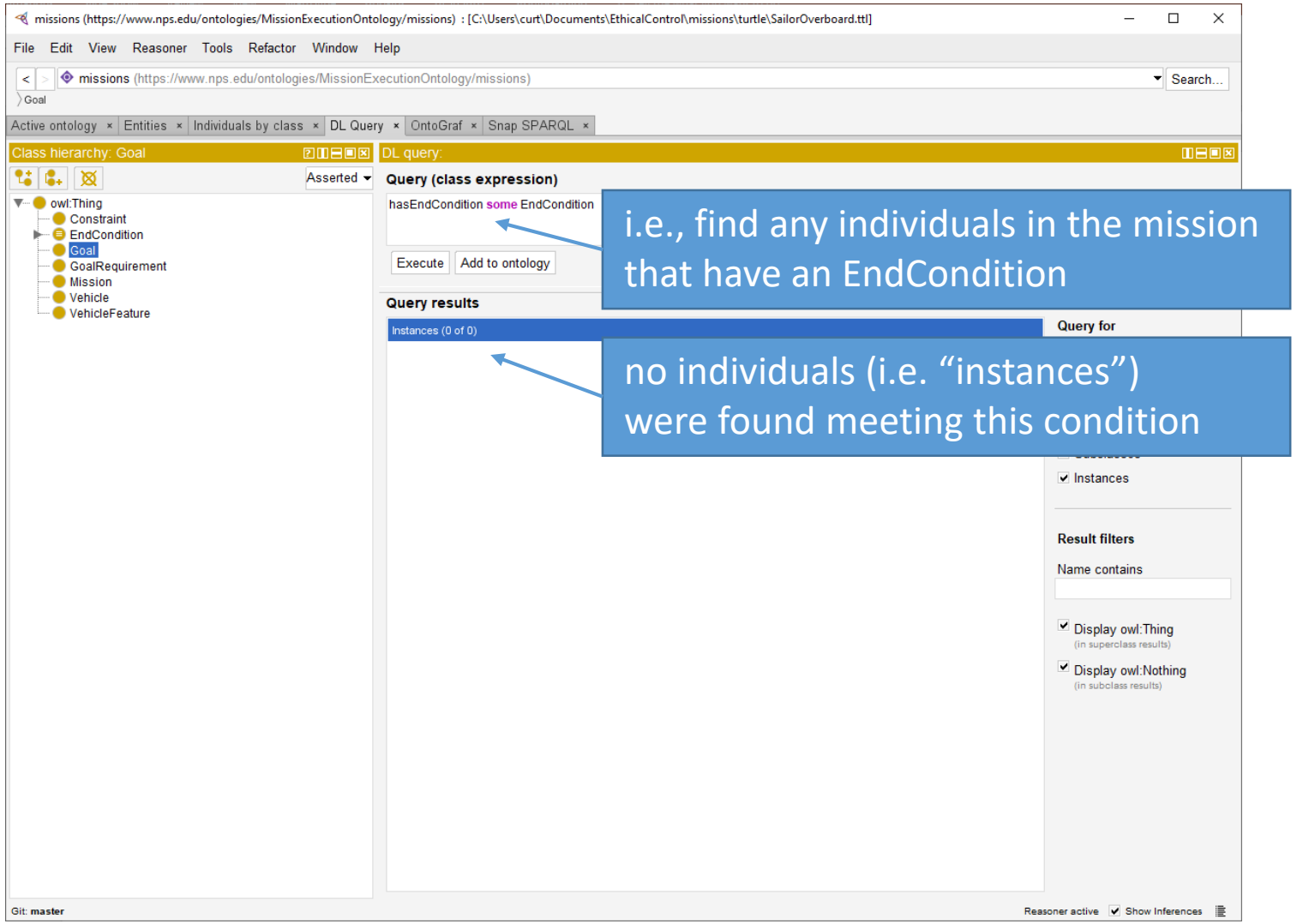


… now run Reasoner to see what can be inferred from the formal *Mission* definition



Inferred axioms are highlighted

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

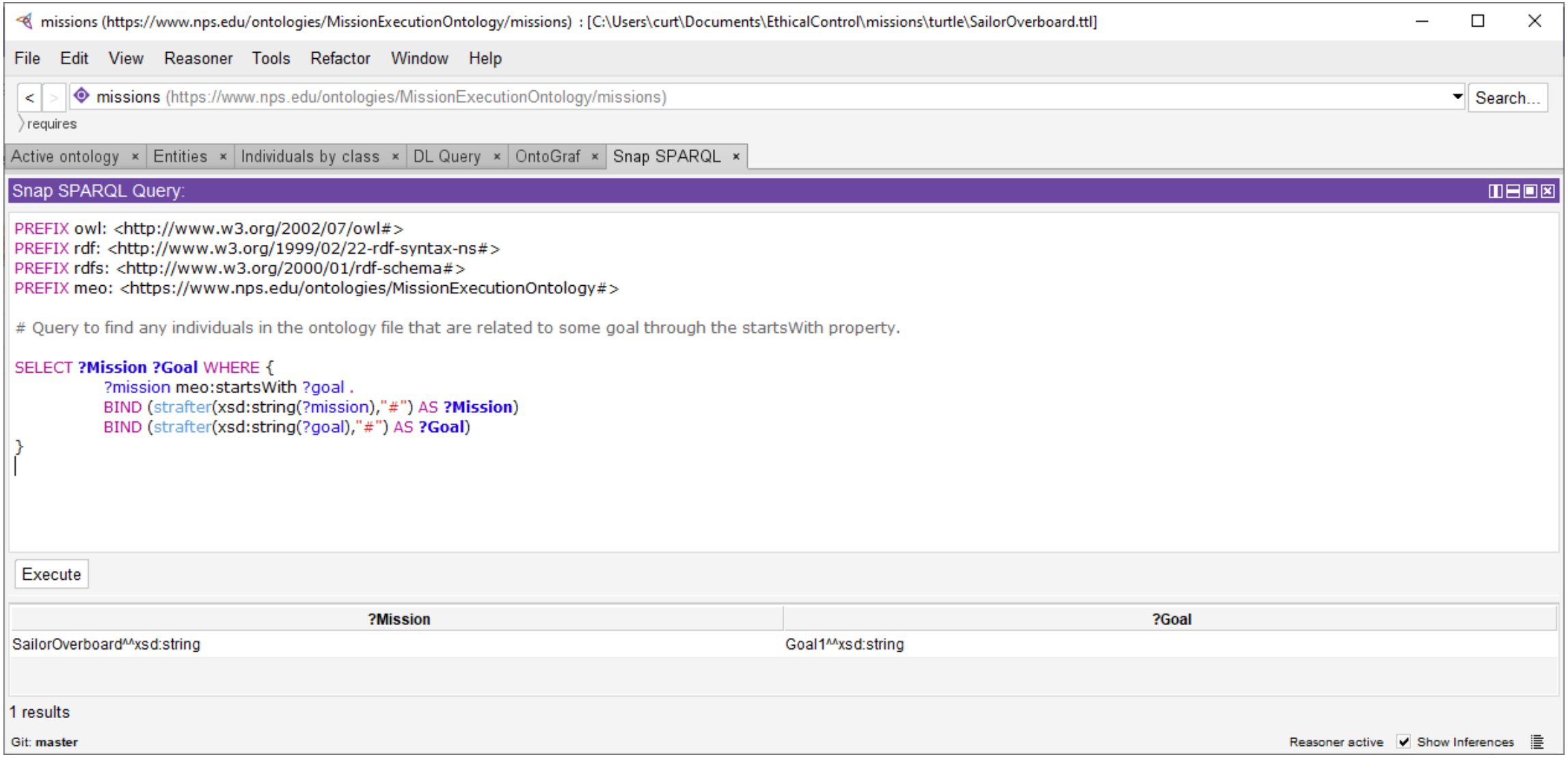


Here, none of the Goal individuals in the mission ontology satisfy the property constraint “hasEndCondition some EndCondition”. The mission definition does not satisfy some of the initial design ideas encoded into the base ontology rules.

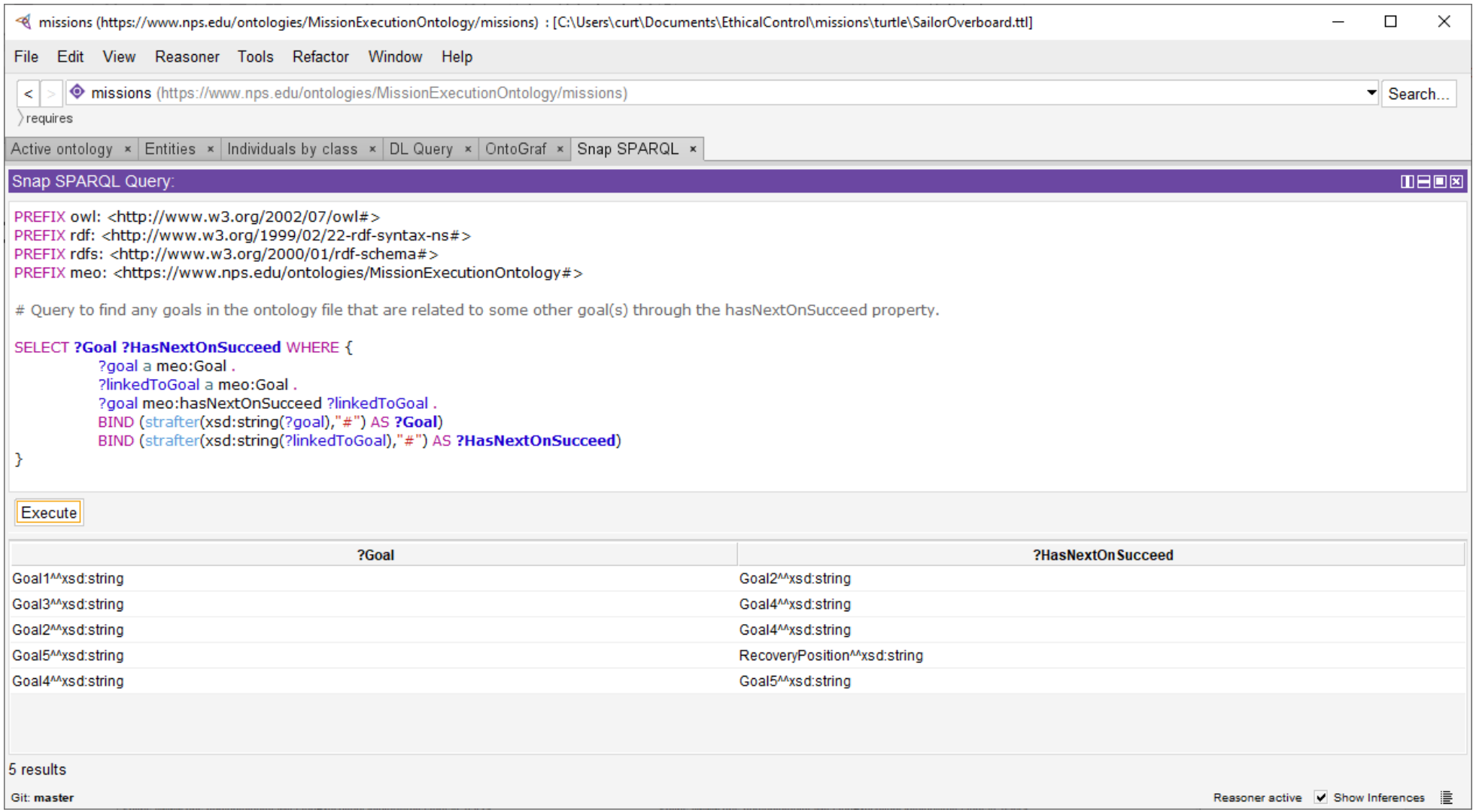
TODO: review, fix!

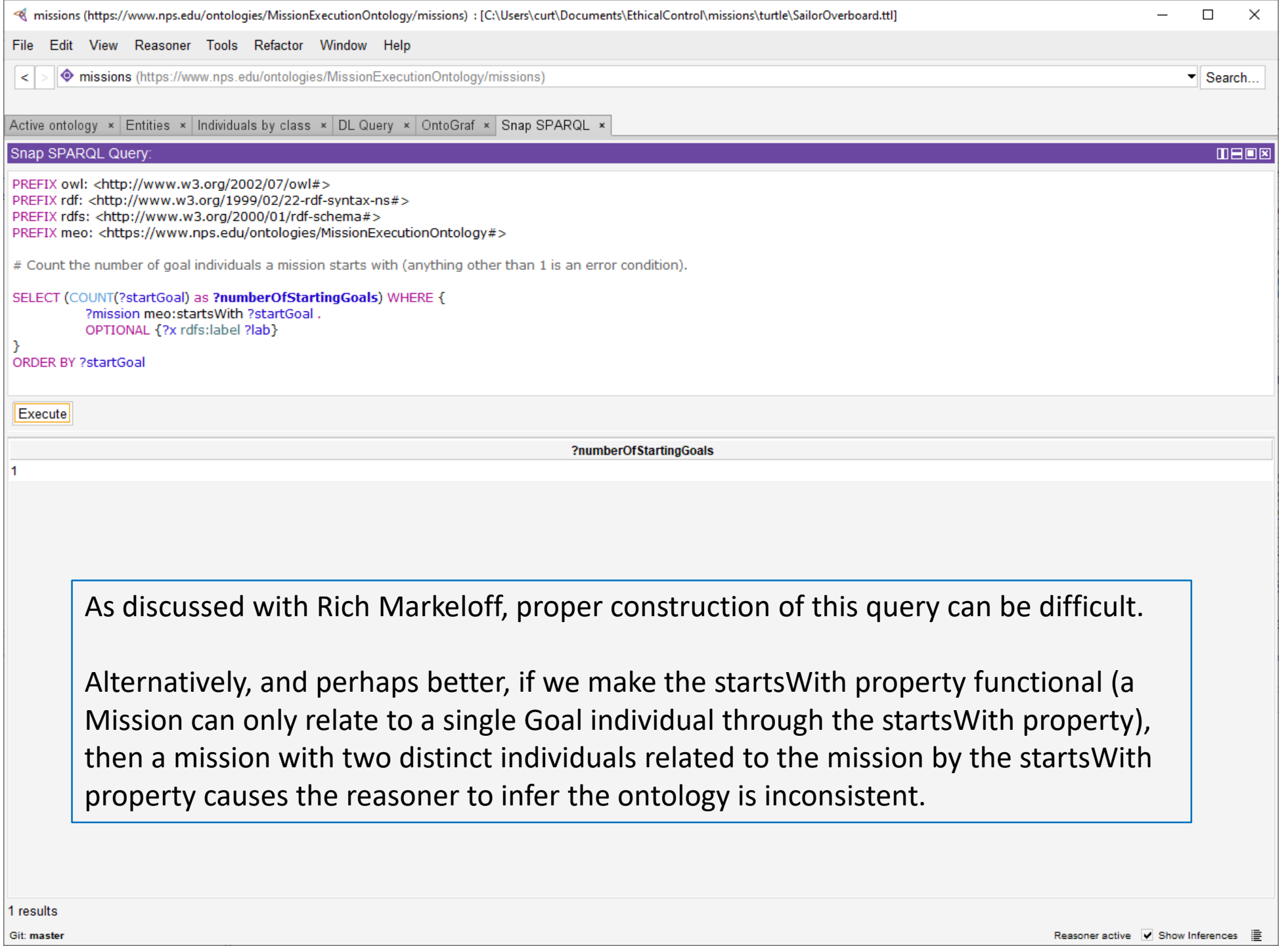
Can also use SPARQL for such queries, which can be executed inside or outside a tool like Protégé.

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



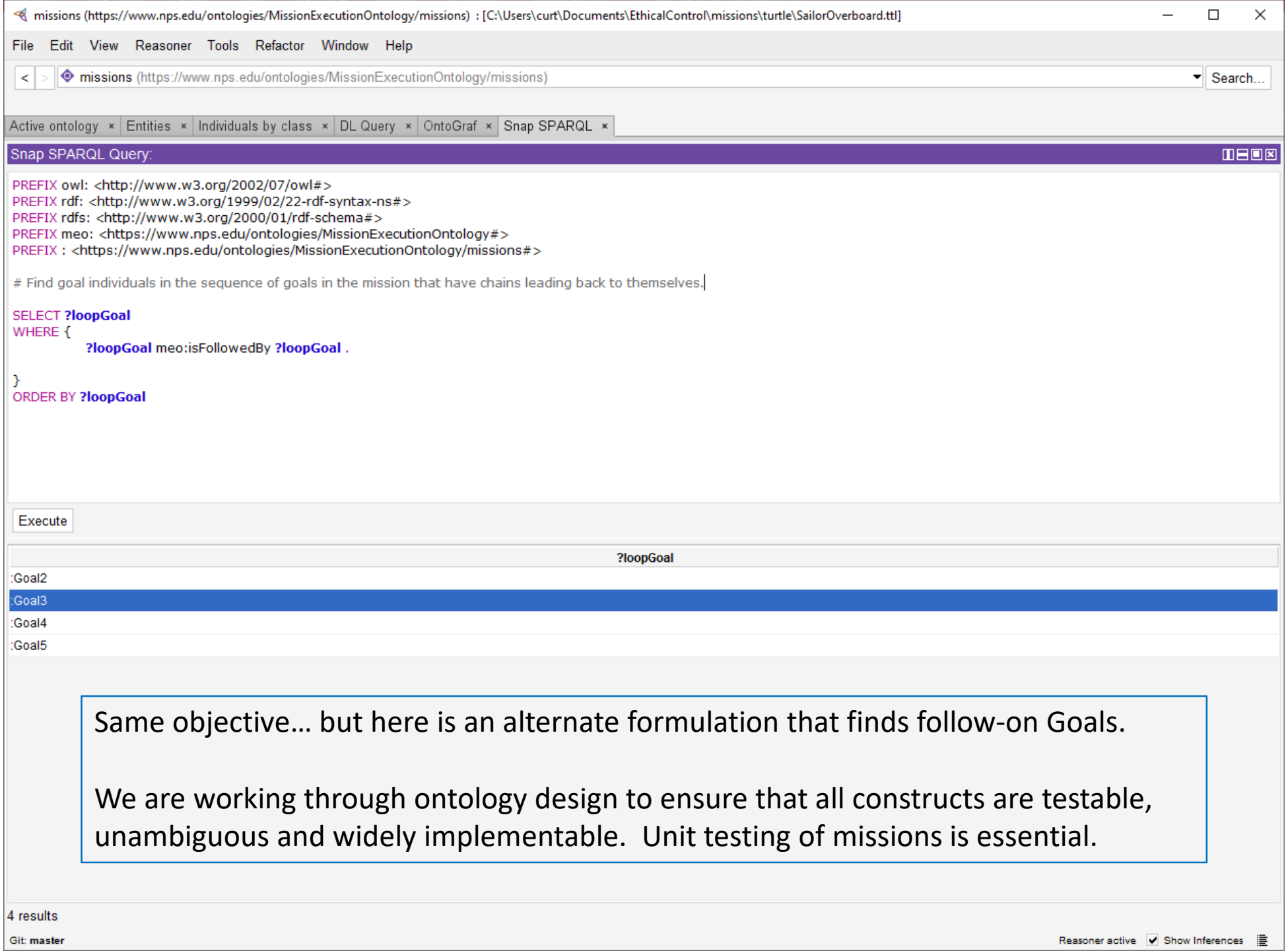
SPARQL query: Find *Goal* individuals linked to other *Goal* individuals through *hasNextOnSucceed* property





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.



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