Proof of Concept: Controlled Natural Language Logic For Knowledge Engineering At United Utilities

Anthony Beck

Abstract

As businesses grow in complexity the ability to express operational concepts in a simple and explicit manner is becoming increasingly important to effectively communicate within and between organisations. This covers all aspects of a business from describing conceptual operational structures, underlying data frameworks and the intent of operational processes.

To facilitate this *Controlled Natural Languages* (CNL such as *RuleSpeak*) have been developed to both capture and communicate business rules. The point is to build an understanding of how the business systems are expected to function (the 'as-design' model) and compare this to how they actually function (the 'as-built' model) mediated by the problems the business systems are expected to solve (the business process model). Much of this information is undocumented and needs extracting from domain experts. Domain experts are not knowledge engineers or logicians. CNL helps extract this information in a manner which the domain experts can understand and provide feedback on. Hence, the abstracted model is a closer reflection to reality.

Whilst the RuleSpeak CNL reduces or removes ambiguity it is only understandable by a human reader. The Knowledge Engineering community have developed other CNLs which can be used to create *Formal Ontologies*.

Ontologies are a way to capture and share people's knowledge about the world in a way that is processable by computer systems. Ontologies have the potential to serve as a bridge between the human conceptual understanding of the world and the data produced, processed and stored in computer systems. CNL can be used to develop both a common understanding of business processes and ontologies which in turn can be used to enhance and improve business systems and to provide new *value-add* services.

During the SPA project the *cleanse and match team* required such information to determine:

- the actual nature of the data (a data profiling exercise)
- the expected nature of the data (an 'as-design' data modelling exercise)
- the business problems which the data is expected to help solve (a business modelling exercise)
- the business operational processes (an 'as-built' data modelling exercise)

This document describes a proof of concept implementation of the ACE/APE CNL which can be used to derive an OWL formal ontology which in turn can be used for automated reasoning. The data is based upon a real case study from United Utilities based on the Open Water policy initiative.

It is argued that the following benefits could ensueP

- Rapidly understand systems
- Rapidly visualize relationships and frameworks
- Be able to undertake quality improvement from Top down (*conceptual*) and bottom up (*how the data has been physically modelled*)
- Transfer knowledge and re-use within new environments
- Communicate knowledge more effectively within teams and to third parties
- Store knowledge so it can be re-used in downstream projects and products
- Sell services based upon the knowledge model

This document has been written in CommonMark: an unambiguous implementation of Markdown for scholarly writing.

Contents

Current Distribution list:

- Tim Stanley (Tim.Stanley@uuplc.co.uk)
- Paul Williams (Paul.Williams@uuplc.co.uk)

Original Distribution list (2014-08-27):

- John Daniels (John.Daniels@uuplc.co.uk)
- Stu Mitchell (Stuart.Mitchell@uuplc.co.uk)
- Lee Mooney (Lee.Mooney@uuplc.co.uk)

1 Expressing knowledge: Rules and Ontologies

As businesses grow in complexity the ability to express operational concepts in a simple and explicit manner is becoming increasingly important to effectively communicate within and between organisations. This covers all aspects of a business from describing conceptual operational structures, underlying data frameworks and the intent of operational processes.

To facilitate this *Controlled Natural Languages* (CNL such as *RuleSpeak*) have been developed to both capture and communicate business rules. The point is to build an understanding of how the business systems are expected to function (the 'as-design' model) and compare this to how they actually function (the 'as-built' model) mediated by the problems the business systems are expected to solve (the business process model). Much of this information is undocumented and needs extracting from domain experts. Domain experts are not knowledge engineers or logicians. CNL helps extract this information in a manner which the domain experts can understand and provide feedback on. Hence, the abstracted model is a closer reflection to reality.

Whilst the RuleSpeak CNL reduces or removes ambiguity it is only understandable by a human reader. There are other CNLs which have been developed in the Knowledge Engineering community which focus on developing Formal Ontologies. This extends the utility of the rules by ensuring they are human *and* computer readable.

Ontologies have been proposed and studied in the last couple of decades as a way to capture and share people's knowledge about the world in a way that is *computer readable*. This allows *inference engines* to test the *validity* of *facts/statements* and to *infer new knowledge*. Ontologies have the potential to serve as a bridge between the human conceptual understanding of the world and the abstracted view of the world stored in computer systems.

One of the main reasons for the lack of widespread adoption of ontologies is the steep learning curve for authoring them: most people find it too difficult to learn the syntax and formal semantics of ontology languages. CNL can be used to develop both a common understanding of business processes and ontologies which in turn can be used to enhance and improve business systems and to provide new *value-add* services.

During the SPA project the *cleanse and match team* required such information to determine:

- the actual nature of the data (a data profiling exercise)
- the expected nature of the data (an 'as-design' conceptual modelling exercise)
- the business problems which the data is expected to help solve (a business requirements exercise)
- the business operational processes (an 'as-built' business modelling exercise)

This document outlines a proof of concept approach for generating such models.

2 Controlled Natural Language systems for Ontology engineering

2.1 Ontology creation

Traditional ontology development requires at least two types of people:

- A knowledge engineer - who can structure the underlying logic that supports the knowledge model
- A domain expert - who has in depth understanding and experience of the phenomena to be modelled

The mathematical nature of description logic has meant that domain experts find them hard to understand. This forms a significant impediment to the creation and adoption of ontologies. Ontology focussed Controlled Natural Language systems constrain language syntax and allow statements to be translated into the Web Ontology Language (OWL) with the aim of achieving both comprehension by domain experts and computational precision.

2.2 The benefit of CNLs for ontology creation

The fundamental principles underlying the use of CNLs for Ontology engineering are (after Denaux (2013)):

- To allow the domain expert, with the aid of a knowledge engineer and tool support, to express their knowledge as easily and simply as possible and in as much detail as necessary.
- To have a well defined grammar and be sufficiently formal to enable those aspects that can be expressed as OWL to be systematically translatable.
- To be comprehensible by domain experts with little or no knowledge of OWL.
- To be independent of any specific domain.

These principles can be achieved - at least in part. One such example is the use of the Rabbit CNL developed by Ordnance Survey (Hart et al. (2008)):

> The original intention was for Rabbit to enable domain experts alone to author ontologies. However, practice showed that whilst domain experts could build ontologies, these ontologies often contained many modelling errors not related to the language but the modelling processes. None-the-less Rabbit still enables the domain expert to take the lead and to author ontologies with guidance in modelling techniques from a knowledge engineer. Rabbit also enables other domain experts to verify the ontology.

2.3 Candidate systems

Two candidate systems were considered:

- ROO and Rabbit (Denaux 2013, Hart et al. (2008))
- APE and ACE (Anon. 2014)

ROO and Rabbit are not as actively developed as APE and ACE. For this reason the proof of concept was undertaken using APE and ACE. Both languages have limitations and a more rigorous review of CNLs and CNL ontology modelling would need to be undertaken if these approaches are implemented by 1Spatial.

In the future other systems should be reviewed including Fluent Editor which includes CNL for SWRL rule authoring.

2.4 Ontology CNLs compared to formal language

CNLs should contain a number of language element. Those used to

- express axioms
- introduce (or declare) concepts, relationships and individuals
- define relationships between concepts
- define properties of concepts and individuals

The table below contrasts the ACE representation with a “formal language” representation that is normally used in such ontologies. This is intended to show that natural language, if carefully controlled, provides a more readable and writable ontology representation, without a loss in generality and expressive power.

Table 1: example statements represented in ACE and formal logic notation

ACE notation	Formal logic notation
Every man is a human.	man $\subseteq$ human
Every human is a male or is a female.	human $\subseteq$ male $\cup$ female
John is a student and Mary is a student.	$\{$John, Mary$\}$ $\subseteq$ student
No dog is a cat.	dog $\subseteq \neg$ cat
Every driver owns a car.	driver $\subseteq \exists$ own car
Everything that a goat eats is some grass.	goat $\subseteq \forall$ eat grass
John likes Mary.	$<$John, Mary$>$ $\in$ like
Everybody who loves somebody likes him/her.	love(X, Y) $\implies$ like(X, Y)

2.5 Computer parsable CNLs compared to RuleSpeak

Essentially, from a human persepctive, a computer parsable CNL is no different from a human parsable CNL such as *RuleSpeak*. The difference is that a *computer parsable CNL* can be used to build an Ontology.

2.6 Why ontologies (reasoning and generalisation)

An ontology is a logic framework which provides powerful functionality. The rich model that describes the relationships between and properties of concepts and individuals can be used to generalise data and infer new knowledge. The geographical potential of such approaches are described in detail in Hart & Dolbear (2013).

For example Figure 1 describes the relationship between the different levels of the Corine land-cover classification hierarchy expressed within an OWL ontology. This encoding can be used to collapse the groupings and generalise the data based upon properties and relations within the ontology. Hence, individual instances can be generalised up the hierarchy.

Inspire is also transitioning into an ontology led framework employing Linked Open Data (LOD).

The reasoning and inference capabilities are powerful. Assertions can be tested for logical validity. Statements can be tested for their validity based upon rules:

- Rule
  - No man is a woman.
- Statements
  - John is a woman.
  - John is a man.

the reasoner will find an inconsistency because John has been asserted as a man and men have been described as *disjoint* from women (*disjoint* means that no *thing* can be both a man **and** a woman (or members of the set of *man* do not exist in members of the set *woman*). This is shown in Figure 2.

Reasoning about transitivity and *HasValue* will find the following text inconsistent (see Figure 3).

- If something X follows something that follows something Y then X follows Y.
- John follows Mary who follows Bill who follows John.
- No manager follows Mary.
- Bill is a manager.

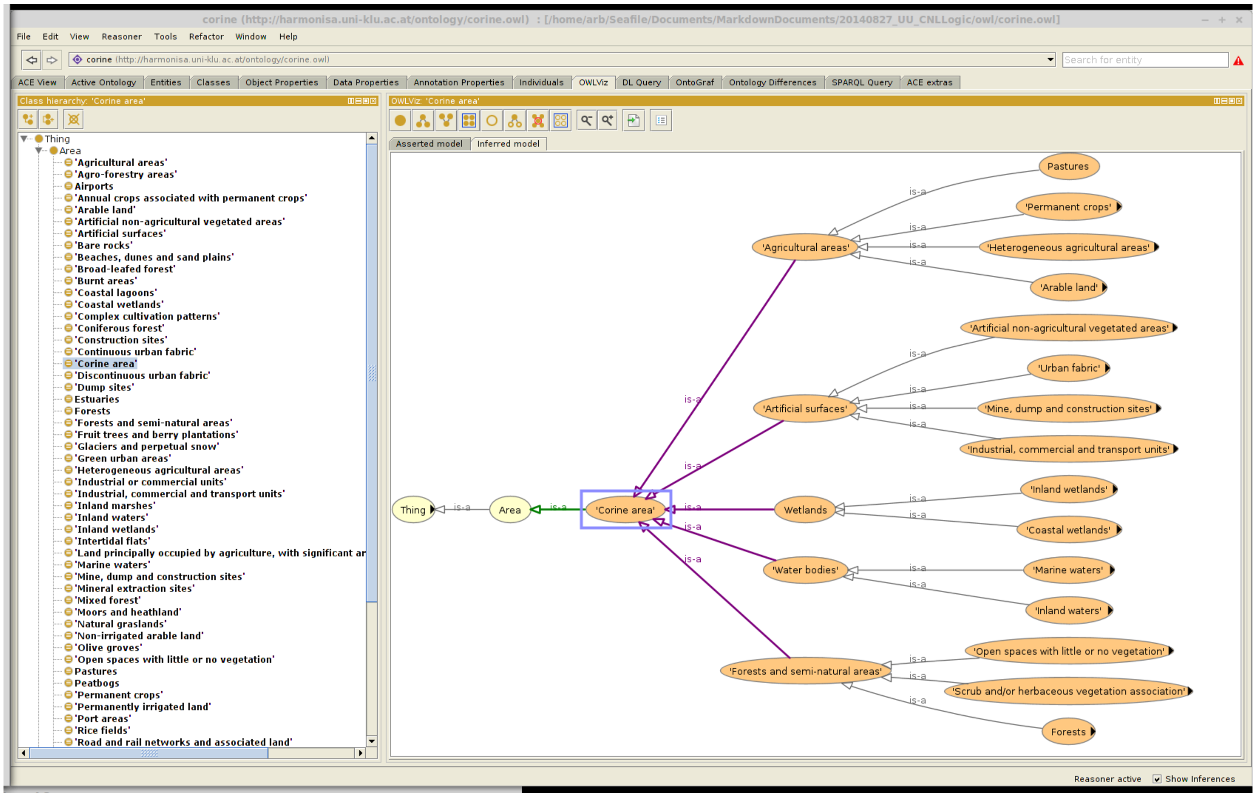

Figure 1: The relationship between the different levels of the Corine land-cover classification hierarchy expressed within an OWL ontology

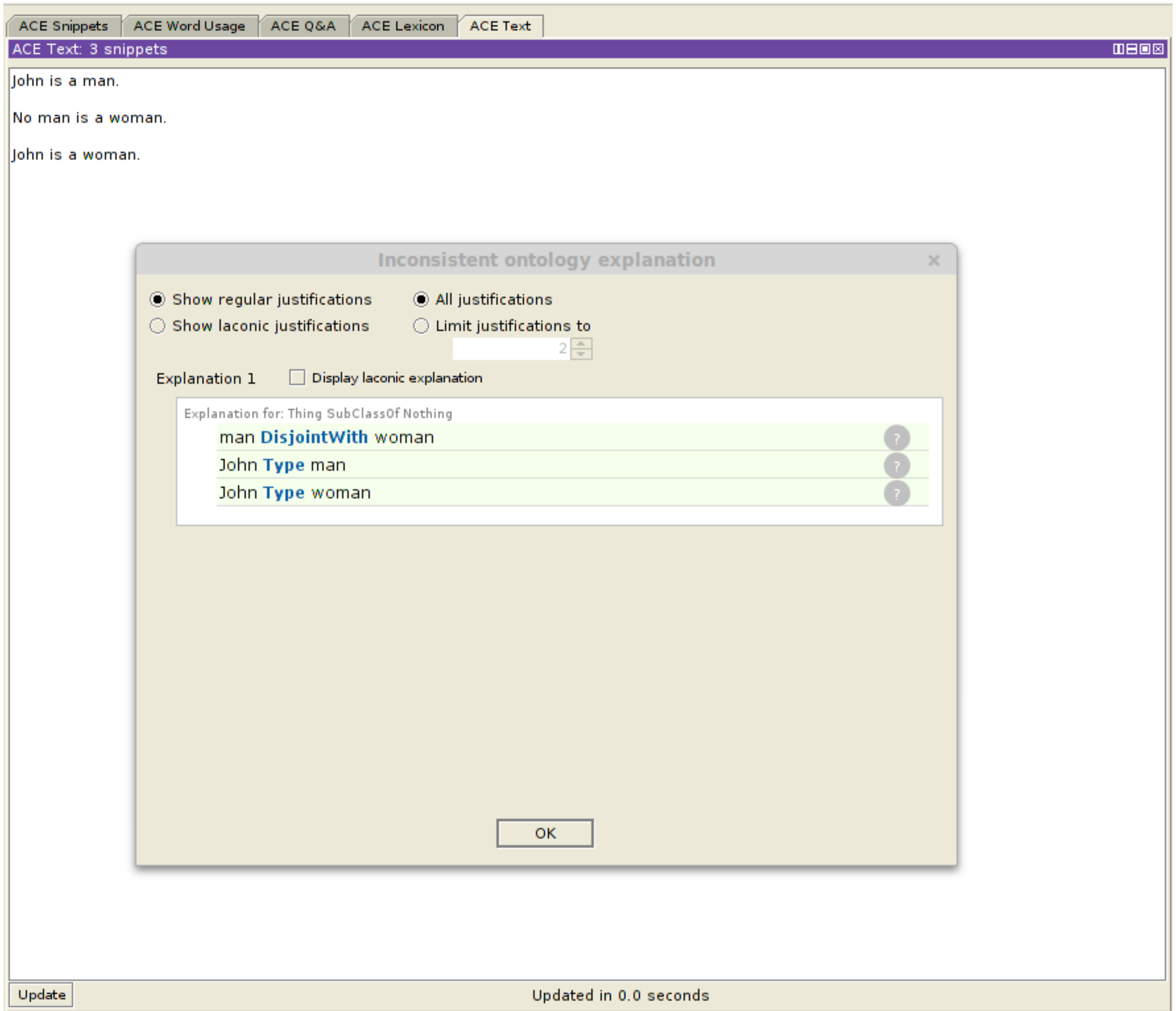

Figure 2: The protege error message for an invalid assertion on disjoint

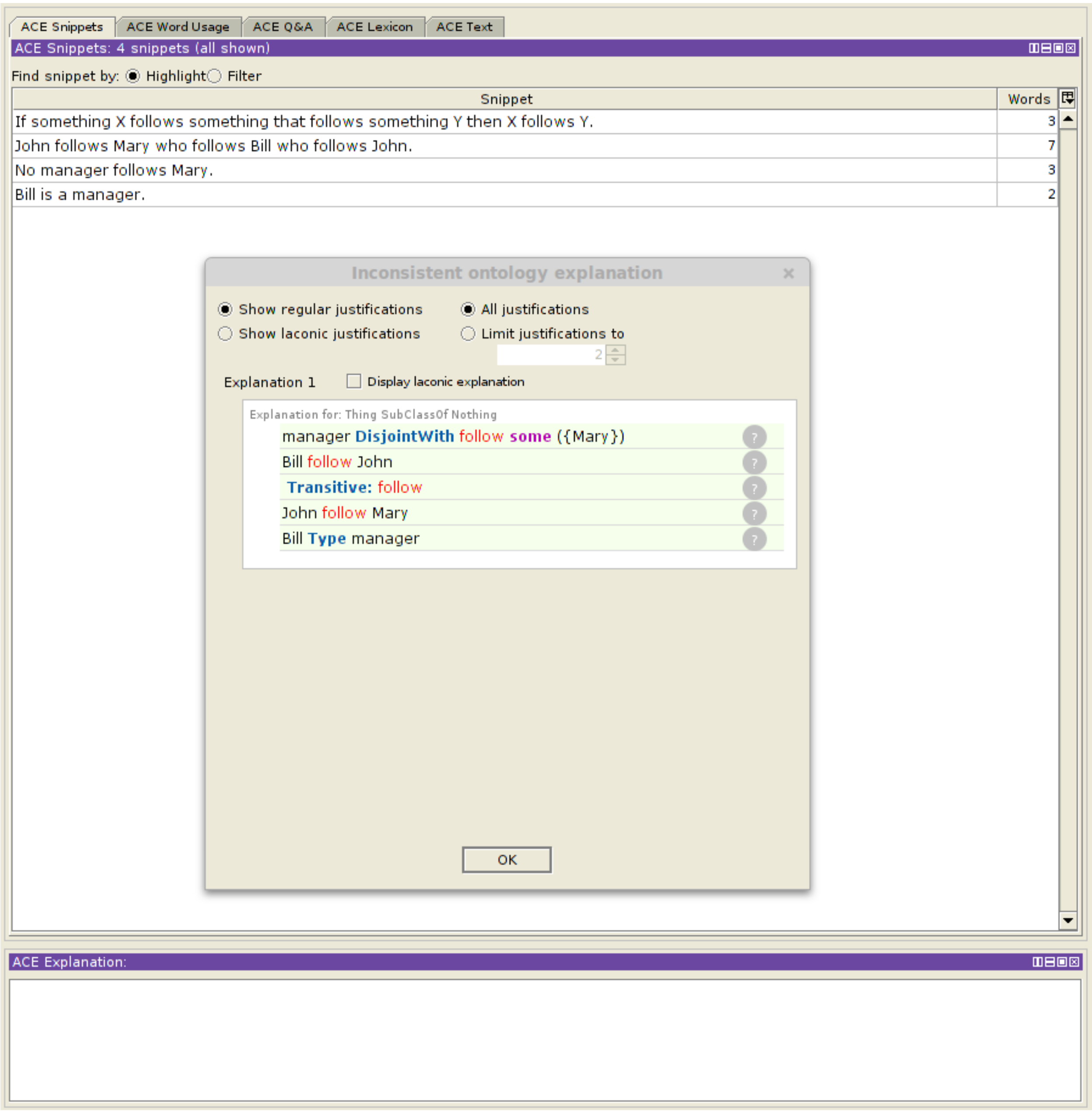

Figure 3: The protege error message for an invalid assertion on transitivity

3 Proof of concept problem - The impact of Open Water

This document was written on 2014-08-27 and reflects the knowledge acquired prior to this point. It has not been updated since this time to reflect developments in SPA or FCS.

3.1 United Utilities

United Utilities (UU) provides both *water* and *wastewater services*. The location of the *termination point* of a United Utilities *wholesale service asset* to a customer is called a *service point*. These service points may be for water supply, sewerage services, metering, trade effluent and any other service within the wholesale service catalogue. A service point is a sub-set of the UU wholesale assets which is uniquely billable and has a geographical location. This locational geography needs to adequately represent billing and operational activities.

The Address Management System (AMS) fulfills parts of this need. AMS is a bespoke data repository holding *Service Point Addresses* for water, wastewater and electricity (now deprecated) domains. AMS is not an address gazetteer. Figure 4 provides an overview of AMS.

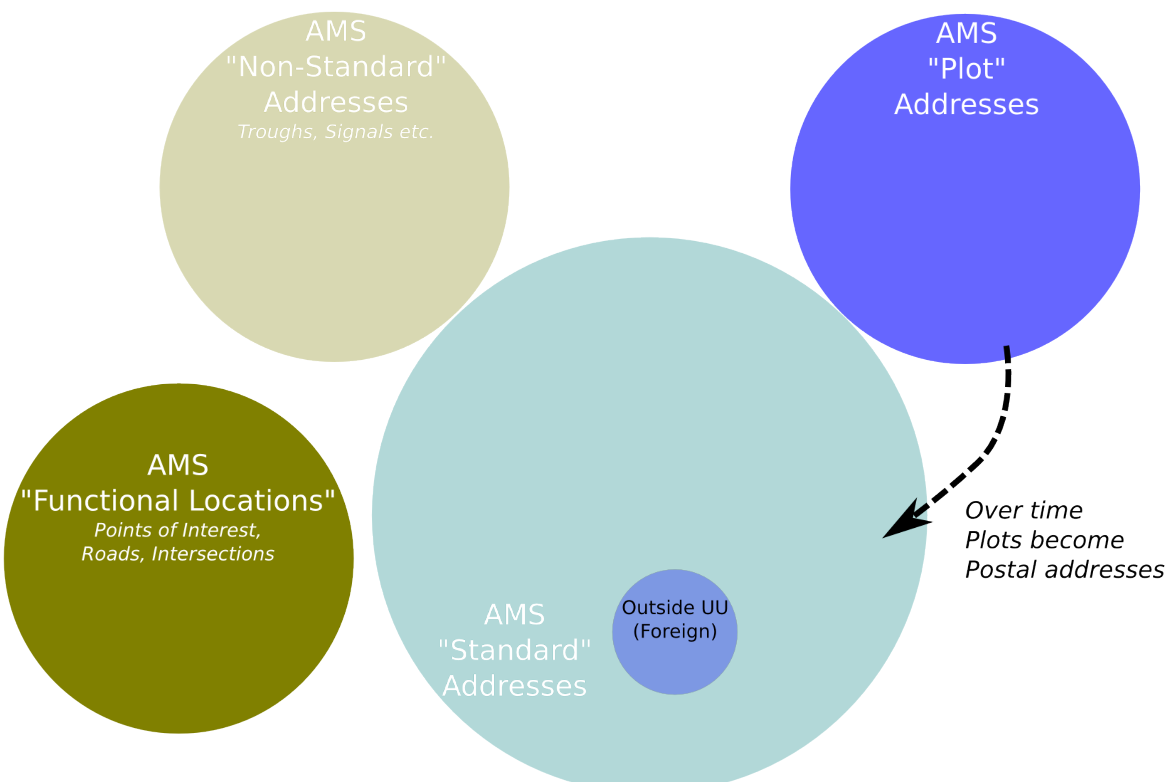

Figure 4: The UU Address Management System

AMS records are related to a *service point*:

- In most cases these service points are at an addressable property (or a *standard address*) such as a house or office. These have an *address_type* of *postal* (PO).

  - There are also some legacy records that refer to postal addresses outside the UU operational area. These have an *address_type* of *foreign* (FO).

- Some AMS records are related to potential future service points which will be constructed. These represent *plot addresses*. These have an *address_type* of *plot* (PL).
- The AMS record might refer to a service point feeding a cattle trough or a railway signalling box. These represent *non-standard addresses*. These have an *address_type* of *locational* (LO).

3.2 Open Water

At present UU only provides *wholesale* services. When the Open Water initiative is enacted UU will be split into a *wholesale* and *retail* business. The wholesale part of UU will sell services to a small number of elegible third parties (including the retail part of UU) who will act as retail agents. These elegible retail agents will sell services to individual consumers. The wholesale business will provide transparent systems to all eligible retail businesses. Key to this will be the interoperable exchange of *service point* information. This implies that after Open Water has been enacted there will be both *Retail Service Points* and *Wholesale Service Points*. UU has recognised the important and potentially disruptive impact of Open Water. This recognition underpins the Service Point Address (SPA) project:

> In the coming competitive environment, the importance of the service point address will increase. On 2nd January 2014, the OpenWater Group published documentation on how it sees the competitive market operating. A fundamental requirement for market operation is "for registration and switching, a centrally held record of premises, service points, and associated meters and market participants". The ability of suppliers to interact and automatically match to this data will be critical in running an efficient business.
>
> The current addressing system suffers from poor integration with existing systems including Alto and IWM, poor data quality and is based on an end of life, custom application.
>
> Service point address (SPA) will provide a single version of the truth for property addresses and service point locations within the strategic UUGIS (LAM) application and integrate to other key enterprise applications (IWM, Alto). Our critical success will be ensuring that when dealing with a property which receives a UU service, all of our teams are presented with consistent, accurate information about the property address and location.

The Open Water policy initiative will be very disruptive to the Water domain. Other utility companies in the water domain will be responding to this policy initiative.

3.3 The CNL developed for Open Water based on UU views of its assets

On 17th July there was a meeting at UU as part of the SPA project which discussed the forthcoming *Open Water* initiative in significant detail. It became apparent that the impact of Open Water was unclear. This was not helped by semantic and conceptual issues. For example the terminology had multiple interpretations that could be construed differently depending upon the mental models employed. Based upon this experience it was determined that a better way of sharing this information was required[1].

This meeting resulted in Figure 5: a conceptual model that identifies the nature of Service Points and connectivity of the physical assets in relation to Open Water. These relationships have been encoded as CNL as detailed below

[1]this is an important point: the benefits described here are not primarily about the moelling and inferencing capabilities of an ontology but in providing a team with clear and unambiguous representations

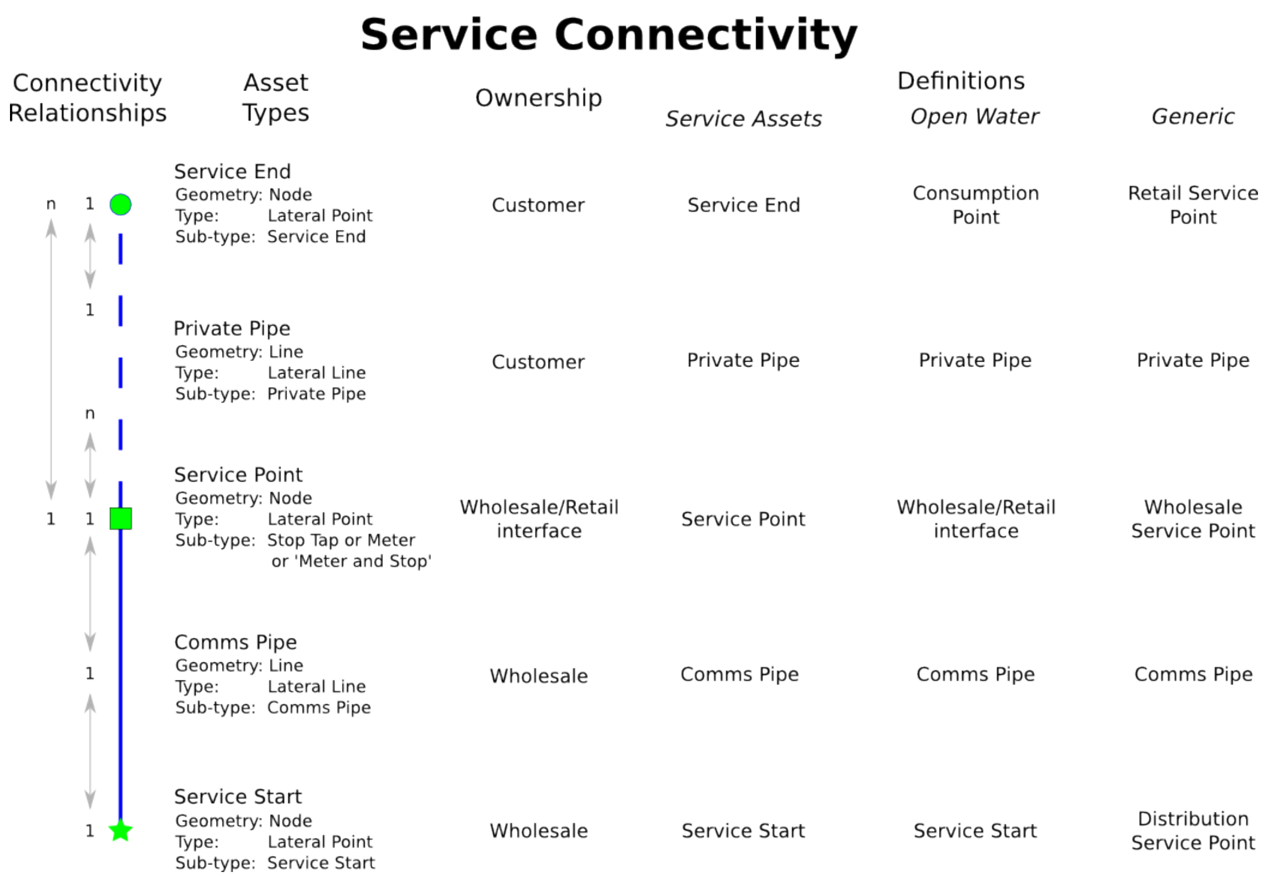

Figure 5: Service Point connectivity at United Utilities

3.3.1 Classes and properties

A series of simple statements that define classes and properties. The validity of these statements can be easily tested by domain experts in the business. It is important to ensure that edge cases are detected[2]

Every lateralPoint is an asset .

Every lateralPoint hasGeometry node .

Every lateralLine is an asset .

Every lateralLine hasGeometry line .

Every main is a lateralLine .

Every main is a wholesaleAsset .

Every commsPipe is a lateralLine .

Every commsPipe is a wholesaleAsset .

Every privatePipe is a lateralLine .

Every privatePipe is a retailAsset .

Every ferrule is a lateralPoint .

Every meter is a lateralPoint .

Every stopTap is a lateralPoint .

Every meterAndStop is a lateralPoint .

Every serviceStart is a lateralPoint .

Every serviceStart is a ferrule .

Every serviceStart is a wholesaleAsset .

Every distributionServicePoint is a serviceStart .

Every servicePoint is a lateralPoint .

Every servicePoint is a wholesaleAsset and is a retailAsset.

Every wholesaleServicePoint is a servicePoint .

Every serviceEnd is a lateralPoint .

Every serviceEnd is a retailAsset .

3.3.2 General constraints

Constraints have been given to the classes (and by extension their instances) which are defined above.

Every serviceStart is physicallyConnected to 1 main and 1 commsPipe .

Every servicePoint is a stopTap or is a meter or is a meterAndStop .

Every servicePoint is physicallyConnected to 1 main and at least 1 commsPipe .

Every serviceEnd is physicallyConnected to 1 commsPipe .

[2] for example - whilst the statement holds for most things it doesn't work when X or Y

3.3.3 Transitivity rules

Some transitivity rules are detailed below. That explain that if A is *physicallyConnected* to B and B *physicallyConnected* to C then A is part of the same network as C. There is also an inverse statement so that connections can be identified irrespective of direction.

If something X follows something that follows something Y then X follows Y.

If something X physicallyConnected something that physicallyConnected something Y then X networkConnected Y.

If something X physicallyConnected something Y then something Y physicallyConnected X.

If something X physicallyConnected something Y then something X networkConnected Y.

If something X networkConnected something Y then something Y networkConnected X.

If something X is taller than something Y and Y is taller than something Z then X is taller than Z.

If something X is physicallyConnected something Y and Y is physicallyConnected something Z then X is networkConnected Z.

3.3.4 Individuals

Some individuals have been created. These can be used to test the validity of statements about concepts and to test connectivity. A *physicallyConnected* network has been generated.

ID_12345 is a ferrule .

ID_12345 is a serviceStart .

ID_12345 physicallyConnected ID_45678 .

ID_45678 physicallyConnected ID_99999 .

ID_99999 physicallyConnected ID_99998 .

Every retailServicePoint is a serviceEnd .

3.3.5 Questions for the reasoner to answer

Some questions have been stated for the reasoner to answer:

What is a ferrule ?

What is ID_12345 ?

What is an asset ?

What is a wholesaleAsset ?

Is ID_99998 networkConnected ID_12345 ?

3.3.6 Entailment (what does this imply)

To support this section a 4 minute video has been created that shows how the ACE/APE CNL statements can be loaded into the Protege ontology editor. There are two versions of this video (if there are issues in accessing these videos then they can be supplied in alternative ways):

- Original webm
- Converted wmv (for windows users who do not have the appropriate codecs)

Relationships between the statements can be quickly visualised in Protege as shown in Figure 6. This facility alone is an enhancement to that offered by RuleSpeak.

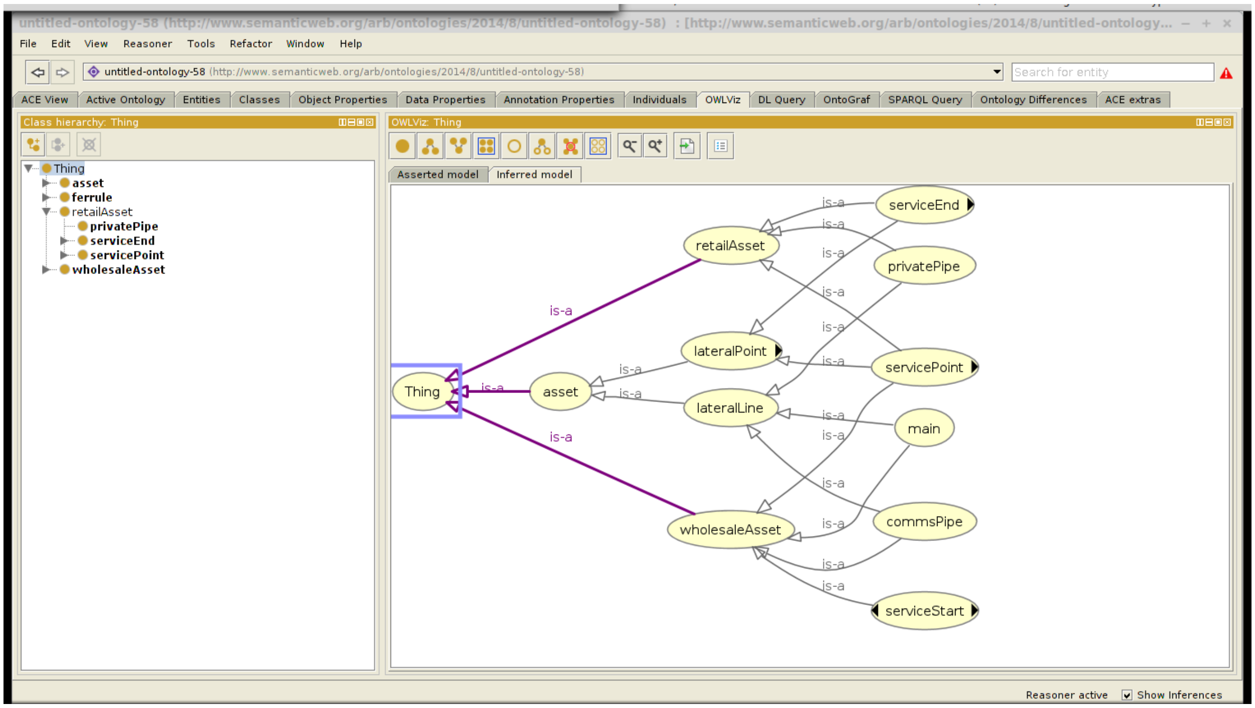

Figure 6: A visualization of the CNL concepts in protege

As explained in the videos a number of different visualizations, views and facts can be seen using the built in tools in Protege. In addition we have used ACE/APE to ask a number of questions of the ontology as detailed in the figures below:

3.4 The CNL developed for Open Water based on the Systems Architecture and Data Model document

Open Water have produced a number of consultation documents. This architectural overview has been taken from Open-Water (2014a) and Open-Water (2014b).

This was *partially* modelled to create the following CNL and is therefore not fully articulated.

3.4.1 Classes

3.4.1.1 Nature of the operators Every wholesaleOperator is a marketParticipant .

Every retailOperator is a marketParticipant .

Every marketOperator is a marketParticipant .

marketOperator is at most one thing .

Every wholesaleOperator is a waSC or is a wOC or is a nAV .

Every retailOperator is a waSC or is a wOC or is a nAV .

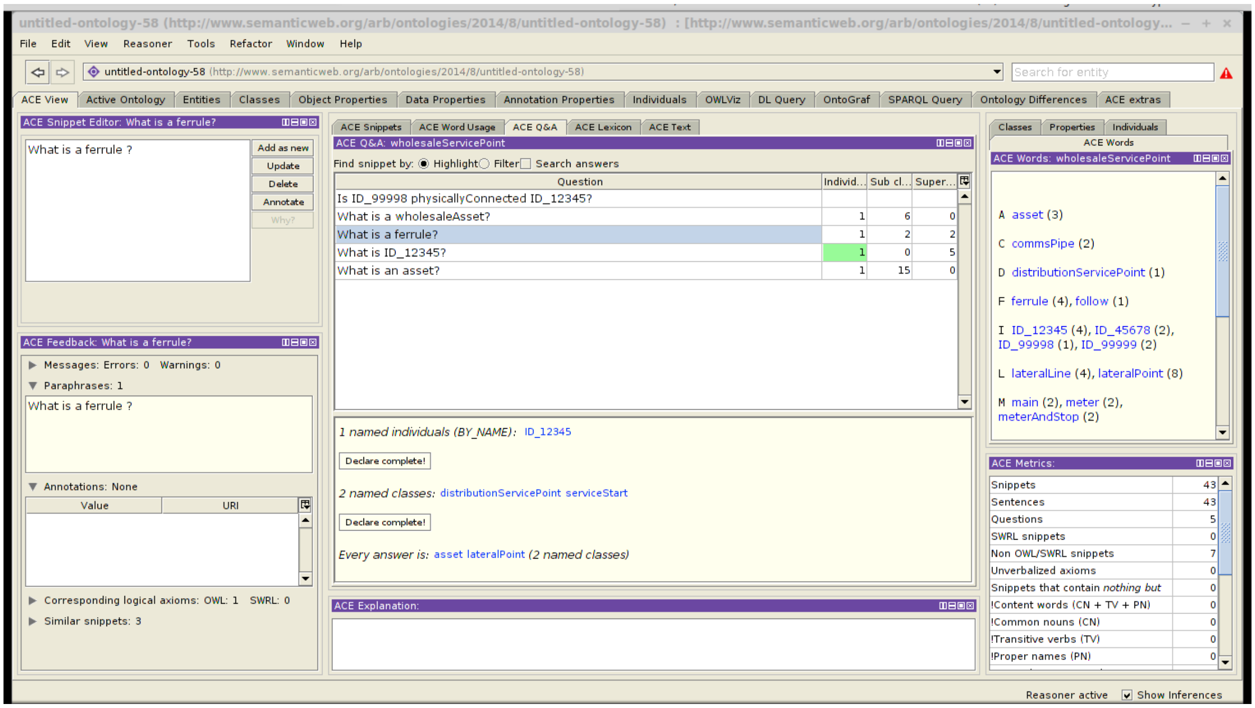

Figure 7: What is a Ferrule?

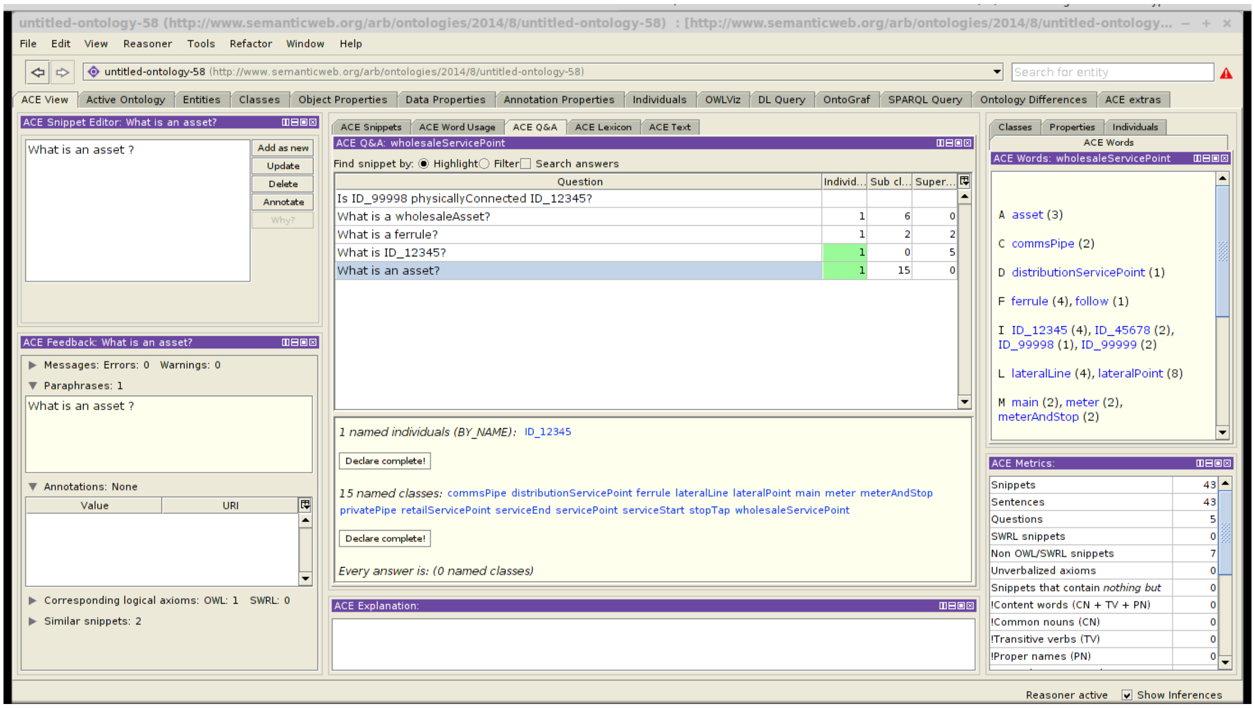

Figure 8: What is an Asset?

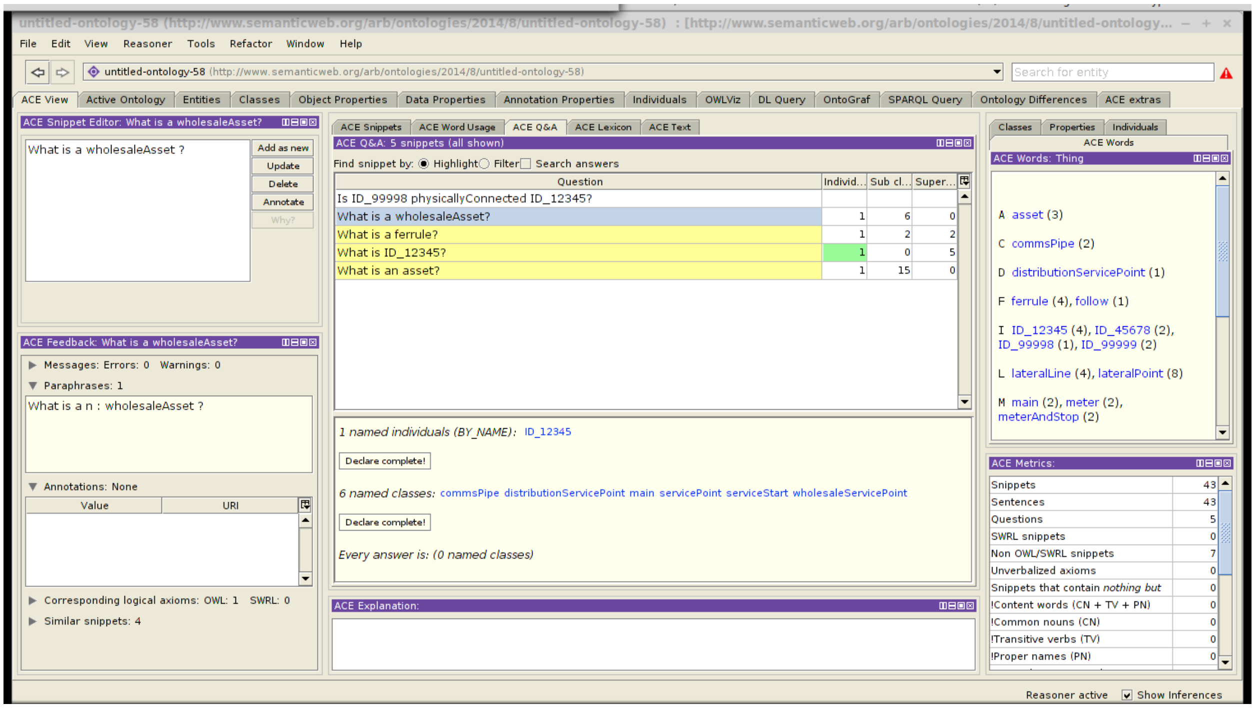

Figure 9: What is a Wholesale Asset?

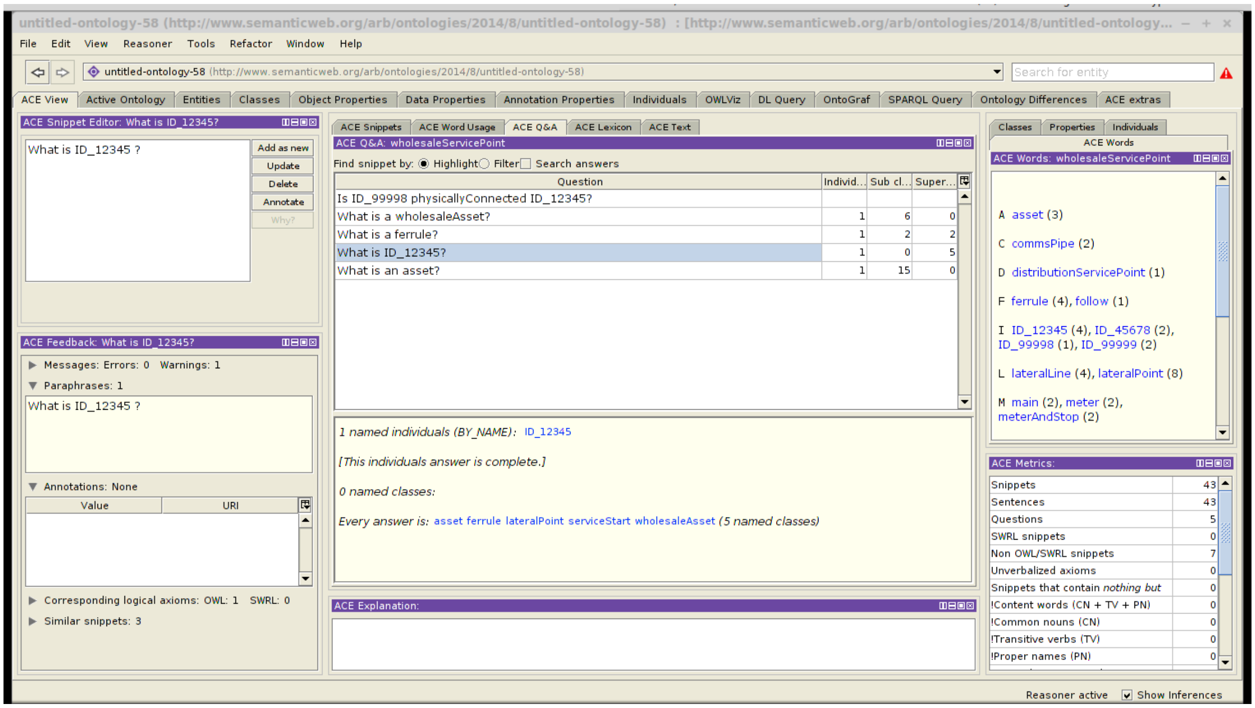

Figure 10: What is ID_12345?

3.4.1.2 IT requirements

Every registrationDatabase is a database .

Every registrationDatabase holdsInformation sites .

Every registrationDatabase holdsInformation parties .

Every registrationDatabase provides registrationServices .

Every registrationDatabase provides switchingServices .

Every switchingService is a service .

Every registrationService is a service .

Every financialDatabase is a database .

Every financialDatabase holdsInformation meterReadings .

Every financialDatabase provides financialServices .

Every financialDatabase provides settlementServices .

Every settlementService is a financialService .

Every settlementService requires a calculationSystem and a meterReading and a wholesaleOperator and a retailOperator .

Every financialService is a service .

3.4.2 Properties

Every servicePoint hasCustomerUPRN a UPRN.

Every servicePoint hasCustomerUPRN at most one thing .

Every servicePoint hasRetailer a retailOperator

Every servicePoint hasRetailer exactly one thing .

Every servicePoint hasWholesaler a wholesaleOperator.

Every servicePoint hasWholesaler exactly one thing .

Every meter hasServicePoint a servicePoint.

Every meter hasServicePoint at least one thing .

Every meter hasMeterType a meterType.

Every meter hasMeterType exactly one thing .

Every meter hasSerialNumber a serialNumber .

Every meter hasSerialNumber exactly one thing .

Every meter hasInstallDate a installDate .

Every meter hasInstallDate exactly one thing .

Every meter hasUseStatus a useStatus .

Every meter hasUseStatus exactly one thing .

Every meter hasMeterReading a meterReading .

Every meter hasMeterReading at least one thing .

3.4.3 Elements not yet modelled

- a conceptual model for registration data;
- a conceptual model for service request data;
- a conceptual data model for consumption and financial settlement data;
- and indicative event and data volumes.

3.4.4 Individuals

UU is a waSC .

UU is a wholesaleOperator .

UU is a wholesaleOperator .

UU is not a marketOperator .

3.4.5 Entailment

Figure 11 is a visualization of the Systems Architecture and Data Model concepts described in CNL and turned into an OWL ontology.

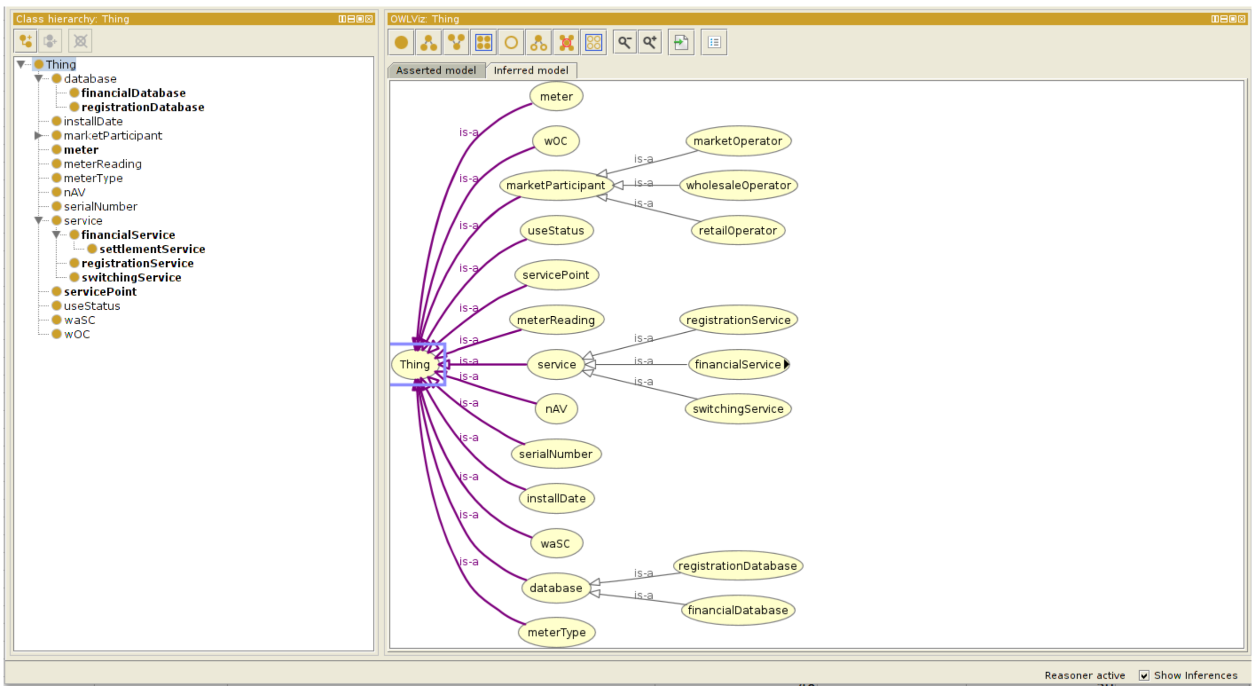

Figure 11: A visualization of the Systems Architecture and Data Model concepts in CNL

4 Summary

- Rapidly understand systems
- Rapidly visualize relationships and frameworks
- Be able to undertake quality improvement from Top down (*conceptual*) and bottom up (*how the data has been physically modelled*)

- Transfer knowledge and re-use within new environments
- Communicate knowledge more effectively within teams and to third parties
- Store knowledge so it can be re-used in downstream projects and products
- Sell services based upon the knowledge model

Bibliography

Anon., 2014. Attempto controlled english. *Wikipedia, the free encyclopedia*. Available at: http://en.wikipedia.org/w/index.php?title=Attempto_Controlled_English&oldid=620928248 [Accessed September 17, 2014].

Denaux, R., 2013. *Intuitive ontology authoring using controlled natural language*. PhD thesis. University of Leeds. Available at: http://etheses.whiterose.ac.uk/4446/ [Accessed September 17, 2014].

Hart, G. & Dolbear, C., 2013. *Linked data: A geographic perspective*, CRC Press.

Hart, G., Johnson, M. & Dolbear, C., 2008. Rabbit: Developing a control natural language for authoring ontologies. In S. Bechhofer et al., eds. *The semantic web: Research and applications*. Lecture notes in computer science. Springer Berlin Heidelberg, pp. 348–360.

Open-Water, 2014a. *Systems architecture and data model*, Open Water. Available at: http://www.open-water.org.uk/media/1047/systems-architecture-and-data-model.pdf.

Open-Water, 2014b. *Systems architecture and data model: Consultation responses and analysis*, Open Water. Available at: http://www.open-water.org.uk/media/1084/system-architecture-and-data-model-redacted-responses-and-analy pdf.