Astronomical Object Types Ontology

INAF / CDS

Alexandre RICHARD (INAF)
Sébastien DERRIERE (CDS)
Andrea PREITE MARTINEZ (INAF)

Overview

- Ontology updates
- Mapping use-cases
- From concepts to instances
- Designing an automated instantiation process

Ontology updates

- Third major revision including feedback from the software prototypes *(ObjectTypes_1.00.owl)*
- All non-necessary conditions fully re-written backwards

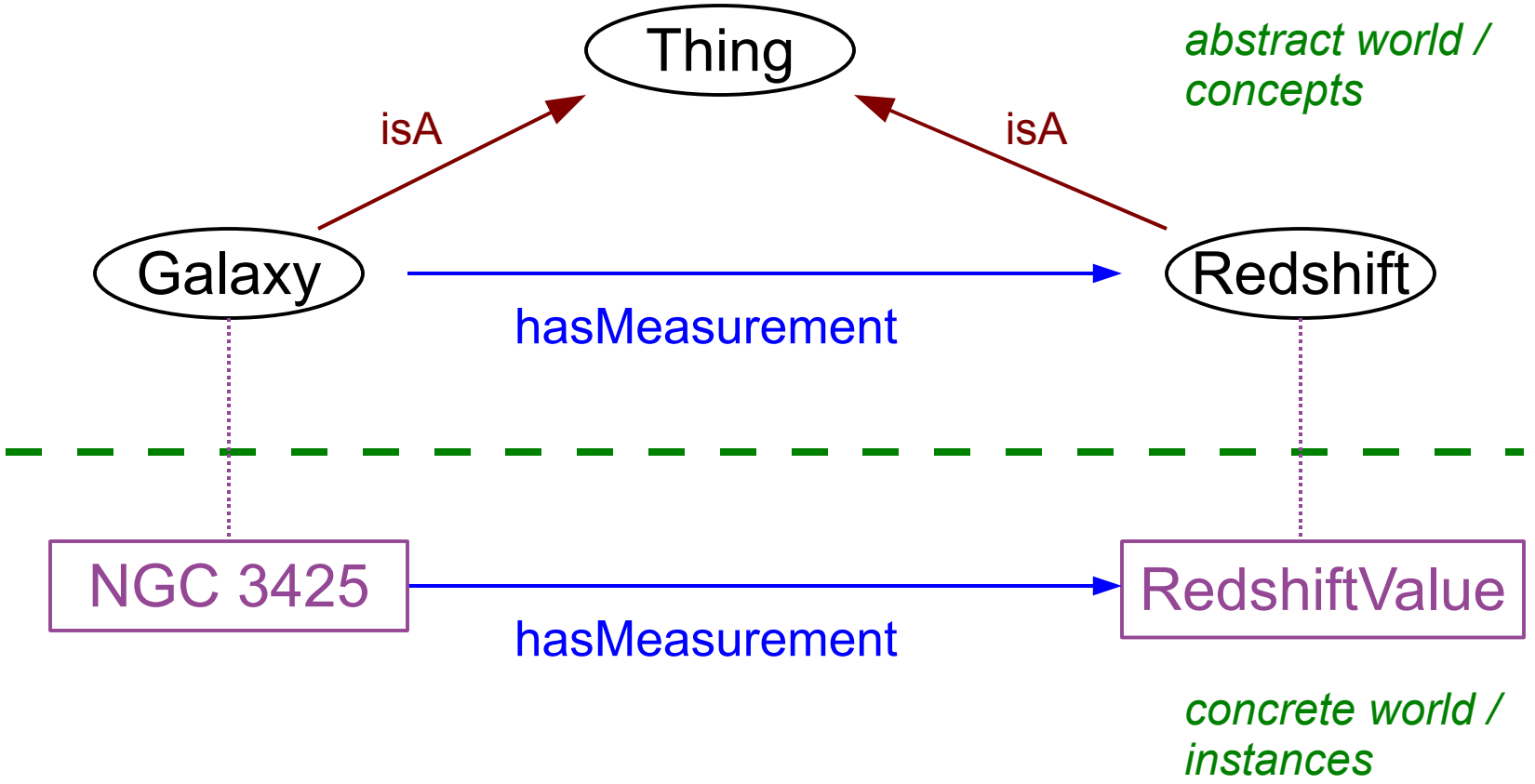
  e.g. VariableObject hasMeasurement some VariabilityPeriod
  VariabilityPeriod isMeasuredFor only VariableObject
- Datatype added to annotations
- OWL2 compliance
- All changes documented in an updated Technical Note published in the IVOA

Mapping use-cases

- Mapping of keywords use-case extended
  - Any keyword present in the ontology usable
  - Mapping strategy can exploit any ontology relationship
- Can be used to enrich existing sets of keywords
  - Annotate concepts with keywords
  - Derive broader/narrower/related terms from the subsumption and properties

- Call for use-cases

Concepts and instances



From concepts to instances (1/2)

- Checking the consistency of NED or SIMBAD objects using concepts only:
  - Proved to work...
  - ...but a full concept classification of the ontology is a heavy process (~25 database entries checked/min)
- Using instances instead:
  - Would be a much more natural approach
  - Would address the performance issue for such use-cases
  - Requires an automated process for creating instances from an astronomical object data source
    ...which itself would allow the creation of persistent sets of instances corresponding to astronomical objects

From concepts to instances (2/2)

- Converting data to ontology elements was partly explored with the consistency checker use-case
- Data was converted into ontology elements to help define a concept to be classified
  - Object types converted to subsumers
    - e.g. CV* (otype) => CataclysmicVariable (concept)
  - Other data (e.g. measurements, morphology, components) converted to restrictions.
    - e.g. hasMeasurement some Redshift
      hasMorphology some GalaxyMorphologyBarredSpiral

      hasComponent 2
      hasComponent some WhiteDwarf

Designing an automated instantiation process (1/2)

- Non-necessary conditions expressed backwards are not an issue because they use universal quantification:
  - VariableObject hasMeasurement some VariabilityPeriod
    VariabilityPeriod isMeasuredFor only VariableObject
- Leave out data that do not translate into symbolic properties
  - General goal is providing a layer of semantic tools on top of a data source, not remaking the data source
- If any object type associated with an astronomical object then take it into account for the instance
  - Keep as much information as possible
  - A possible inconsistency provides information

Designing an automated instantiation process (2/2)

- Ontology manipulation class being ported to OWLAPI
  - Better performance and support
  - Direct interfacing with reasoners

- Instantiation process and experience from use-cases presented during the 2nd Practical Semantic Astronomy Workshop (2-5 March, 2009 - Glasgow)