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

Anthony Beck

October 9, 2014

Abstract

Good quality data with a well understood scope leads to better decisions.
1Spatial consultants need to rapidly determine a number of things:

- 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)[1]

The ability to express rules and logic in syntactically simple phrases is becoming increasingly important to how we develop an understanding and effectively communicating our clients business needs (expecially in helping to distinguish between their pre-conceived and actual need). To facilitate this process 1Spatial uses *RuleSpeak* to both capture and communicate business rules. RuleSpeak is an example of a Controlled Natural Language (CNL). 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[2]. 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 ontologies then we can use to enhance our approaches and improve our services.

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.

This proof-of-concept has not been produced on 1Spatial company time.

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

[1]Not completely happy with this
[2]I don't know of a RuleSpeak Natural Language Processing (NLP) parser

Contents

Distribution list:

- 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

The ability to express rules and logic in syntactically simple phrases is becoming increasingly important to how we develop an understanding of clients business needs and then effectively communicating this understanding to our clients. This is particularly important when identifying the difference between a client's pre-conceived need and their actual need (as reflected in the data and the surrounding business processes). To facilitate this 1Spatial use *RuleSpeak* to both capture and communicate business rules. RuleSpeak is an example of a Controlled Natural Language (CNL). Whilst Rulespeak reduces or removes ambiguity it is only understandable by a human reader[3]. 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. If CNL can be used to develop ontologies then we can produce a more formal and rigorous method of modelling data, conceptual and business processes and sharing these models in meaningful ways with our business partners.

During a project 1Spatial consultants need to rapidly determine a number of things:

- 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)[4]

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

2 Controlled Natural Language systems for Ontology engineering

2.1 Ontology creation

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 preciseness.

[3] I don't know of a RuleSpeak Natural Language Processing (NLP) parser
[4] Not completely happy with this

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.

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)

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

2.5 Ontology CNLs compared to RuleSpeak[5]

The conclusion of this section should be that Ontology CNLs are no harder to construct than RuleSpeak but provide greater value. However, it is likely that, at least in the short term, RuleSpeak provides more flexibility (i.e. not all concepts may be directly tractable/mappable to OWL ontologies).

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).

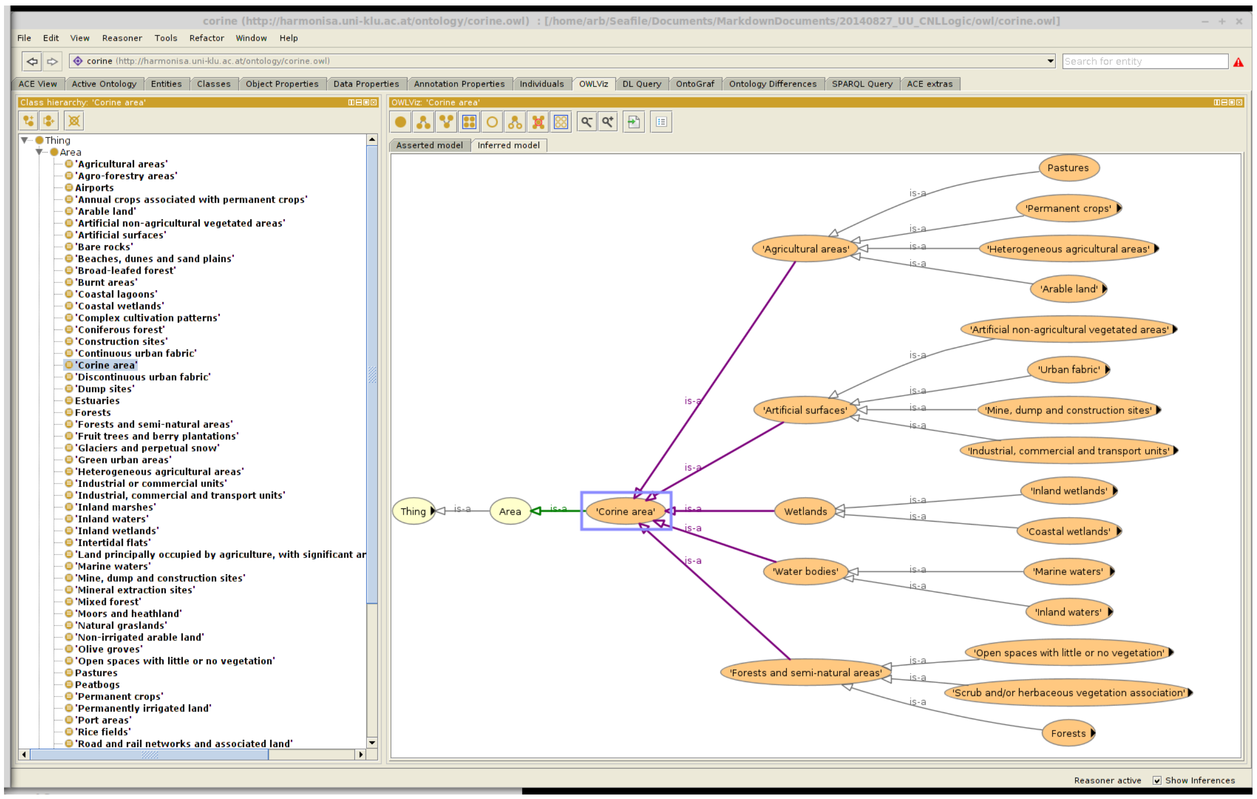
The reasoning and inference capabilities are powerful. Assertions can be tested for logical validity. Statements can be tested for their validity based upon rules:

- 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. This is shown in Figure 2. This complements the rules which can be expressed within 1Spatial's 1Integrate.

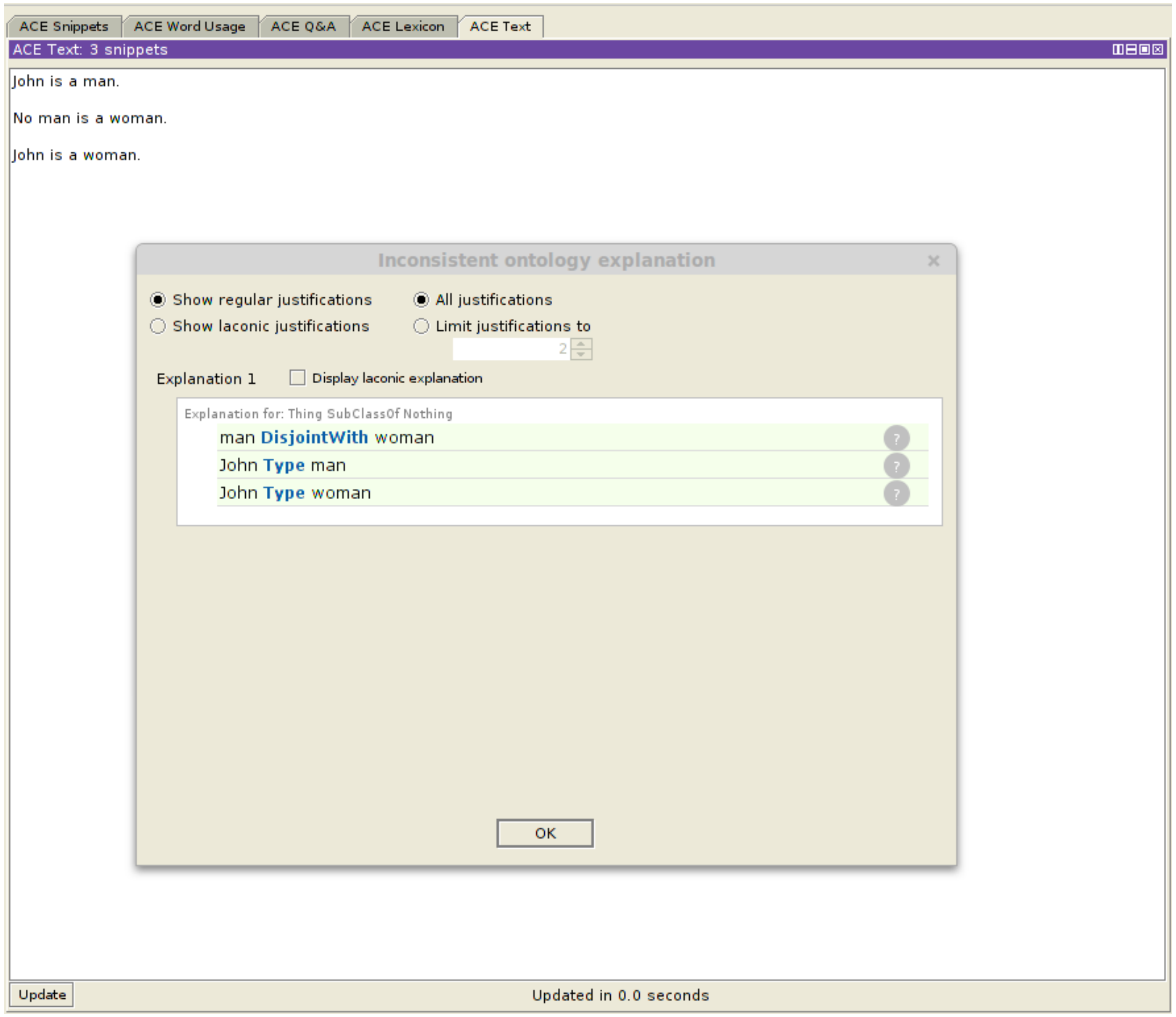
Reasoning about transitivity and *HasValue* will find the following text inconsistent (see Figure 3).

[5]Speak to Stephen Stanley about this



Thing
Area
'Agricultural areas'
'Agro-forestry areas'
Airports
'Annual crops associated with permanent crops'
'Arable land'
'Artificial non-agricultural vegetated areas'
'Artificial surfaces'
'Bare rocks'
'Beaches, dunes and sand plains'
'Broad-leafed forest'
'Burnt areas'
'Coastal lagoons'
'Coastal wetlands'
'Complex cultivation patterns'
'Coniferous forest'
'Construction sites'
'Continuous urban fabric'
'Corine area'
'Discontinuous urban fabric'
'Dump sites'
Estuaries
Forests
'Forests and semi-natural areas'
'Fruit trees and berry plantations'
'Glaciers and perpetual snow'
'Green urban areas'
'Heterogeneous agricultural areas'
'Industrial or commercial units'
'Industrial, commercial and transport units'
'Inland marshes'
'Inland waters'
'Inland wetlands'
'Intertidal flats'
'Land principally occupied by agriculture, with significant ar
'Marine waters'
'Mine, dump and construction sites'
'Mineral extraction sites'
'Mixed forest'
'Moors and heathland'
'Natural graslands'
'Non-irrigated arable land'
'Olive groves'
'Open spaces with little or no vegetation'
Pastures
Peatbogs
'Permanent crops'
'Permanently irrigated land'
'Port areas'
'Rice fields'
'Road and rail networks and associated land'

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



John is a man.

No man is a woman.

John is a woman.

[✓ Show regular justifications]
Show laconic justifications

[✓ All justifications]
Limit justifications to 2

Explanation 1

Display laconic explanation

Explanation for: Thing SubClassOf Nothing

man **DisjointWith** woman

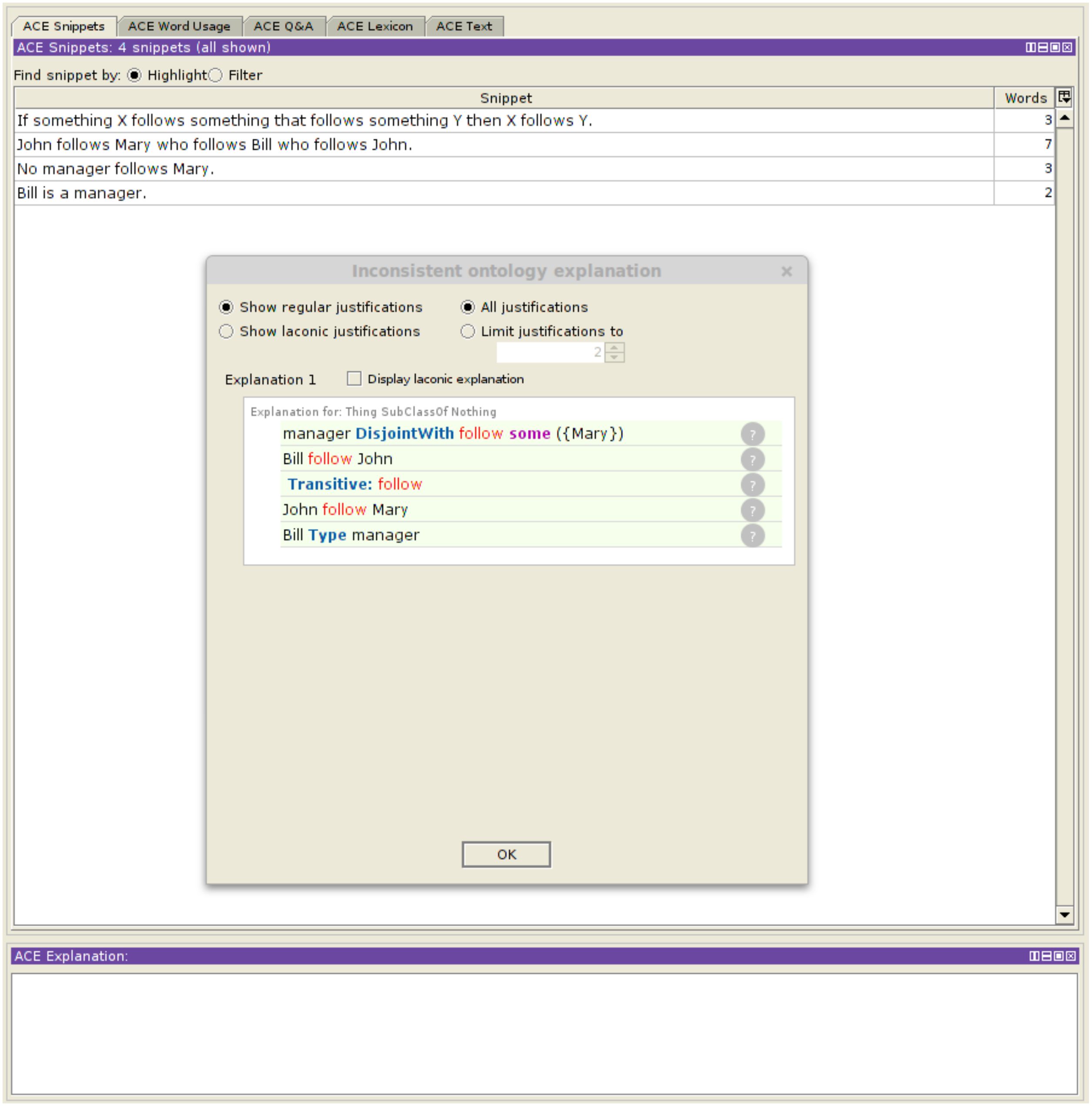
John **Type** man

John **Type** woman

Updated in 0.0 seconds

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

- 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.


ACE Snippets: 4 snippets (all shown)

Find snippet by: (●) Highlight ( ) Filter

Snippet	Words
If something X follows something that follows something Y then X follows Y.	3
John follows Mary who follows Bill who follows John.	7
No manager follows Mary.	3
Bill is a manager.	2

Inconsistent ontology explanation

(●) Show regular justifications (●) All justifications
( ) Show laconic justifications ( ) Limit justifications to 2

Explanation 1 ☐ Display laconic explanation

Explanation for: Thing SubClassOf Nothing

manager DisjointWith follow some ({Mary})
Bill follow John
Transitive: follow
John follow Mary
Bill Type manager

ACE Explanation:

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

3 Proof of concept problem - The impact of Open Water

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.

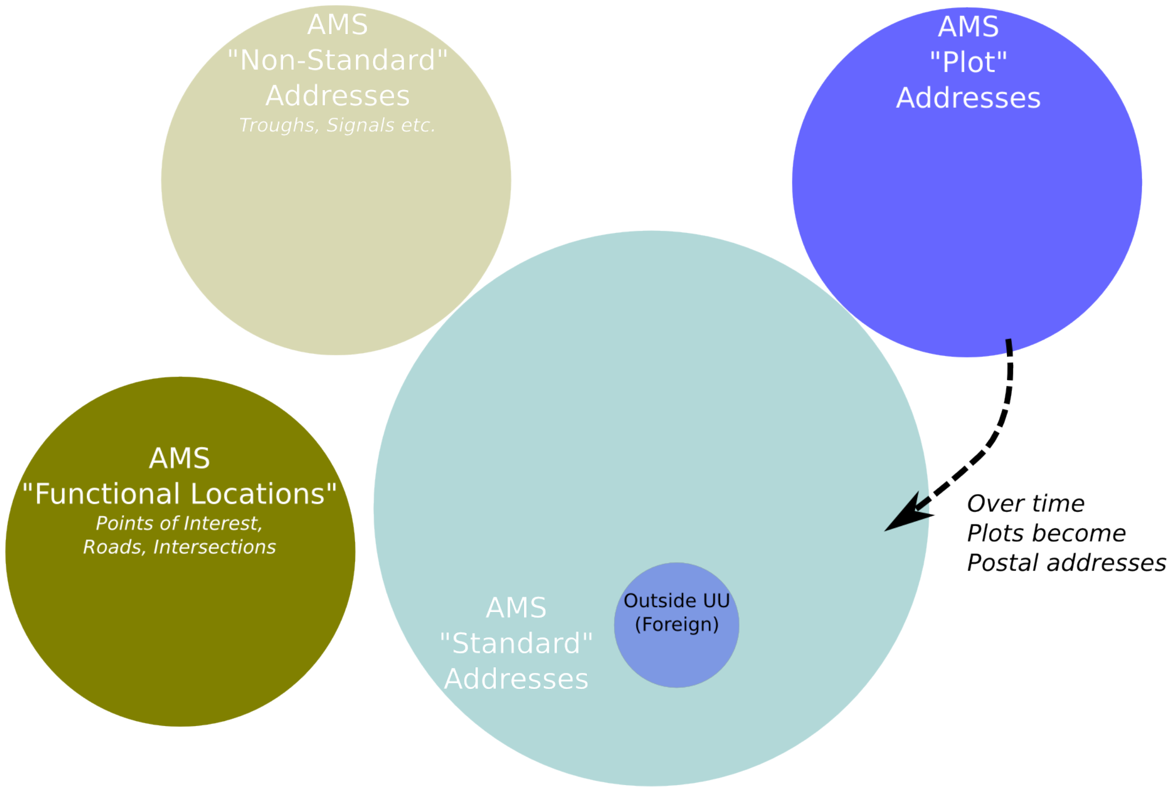


Figure 4: The UU Address Management System

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[6].

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

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[7]

[6]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

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

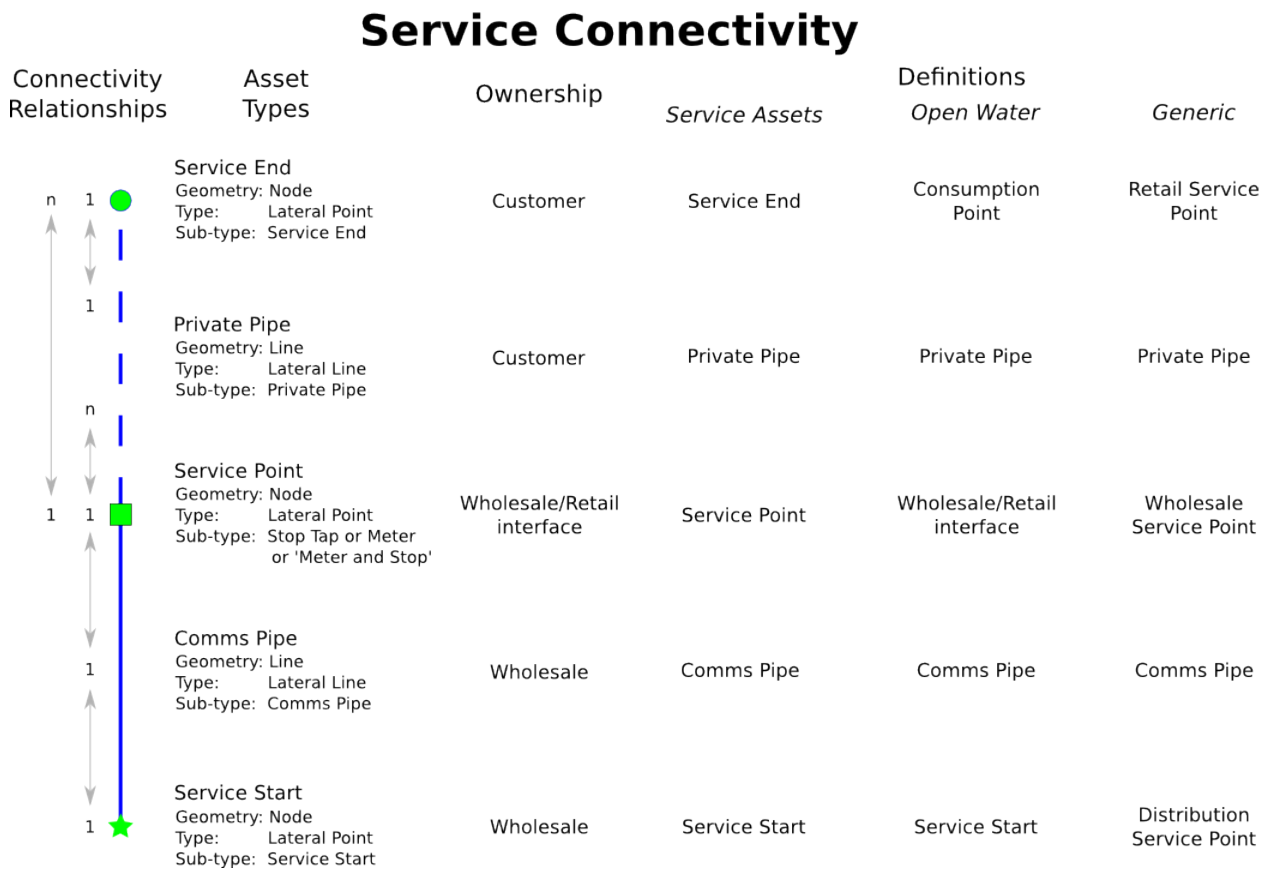


Figure 5: Service Point connectivity at United Utilities

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 .

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:

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

> Create some stories around the models in protege and the questions which have been asked.
>
> Also be honest that this (the CNL) currently has holes and needs some other things to support it (like the concepts in the supporting OWL). In this example the network connected relationships have not come through (because I need to hand crank the logic directly in Protege as it's not accepted in the statements)

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[8].

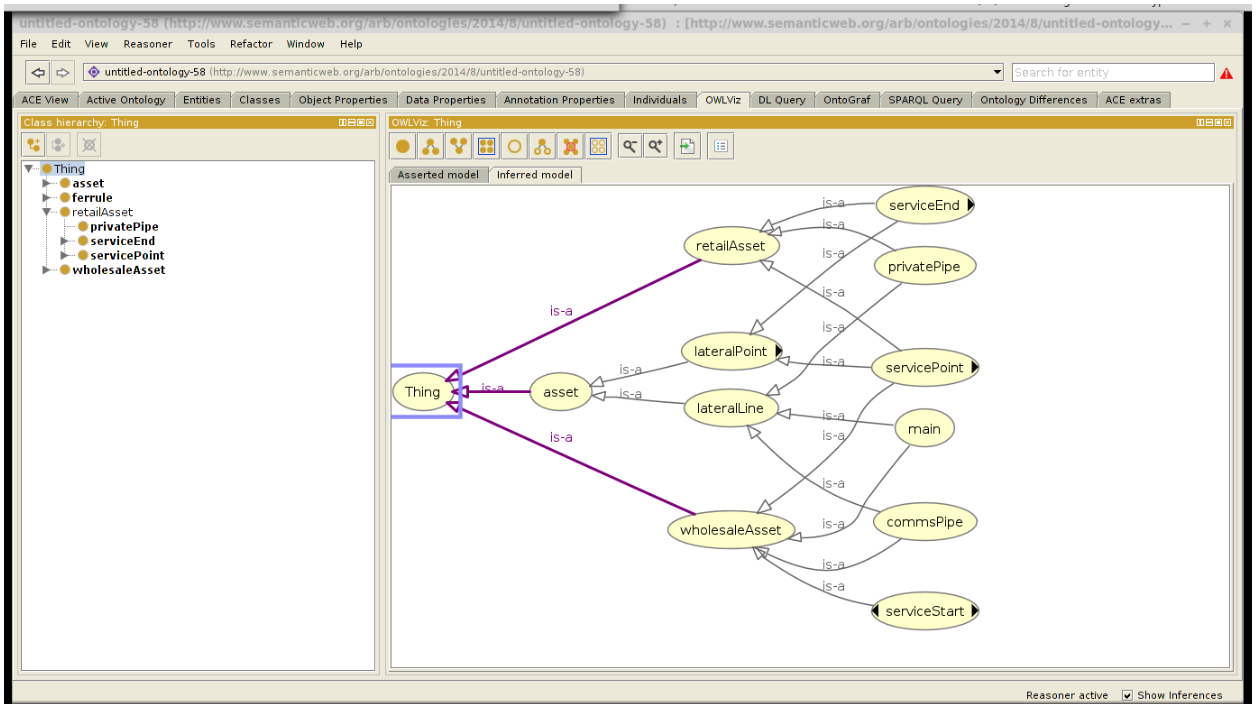


Figure 6: A visualization of the CNL concepts in protege

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[9].

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 .

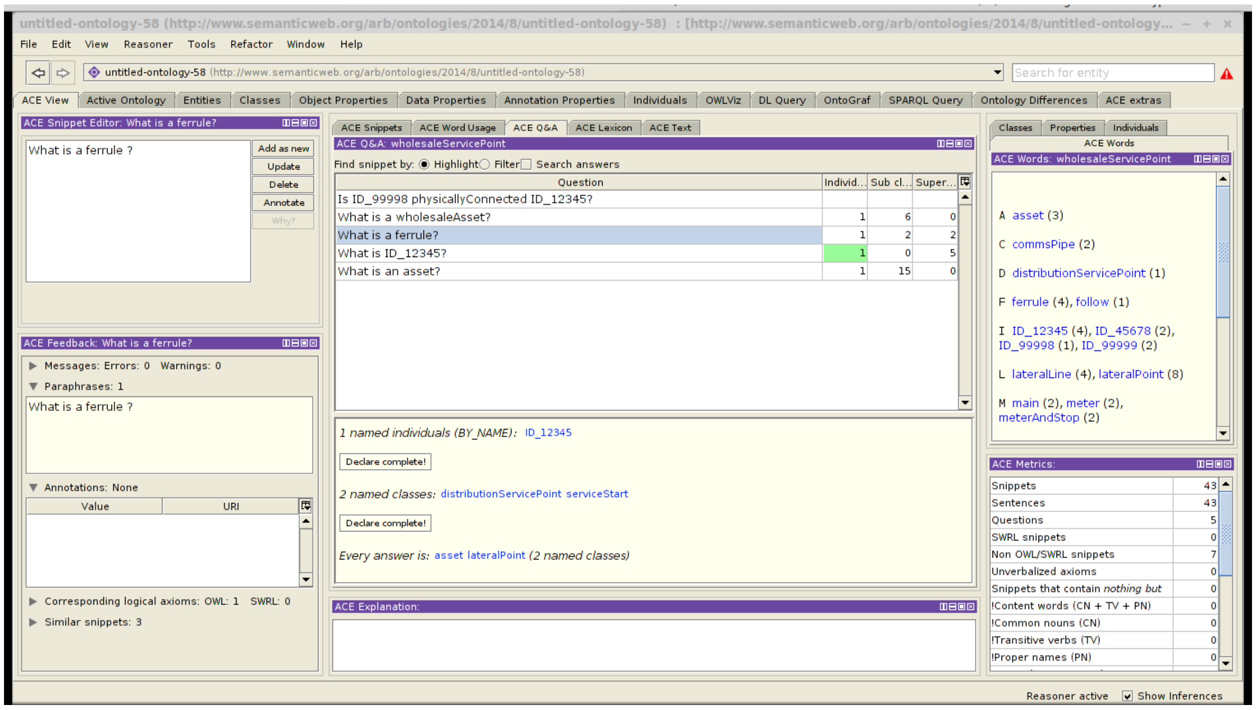
marketOperator is at most one thing .

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 .

[8]this is a bold statement - ensure this is run past Stephen Stanley first (and at an early stage)

[9]This is where I got up to. It's not finished and I haven't articulated everything


untitled-ontology-58 (http://www.semanticweb.org/arb/ontologies/2014/8/untitled-ontology-58) : [http://www.semanticweb.org/arb/ontologies/2014/8/untitled-ontology...
File Edit View Reasoner Tools Refactor Window Help
untitled-ontology-58 (http://www.semanticweb.org/arb/ontologies/2014/8/untitled-ontology-58)
Search for entity
ACE View | Active Ontology | Entities | Classes | Object Properties | Data Properties | Annotation Properties | Individuals | OWLViz | DL Query | OntoGraf | SPARQL Query | Ontology Differences | ACE extras
ACE Snippets | ACE Word Usage | ACE Q&A | ACE Lexicon | ACE Text
Classes | Properties | Individuals
Add as new
Update
Delete
Annotate
Why?
Declare complete!
Reasoner active
Show Inferences

ACE Snippet Editor: What is a ferrule?

What is a ferrule ?

ACE Feedback: What is a ferrule?

Messages: Errors: 0 Warnings: 0

Paraphrases: 1

What is a ferrule ?

Annotations: None

Value	URI

Corresponding logical axioms: OWL: 1 SWRL: 0

Similar snippets: 3

ACE Q&A: wholesaleServicePoint

Find snippet by: ● Highlight ○ Filter ☐ Search answers

Question	Individ...	Sub cl...	Super...
Is ID_99998 physicallyConnected ID_12345?			
What is a wholesaleAsset?	1	6	0
What is a ferrule?	1	2	2
What is ID_12345?	1	0	5
What is an asset?	1	15	0

1 named individuals (BY_NAME): ID_12345

2 named classes: distributionServicePoint serviceStart

Every answer is: asset lateralPoint *(2 named classes)*

ACE Explanation:

ACE Words

ACE Words: wholesaleServicePoint

A asset (3)

C commsPipe (2)

D distributionServicePoint (1)

F ferrule (4), follow (1)

I ID_12345 (4), ID_45678 (2), ID_99998 (1), ID_99999 (2)

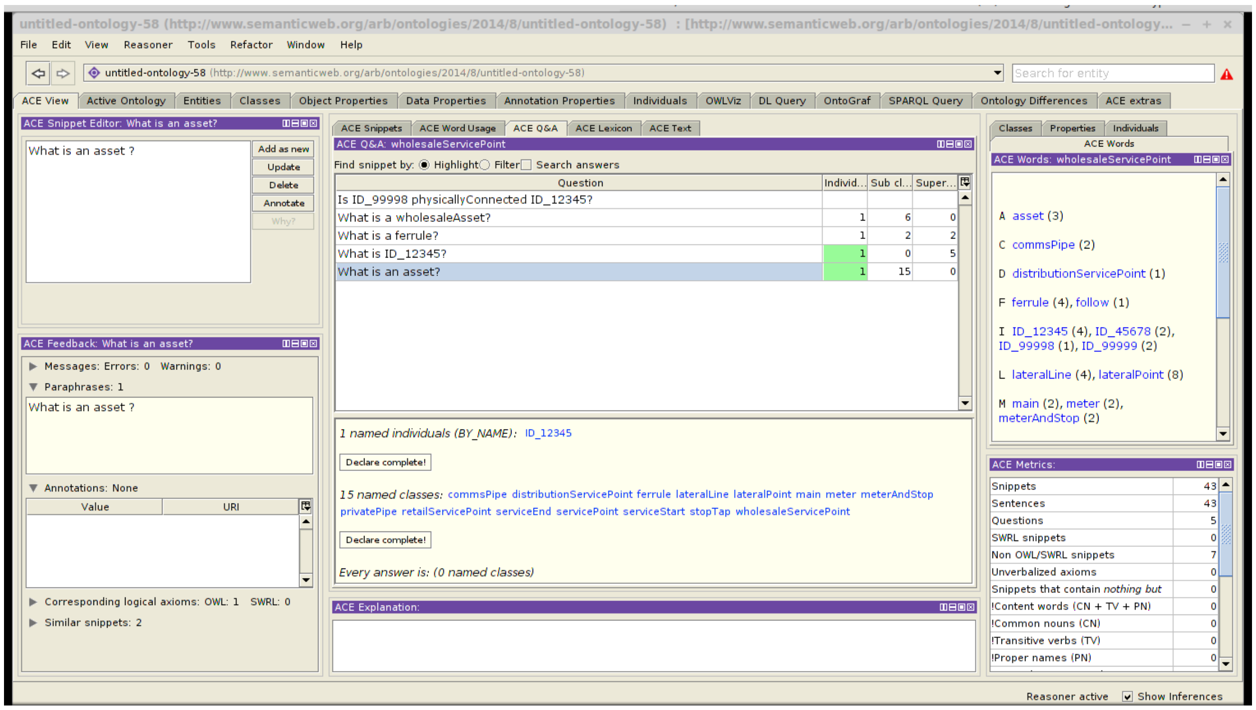
L lateralLine (4), lateralPoint (8)

M main (2), meter (2), meterAndStop (2)

ACE Metrics:

Snippets	43
Sentences	43
Questions	5
SWRL snippets	0
Non OWL/SWRL snippets	7
Unverbalized axioms	0
Snippets that contain *nothing but*	0
!Content words (CN + TV + PN)	0
!Common nouns (CN)	0
!Transitive verbs (TV)	0
!Proper names (PN)	0

Figure 7: What is a Ferrule?


untitled-ontology-58 (http://www.semanticweb.org/arb/ontologies/2014/8/untitled-ontology-58) : [http://www.semanticweb.org/arb/ontologies/2014/8/untitled-ontology...
File Edit View Reasoner Tools Refactor Window Help
untitled-ontology-58 (http://www.semanticweb.org/arb/ontologies/2014/8/untitled-ontology-58)
Search for entity
ACE View | Active Ontology | Entities | Classes | Object Properties | Data Properties | Annotation Properties | Individuals | OWLViz | DL Query | OntoGraf | SPARQL Query | Ontology Differences | ACE extras
ACE Snippets | ACE Word Usage | ACE Q&A | ACE Lexicon | ACE Text
Classes | Properties | Individuals
Add as new
Update
Delete
Annotate
Why?
Declare complete!
Reasoner active
Show Inferences

ACE Snippet Editor: What is an asset?

What is an asset ?

ACE Feedback: What is an asset?

Messages: Errors: 0 Warnings: 0

Paraphrases: 1

What is an asset ?

Annotations: None

Value	URI

Corresponding logical axioms: OWL: 1 SWRL: 0

Similar snippets: 2

ACE Q&A: wholesaleServicePoint

Find snippet by: ● Highlight ○ Filter ☐ Search answers

Question	Individ...	Sub cl...	Super...
Is ID_99998 physicallyConnected ID_12345?			
What is a wholesaleAsset?	1	6	0
What is a ferrule?	1	2	2
What is ID_12345?	1	0	5
What is an asset?	1	15	0

1 named individuals (BY_NAME): ID_12345

15 named classes: commsPipe distributionServicePoint ferrule lateralLine lateralPoint main meter meterAndStop privatePipe retailServicePoint serviceEnd servicePoint serviceStart stopTap wholesaleServicePoint

Every answer is: (0 named classes)

ACE Explanation:

ACE Words

ACE Words: wholesaleServicePoint

A asset (3)

C commsPipe (2)

D distributionServicePoint (1)

F ferrule (4), follow (1)

I ID_12345 (4), ID_45678 (2), ID_99998 (1), ID_99999 (2)

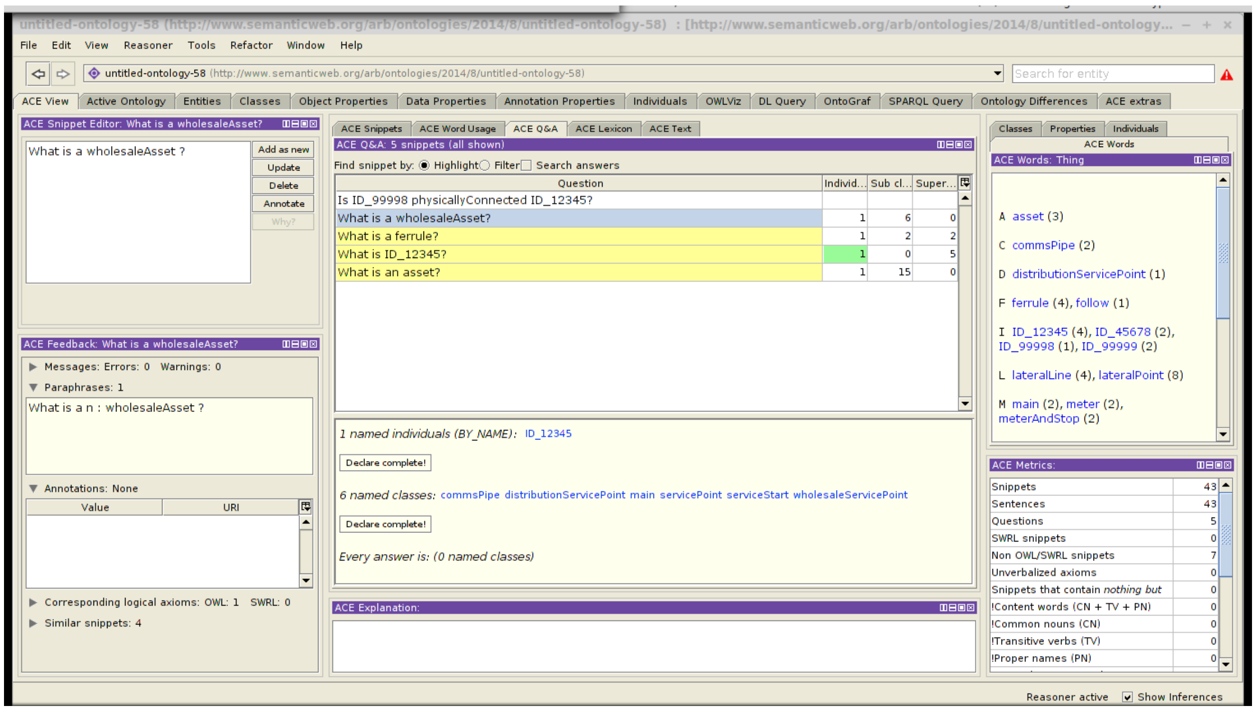
L lateralLine (4), lateralPoint (8)

M main (2), meter (2), meterAndStop (2)

ACE Metrics:

Snippets	43
Sentences	43
Questions	5
SWRL snippets	0
Non OWL/SWRL snippets	7
Unverbalized axioms	0
Snippets that contain *nothing but*	0
!Content words (CN + TV + PN)	0
!Common nouns (CN)	0
!Transitive verbs (TV)	0
!Proper names (PN)	0

Figure 8: What is an Asset?


untitled-ontology-58 (http://www.semanticweb.org/arb/ontologies/2014/8/untitled-ontology-58) : [http://www.semanticweb.org/arb/ontologies/2014/8/untitled-ontology...

ACE Snippet Editor: What is a wholesaleAsset?

What is a wholesaleAsset ?

ACE Feedback: What is a wholesaleAsset?

Messages: Errors: 0 Warnings: 0

Paraphrases: 1

What is a n : wholesaleAsset ?

Annotations: None

Corresponding logical axioms: OWL: 1 SWRL: 0

Similar snippets: 4

ACE Q&A: 5 snippets (all shown)

Find snippet by: Highlight Filter Search answers

Question	Individ...	Sub cl...	Super...
Is ID_99998 physicallyConnected ID_12345?			
What is a wholesaleAsset?	1	6	0
What is a ferrule?	1	2	2
What is ID_12345?	1	0	5
What is an asset?	1	15	0

1 named individuals (BY_NAME): ID_12345

Declare complete!

6 named classes: commsPipe distributionServicePoint main servicePoint serviceStart wholesaleServicePoint

Declare complete!

Every answer is: (0 named classes)

ACE Explanation:

ACE Words: Thing

A asset (3)

C commsPipe (2)

D distributionServicePoint (1)

F ferrule (4), follow (1)

I ID_12345 (4), ID_45678 (2), ID_99998 (1), ID_99999 (2)

L lateralLine (4), lateralPoint (8)

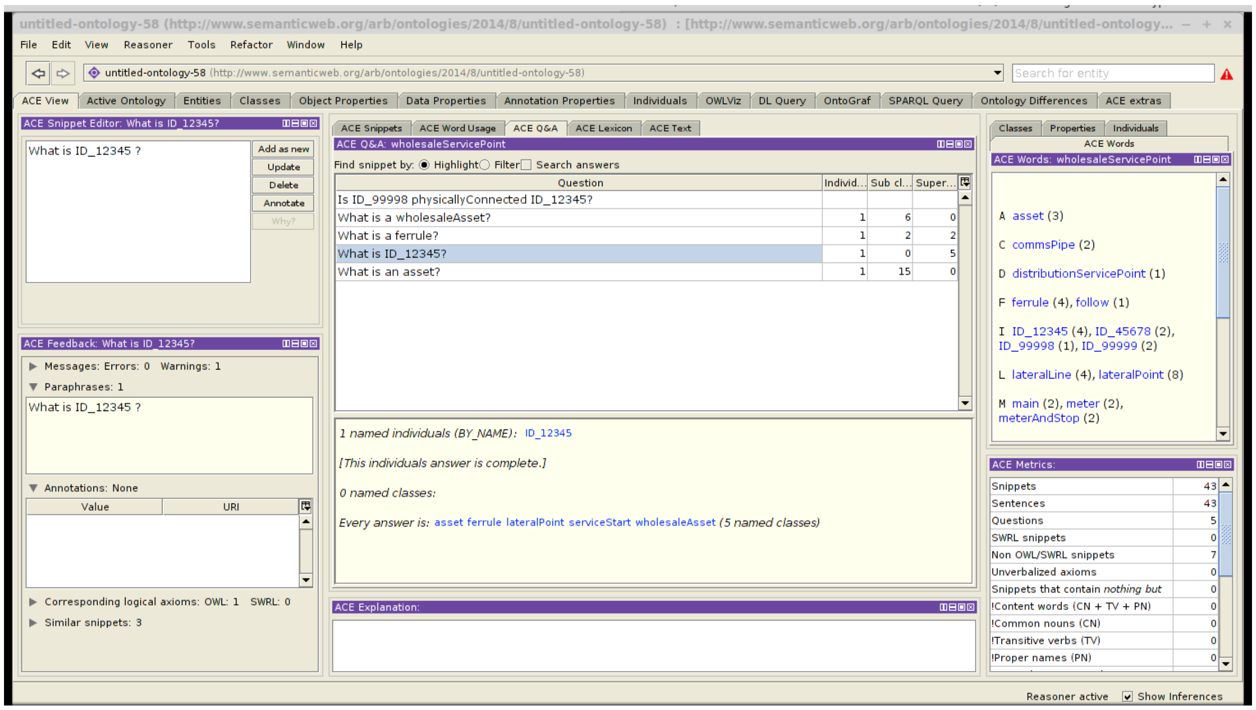
M main (2), meter (2), meterAndStop (2)

ACE Metrics:

Snippets	43
Sentences	43
Questions	5
SWRL snippets	0
Non OWL/SWRL snippets	7
Unverbalized axioms	0
Snippets that contain *nothing but*	0
!Content words (CN + TV + PN)	0
!Common nouns (CN)	0
!Transitive verbs (TV)	0
!Proper names (PN)	0

Reasoner active Show Inferences

Figure 9: What is a Wholesale Asset?


untitled-ontology-58 (http://www.semanticweb.org/arb/ontologies/2014/8/untitled-ontology-58) : [http://www.semanticweb.org/arb/ontologies/2014/8/untitled-ontology...

ACE Snippet Editor: What is ID_12345?

What is ID_12345 ?

ACE Feedback: What is ID_12345?

Messages: Errors: 0 Warnings: 1

Paraphrases: 1

What is ID_12345 ?

Annotations: None

Corresponding logical axioms: OWL: 1 SWRL: 0

Similar snippets: 3

ACE Q&A: wholesaleServicePoint

Find snippet by: Highlight Filter Search answers

Question	Individ...	Sub cl...	Super...
Is ID_99998 physicallyConnected ID_12345?			
What is a wholesaleAsset?	1	6	0
What is a ferrule?	1	2	2
What is ID_12345?	1	0	5
What is an asset?	1	15	0

1 named individuals (BY_NAME): ID_12345

[This individuals answer is complete.]

0 named classes:

Every answer is: asset ferrule lateralPoint serviceStart wholesaleAsset *(5 named classes)*

ACE Explanation:

ACE Words: wholesaleServicePoint

A asset (3)

C commsPipe (2)

D distributionServicePoint (1)

F ferrule (4), follow (1)

I ID_12345 (4), ID_45678 (2), ID_99998 (1), ID_99999 (2)

L lateralLine (4), lateralPoint (8)

M main (2), meter (2), meterAndStop (2)

ACE Metrics:

Snippets	43
Sentences	43
Questions	5
SWRL snippets	0
Non OWL/SWRL snippets	7
Unverbalized axioms	0
Snippets that contain *nothing but*	0
!Content words (CN + TV + PN)	0
!Common nouns (CN)	0
!Transitive verbs (TV)	0
!Proper names (PN)	0

Reasoner active Show Inferences

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 NOT DEALT WITH

- 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.

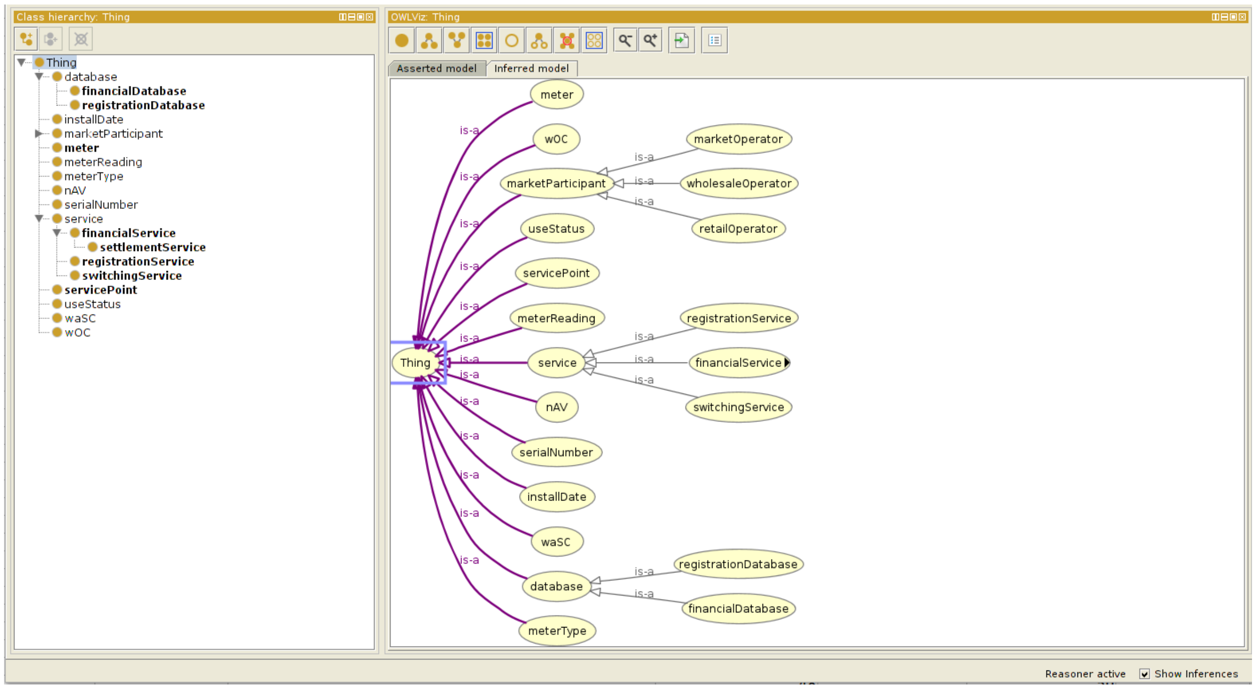


Figure 11: A visualization of the Systems Architecture and Data Model concepts in CNL

4 Summary and discussion

- 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 clients
- Store knowledge so it can be re-used in downstream projects and products
- Sell services based upon the knowledge model

5 Appendices

5.1 Writing OWL ontologies in ACE

The following text is structured to separately discuss the three main building blocks of ontologies as used in OWL: individuals, classes and properties.

5.1.1 Individuals

One can talk about individuals, classify them and describe relations between them. For example,

- There is a man. The man is John.
- His age is 31 and he likes Mary.
- Mary loves herself.
- Bill sees something.
- John is not Bill.

The following graph visualizes the model described by the story, with individuals (denoted by ellipses), their belonging to classes (boxes with labels), properties (solid arrows with labels), equality (dashed line) and difference (red line). Data-values are in a box with dotted border. In case the name of an individual is know then it is written into the ellips.

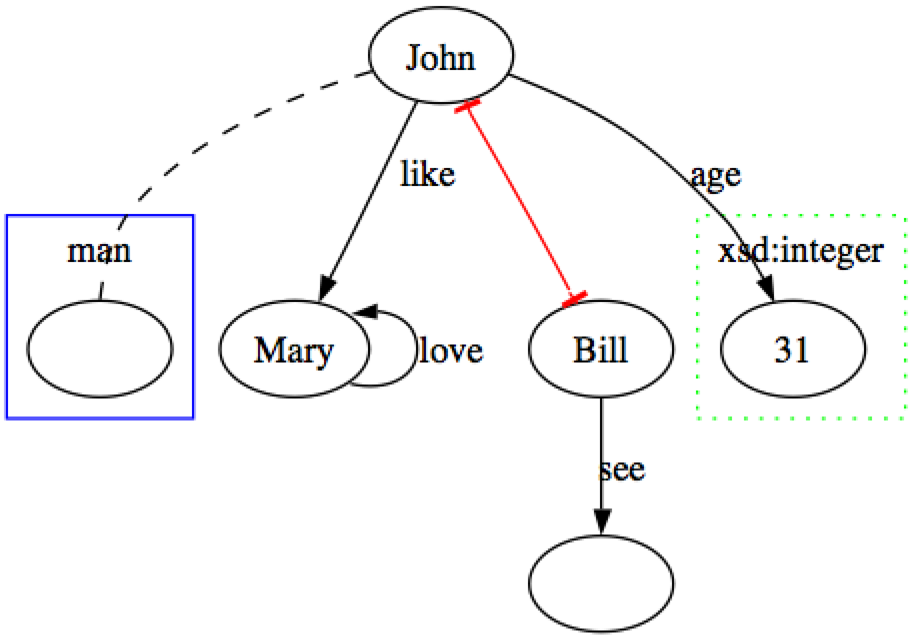


Figure 12: Picture. Representation of individuals as a Venn diagram.

The following should be noted about the above statements:

- *Individuals* are proper nouns that all start with a capital letter
  - I should discuss conventions (i.e Multiple Lettered Things and camelCase)

  - I should also say that common nouns can represent the base level of class membership[10]

OWL does not make a unique name assumption, i.e. an individual can have different names and one has to explicitly state the equivalence or difference of two differently named individuals. Therefore, whether Mary is John or is Bill or is different from both of them, is still open, we only know that John is not Bill. The fact that John belongs to the class *man* is not explicit in the picture, but can be derived by simple reasoning. Note also that John, Mary, Bill and the thing that Bill sees, all belong to the topmost class *owl:Thing*.

One can also classify the individuals by complex class descriptions, e.g.

- John is not a woman. (i.e. *John* belongs to the complement of the class *woman*)
- John is a programmer or is a manager. (i.e. *John* belongs to the union of classes *programmer* and *manager*)

5.1.2 Classes

One can describe classes by specifying relations to other classes. Such relations can be a simple inheritance from a named class ("Every man is a human.") or more complex property restrictions ("Every driver owns a car."). Remember that the concepts are common nouns in lower case. In OWL, the classes do not need names, they can be abstract descriptions, e.g. "(Every) man who owns a dog …". Such descriptions can be made arbitrarily complex by negation (*no*, *is not*, *does not*), disjunction (*or*, *or that*) and conjunction (*and*, *that*, *and that*).

Both the *every*-sentences ("Every man who has a driving-license drives a car.") and the *if-then* sentences ("If a man has a driving-license then he drives a car.") can be used. The *if-then* sentences are more flexible in the way the verbs can refer to existing nouns. However, it is a good idea to only use *every*-sentences

- Every man is a human. (man $\subseteq$ human)
- If there is a man then the man is a human. (man $\subseteq$ human)
- Every professor is a scientist and is a teacher. (professor $\subseteq$ scientist $\cap$ teacher)
- Everybody who is not a child is an adult. (owl:Thing $\cap$ $\neg$ child $\subseteq$ adult)

Equality of classes has to be explicitly defined, there is no shorthand construction like OWL's *Equivalent-Classes*.

- Every carnivore is a meat-eater and every meat-eater is a carnivore.

Disjointness of classes can be expressed via ACE's negation which maps to OWL's *ComplementOf*.

- Every man is not a woman. (man $\subseteq$ $\neg$ woman)
- No man is a woman. (man $\subseteq$ $\neg$ woman)
- Nothing is a man and is a woman. (owl:Thing $\subseteq$ $\neg$ (man $\cap$ woman))

Classes can also be described in terms of a property and its cardinality. (Note that the plural of 'thing' will be mapped to *owl:Thing*.)

- Every pipe connects at least 2 fittings. (pipe $\subseteq$ >= 2 connect fittings)

Reflexive pronouns ('itself', 'himself', 'herself') can be used to describe (ir)reflexivity.

[10]the common noun 'publicHouse' is the class whilst the proper noun'CoachAndHorses' is an individual

- Every narcissist likes himself.
- No river flows-into itself.
- No pipe connects itself.

Classes can also be described in terms of a property and its value that is either an individual or data-value.

- Every man knows John. (man $\subseteq \exists$ know {John})
- There is a woman who everybody likes. (w $\in$ woman; owl:Thing $\subseteq \exists$ like {w})
- John likes every dog. (dog $\subseteq \exists$ inv(like) {John})
- Every dog is liked by John. # the same using passive
- Everybody whose age is 32 is a grown-up.
- If there is a man who lives-in Paris then the man's address is "Paris".
- For all water it is false that the temperature of the water is -1.

where the construction is expressed by 'whose', 'man's', 'of the water', the property is expressed by 'age', 'address', 'temperature', and the data-value is '32', "'Paris"'and '-1'.

5.1.3 Properties

OWL properties correspond to ACE's verbs and transitive adjectives (e.g. 'taller than').

A superproperty for a given property can be defined as:

- Everybody who loves somebody likes him/her.

or in a more verbose manner:

- If there is somebody X and X loves somebody Y then X likes Y.

Disjointness of properties can be declared by using negation.

- Nobody who likes somebody hates him/her.

or more verbosely:

- If somebody X likes somebody Y then it is false that X hates Y.

Property chaining is a complex mathematical concept that does not have a simple verbalization in natural language. E.g. the transitivity of a property is formulated in ACE in the following ways (different levels of syntactic compactness are provided):

- 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 taller than something that is taller than something Y then X is taller than Y.

To declare an inverse property of a property, two sentences are needed.

- If something X is taller than something Y then Y is shorter than X.
- If something X is shorter than something Y then Y is taller than X.

The rest of the syntax that OWL provides for describing properties is just syntactic sugar, one can use the already described class relations, cardinality constraints and inverse properties instead.

Properties can have a domain and a range. The domain of a property can be expressed e.g. by

- Everybody who writes something is a human.

The range of a property can be expressed by:

- Everything that somebody writes is a book or is a paper.
- Everything that is written by somebody is a book or is a paper. # the same using passive

Properties can be functional and inverse functional. Functionality means that for a given subject, the object of the property (verb) is always the same. Inverse functionality means that for a given object the subject must always be the same. For functionality, one has to restrict the class *owl:Thing* to have at most one value for a given property (it is OK to have no values at all).

- Everybody orders at most one thing. (owl:Thing $\subseteq$ <= 1 order owl:Thing)

Inverse functional properties are expressed by switching the subject and the object of the verb.

- Everything is something that at most one thing orders. (owl:Thing $\subseteq$ <= 1 inv(order) owl:Thing)
- Everything is ordered by at most one thing. # using passive

Symmetric property is its own inverse. The various ACE syntaxes to express this are:

- If somebody X loves somebody Y then Y loves X.
- Everybody who is loved by somebody loves him/her.
- Everybody who somebody loves loves him/her.

Equivalent properties are expressed by declaring the two properties to be superproperties of each other.

- Everybody who hates somebody despises him/her.
- Everybody who despises somebody hates him/her.

5.1.4 Reasoning

User acceptance of the reasoning results obtained on the OWL level and perceived on the ACE level provides the basis for proving that the mapping from ACE to OWL is useful. Given the ACE text

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

the reasoner will find an inconsistency because John has been asserted as a man and men have been described as disjoint from women.

Here is another example.

- Every child is a man or is a woman.

- No child is a man.
- No child is a woman.

This text may seem strange, but it is not inconsistent, as far as OWL is concerned. All we have done, is constructed an empty class *child*, which is perfectly legal (one can say, though, that the class *child* is unsatisfiable). Only when we claim that there are elements (individuals) in this class, for instance by stating

- John sees a child.

do we end up being inconsistent.

Reasoning about property restrictions finds inconsistency in:

- Every man sees a dog.
- Nothing is a dog.
- John is a man.

Reasoning about properties of properties. The following text is consistent.

- No dog likes a cat.
- Fido is a dog.
- It loves a cat.

But it becomes inconsistent if we add information about the relation between 'like' and 'love':

- Everything that loves something likes it.

We can also reason about the domain and range of properties.

- Everything that somebody writes is a book.
- Everybody that writes something is a human.
- John who is a man writes a paper.
- Every man is a human.

Inconsistency is caused when we now claim that:

- No paper is a book.

Reasoning about cardinality, e.g. functional properties. Imagine a library policy saying that everybody can order only one item from the library, and that John places two orders.

- Everybody orders at most one thing.
- John orders a book X and orders a book Y.

This text is consistent, namely because the reasoner sees nothing strange about taking X and Y to denote the same item (individual). Only when we state that:

- X is not Y.

would it become inconsistent.

Reasoning about inverse functional properties is similar.

- Everything is ordered by at most one thing.
- Mary orders a book X.
- Bill orders the book X.
- Mary is not Bill.

Reasoning about transitivity and *HasValue* will find the following text inconsistent.

- 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.

Reasoning about inverse properties. In ACE, in many cases, one does not have to explicitly assert a named inverse property of a given property. One can simply use a passive verb to create an (unnamed) inverse property. An inconsistent text, containing an inverse property, can be constructed like this:

- John likes Mary.
- Nobody is liked by John.

This statement would be redundant:

- If somebody X likes somebody Y then Y is liked by X.

Reasoning about nominals (i.e. OWL's *OneOf* construct). The following text is inconsistent.

- Every student is somebody who is John or who is Mary.
- Bill is a student who is not John and who is not Mary.

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.