Ontology Supported Comparison in Vague Domains

Karsten Øster Lundqvist

Supervisors:

Prof Shirley Williams

Prof Keith Baker

Submitted for the degree of Doctor of Philosophy

Ambient & Pervasive Intelligence Research Group

School of Systems Engineering

University of Reading

2010

Abstract

It is problematic to use standard ontology tools when describing vague domains. Standard ontologies are designed to formally define one view of a domain, and although it is possible to define disagreeing statements, it is not advisable, as the resulting inferences could be incorrect. Two different solutions to the above problem in two different vague domains have been developed and are presented. The first domain is the knowledge base of conversational agents (chatbots). An ontological scripting language has been designed to access ontology data from within chatbot code. The solution developed is based on reifications of user statements. It enables a new layer of logics based on the different views of the users, enabling the body of knowledge to grow automatically. The second domain is competencies and competency frameworks. An ontological framework has been developed to model different competencies using the emergent standards. It enables comparison of competencies using a mix of linguistic logics and descriptive logics. The comparison results are non-binary, therefore not simple yes and no answers, highlighting the vague nature of the comparisons. The solution has been developed with small ontologies which can be added to and modified in order for the competency user to build a total picture that fits the user's purpose. Finally these two approaches are viewed in the light of how they could aid future work in vague domains, further work in both domains is described and also in other domains such as the semantic web. This demonstrates two different approaches to achieve inferences using standard ontology tools in vague domains.

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Karsten Ø Lundqvist

Acknowledgements

I would like to thank many people for getting me to a place, which I thought was impossible for somebody who grew up in Rødkærsbro:

I would like to dedicate this thesis to my beloved wife! She pushed me, pulled me and stopped me when I needed it. She has put up with my distracted mind many evenings, and surely deserves to see me in full attire after this. I would extend my thanks to the rest of my family for trying to understand this strange world I am working in.

Thanks also go out to both of my supervisors. Shirley's constant enthusiasm for our field, her willingness to "play ball" with my ideas and constant eagerness to explore new routes has been precious. I can only wonder where I would have been if Keith had not been in his office one summer's day in 2005 when I needed it the most. He has given me an opportunity for which I cannot thank him enough.

My colleagues at the University and especially in Odinlab have been a constant sanity check. Pat has been a source of constant eye-openers, silly word games and spelling checks. Will was a good critical friend. Richard, Patrick, Edwin and Rob have been excellent guinea pigs for ideas, both sensible and plain silly. The partners in various EU projects also deserve thanks for giving me opportunities to talk about my ideas to intelligent yet often "uninitiated" people. I especially think of Fabio, Gigliola, Thomas F and Thomas K, Michael, Marga, Serge, actually the list goes on probably indefinitely.

Table of Contents

List of Figures

1 Introduction

The thesis will explore and provide solutions for making inferences and comparisons between different concepts using ontological tools in vague domains. Ontologies provide, as explored in chapter 2, useful methodologies and a range of tools for defining and working with problems within crisp domains, i.e. domains where concepts can be formally defined with agreement between the community who uses them. However many domains are vague in their nature, and it is hard to use ontologies and their tools within such domains. The aim of the research within this thesis is to explore how normal ontological methods and tools can be used in novel ways to work within vague domains. Two different novel approaches are developed and investigated in two different vague domains, the domains of conversational agents (chatbots) and competency management.

1.1 Novel Aspects of Research

This section introduces the aspects of the thesis which are novel contributions to the field.

Within the domain of conversational agents a novel approach has been developed (chapter 4), which allows full access to the knowledgebase contained within available ontologies. This is achieved through the scripting language OwlLang, which has been developed for this purpose. This language also enables storage of new knowledge acquired through chat within the ontology, thus enabling a growing "world knowledge" which can be used in new chat instances. When this happens, new statements might be introduced into the ontologies of the chatbot which contradict the existing knowledge contained within them. A novel solution, that can used to make inferences even when disagreeing statements are present in the ontology, utilising heavy usage of reified statements (statements about statements), has been created which creates a “reified layer” of semantics within ontologies.

This enables the development of an ontology driven conversational agent system, where chat structures (trees of chat) can be developed, which is an often used approach to develop chatbots [1], [2], [3], integrating OwlLang statements, thereby inserting knowledge from connected ontologies at runtime. The benefit of this is that a chatbot developer can create a generic chatbot for a certain situation and by changing only the underlying ontologies the same chatbot can respond and converse about issues within separate domains. A generic survey chatbot has been developed and two different domain ontologies (the e-Learning tool Blackboard and a project website) have been used to test the ability to rapidly change the domain of the survey chatbot.

The second domain that is used within this thesis is that of competency and competency frameworks. As chapter 5 will show, this is a domain where even the definition of the words are disputed by experts. Many different competency frameworks exist which only adds to the vagueness of the domain; still the frameworks are being used in many different areas and levels of society. However, the vague nature makes it difficult to compare and contrast different competencies and competency frameworks. A novel approach to this is developed (chapter 6) which enable “relaxed” semantic comparisons with non-binary results. This was done (except where otherwise specified) solely by the author through work within the TRACE project [4]. To achieve this, a bespoke standard Very Simple Reusable Competency Mapping (VSRCM) has been developed based on the idea of competency mappings [5]. VSRCM is not an industry standard, as it has been developed solely for the work described in this thesis, however efforts, aided by the author, within IEEE working groups are developing a standard based on competency mappings. By using the VSRCM and the industry standard Reusable Competency Definition (RCD) competencies can be described and represented. The graph contained within the VSRCM allows the traversal of the competencies, thus comparing them using the logical relationships inherent in the connections of the graph, furthermore linguistic semantics are introduced to make semantic connections between different competencies. This allows comparisons of RCDs and VSRCMs based on a less stringent logical foundation. A comparison engine has been created which uses these features to compare different competencies, exploring whether they match each other's different competency components and proficiency levels. The approach to handling the vagueness of the domain is achieved through the usage of small ontologies which can be chosen based on appropriateness of the situation in which they are being used. An Application Programming Interface (API) has been created which 3rd parties can use to develop new services based on the underlying competency representation and comparison system.

1.2 Structure of the Thesis

The chapters of the thesis are structured in the following way.

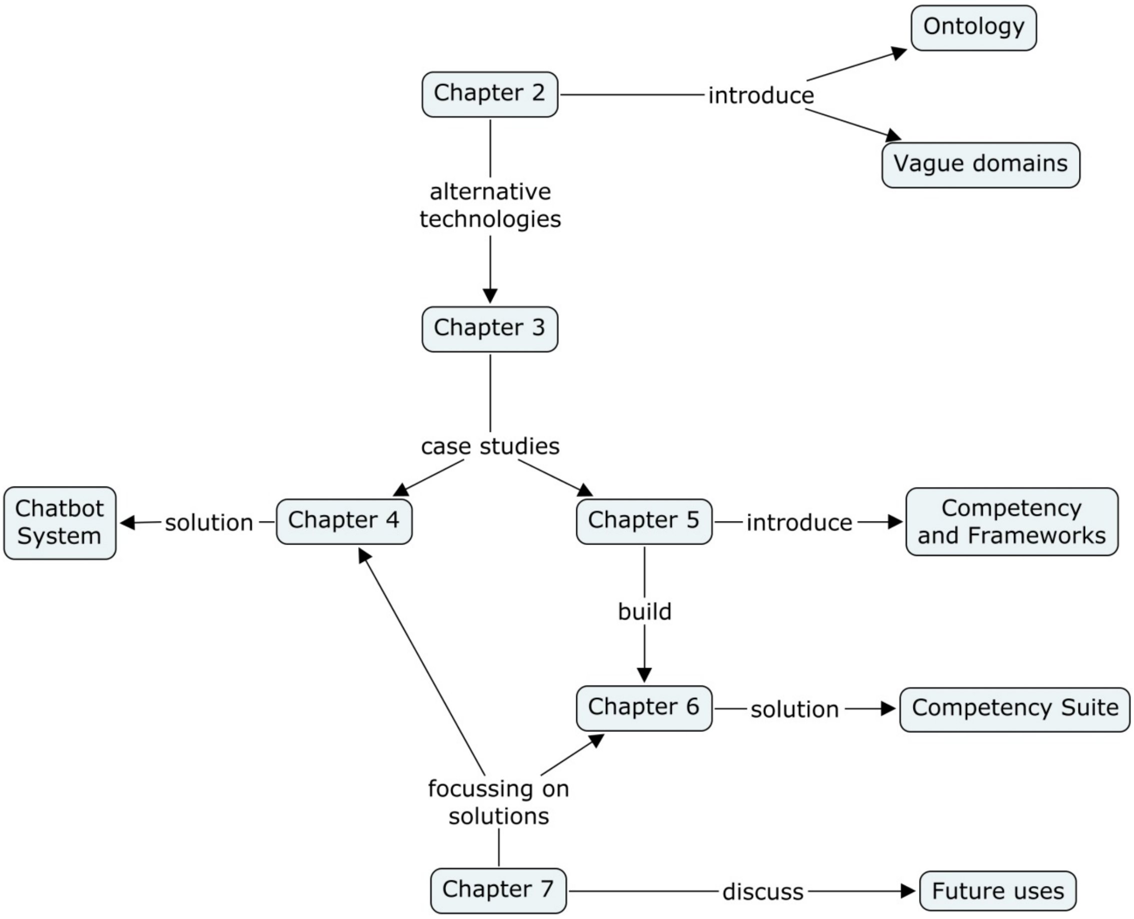


Figure 1.1 Thesis Structure

Chapter 2 will introduce, define and discuss ontologies and ontological tools through a literature review. Ontologies are used throughout the thesis as a core technology.

Chapter 3 will investigate how vague domains have traditionally been tackled in computer science (in literature) especially by using artificial intelligence (AI) techniques such as Neural Networks, Bayesian Networks and Fuzzy Logic. The prospects of Fuzzy Ontologies will also be investigated. The purpose of this is to show, by using the exemplars, that the use of ontologies in new ways could be beneficial.

Chapter 4 will briefly introduce the traditional chatbot design including advantages and disadvantages of the design choices. This forms the basis for the first case study, which is the subject of the rest of the chapter drawing on the ontological background from chapter 2, where an ontological conversational agent system (chatbot) is developed and tested in two different domains.

Chapter 5 will present and investigate the second vague domain, the domain of competencies and competency frameworks, as they are used in practice and presented in literature. This will be used as the basis for the second of the case studies of this thesis. Firstly, different definitions of competency are presented followed by examples of competency frameworks with three different purposes. Lastly the representation standards RCD and SRCM are presented and discussed.

Chapter 6 presents the novel work developed within the domain of competencies as part of this research. It is concerned with the implementation of an ontologically based system, an ontological "Competency Suite", which deals with description and comparison of competencies and competency frameworks. The development processes of the system are described. The competency standard RCD has been used and the SRCM has been transformed into a bespoke VSRCM, which is presented and discussed. The ontologies of the system are showcased. Finally the actual comparison algorithm is explained and examples are shown.

Chapter 7 investigates possible future developments which can be envisaged within the presented domains focussing on the two produced comparison solutions.

2 Ontology

The concept of ontology has, within computer science, been adopted and used in knowledge representation theory [6], [7], [8]. The artificial intelligence (AI) community explored the concept of ontology very early on for specifying domains, for instance in expert systems, and more lately they have become useful in more varied applications, especially for the Internet. The need for ontologies in computer science originates from the necessity to have guaranteed consistency between different computational entities, such as different applications and/or web sites, when sharing data and knowledge sources [6]. This section investigates what is meant by the term ontology and how ontologies are useful.

2.1 Terminology

The term “ontology” has been identified as a concept, which has gained popularity within the knowledge engineering community; however there is seemingly confusion of the actual usage and meaning of the term. Gruber defines ontology as a specification of a conceptualisation [9], however this is an ambiguous definition because the term conceptualisation is not clearly defined by Gruber. The following terminology [10] by Guarino is used throughout this thesis:

- **Ontology** (capital ‘O’): The classical philosophy which deals with the nature and organisation of reality.
- **conceptualisation**: An intensional semantic structure that encodes the implicit rules and constraints relating the structures of a piece of reality (not necessarily explicitly noted).

- **ontological commitment**: A (partial) semantic account of the intended conceptualisation of a logical theory (a logical language of the conceptualisation such as an ontology). Alternatively it can be said that entities (such as agents) are committed to an ontology (the formal specification of a conceptualisation) when their observable actions are consistent with the definitions of that ontology (the intensional semantic structures of the conceptualisation encoded with the ontology) [8].
- **ontological theory**: A set of formulae intended to always be true according to a given conceptualisation.
- **ontology** (lowercase 'o'): A logical theory that gives an explicit partial account of a conceptualisation. Alternatively a synonym of conceptualisation, a usage which was suggested to be avoided by Guarino.

2.1.1 Ontology (philosophy)

In philosophy Ontology is the study of "what is" or existence of things. This is an old school of thought with roots back to Aristotle, who wrote: "There is a science which investigates being as being and the attributes which belong to this in virtue of its own nature. Now this is not the same as any of the so-called special sciences; for none of these others treats universally of being as being. They cut off a part of being and investigate the attribute of this part; this is what the mathematical sciences for instance do." [11] This means that within (the philosophy) Ontology the question "What is being?" is studied, or perhaps more meaningfully, the common features of all beings are examined.

2.1.2 Conceptualisation

Ontology has been defined by Genesereth, from a traditional AI viewpoint, as "a triple of a universe of discourse, a functional basis set for that universe of discourse, and a relational basis set." [12]

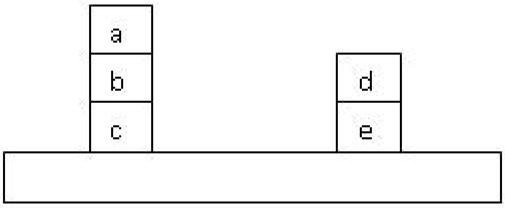


Figure 2.1 Block world conceptualisation example [12]

Considering the "block world" in figure 2.1 a person could view the figure as five boxes on a table, but somebody else could view it as something completely different. Using the boxes on a table conceptualisation of the figure, the universe of discourse could be the set of boxes on the table (it could also include the table, but for the purpose of this example the table is not included).

{a, b, c, d, e}

A functional basis set for the conceptualisation could then be the function *hat* that maps a block onto the block on top of it. So in this example the set would be:

hat = {‹b, a›, ‹c, b›, ‹e, d›}

A relational basis set in this conceptualisation could be the *table* relation that holds all blocks that rests on the table

table = {c, e}

The relation *on* could hold tuples of boxes where the first is immediately above the other (the "opposite" of *hat*), hence:

on = {‹a, b›, ‹b, c›, ‹d, e›}

Using these examples a possible conceptualisation of figure 2.1 would be

{{a, b, c, d, e}, {*hat, table*, *on*}}

Or formally by a structure:

‹**D**, **R**›

Where **D** is a domain of objects and **R** is a set of relationships on **D**. It is worth stressing that this is only a spatial conceptualisation and that many more could be envisaged which Genesereth and Nilsson [12] do in their example.

This view of a conceptualisation is however very restrictive as a completely new conceptualisation would be needed to describe the “block world” in figure 2.2. This is basically the same world, but the state-of-affairs is different, because the blocks are arranged differently.

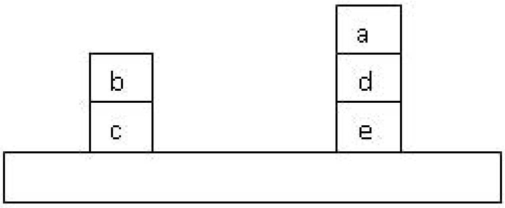


Figure 2.2 Another block world [10]

This problem is addressed by Guarino and Giaretta in [10]. They propose the use of an “intensional” structure, where “intensional relationships” are used instead of relations. The meaning of “intensional” is taken from intensional logics [13] where the “intension”, i.e. the meaning and understanding, of words is explored. For example the words “morning star” and “evening star” actually denote the same object but still do not necessarily mean the same thing. Guarino et al. propose that, instead of looking at the exact set making up a relation (for example the relation *table*), then use the "intensional relation" **table** which defines the intension of the relationships, thus enabling the conceptualisation to encode all "state-of-affairs" in the world of discourse instead of just one particular "state-of-affair".

Then a conceptualisation is represented by the intensional structure:

$\langle \mathbf{W}, \mathbf{D}, \mathfrak{R} \rangle$

Where **W** is the set of possible worlds, **D** is a domain of objects and $\mathfrak{R}$ is a set of intensional relationships on **D**.

2.1.3 Ontology and ontological commitment

An agent is said to make an ontological commitment when it can be observed to take actions which are consistent with the definitions in the ontology in question according to Gruber [8].

Guarino, however, defines it formally and differently in [7]. Given from the previous section we have:

C: Conceptualisation

$\mathbf{C} = < \mathbf{W}, \mathbf{D}, \mathfrak{R} >$, where **W** = possible worlds, **D** = domain and $\mathfrak{R}$ = conceptual relations

In traditional AI without intensional relations, a logical language **L** with a vocabulary **V**, would be defined as a model where:

$\mathbf{L} = \langle S, I \rangle$ where $S = \langle \mathbf{D}, \mathbf{R} \rangle$ is called the world structure comprising the domain (**D**) and the set of relations (**R**). I is an “interpretation function” assigning elements from the domain to constant symbols in the vocabulary, and elements of the set of relations to predicate symbols of the vocabulary. So

$I: \mathbf{V} \rightarrow \mathbf{D} \cup \mathbf{R}$.

But when considering intensional relations, this can be rearranged into an intensional model for **L** where:

L has the structure $\langle \mathbf{C}, \mathfrak{I} \rangle$. $\mathfrak{I}$ is an intensional interpretation function which assigns elements of **D** to constant symbols of **V** and elements of $\mathfrak{R}$ to predicate symbols of **V**. So

$\mathfrak{I}: \mathbf{V} \rightarrow \mathbf{D} \cup \mathfrak{R}$.

(**K**: Ontological commitment)

If $\mathbf{K} = \langle \mathbf{C}, \mathfrak{I} \rangle$ is an ontological commitment for **L**, then this is an intensional interpretation, where the language **L** commits to the conceptualisation **C** by means of the ontological commitment **K**, while **C** is the underlying conceptualisation of **K**

Guarino states in [7] that an ontology **O** for a language **L** approximates a conceptualisation, if there exists an ontological commitment $\mathbf{K} = \langle \mathbf{C}, \mathfrak{I} \rangle$, such that the intended models of **L** according **K** are included in the models of **O.** A language **L** commits to an ontology **O** if it commits to some conceptualisation **C** such that **O** agrees on **C**. Therefore, according to this, conceptualisations are language independent, but ontologies are not, although translations between languages are possible.

An ontology **O** can then be considered as a set of logical axioms designed to account for the intended meaning of a vocabulary, thus with a given language **L** with the ontological commitment **K** an ontology for **L** is a set of axioms designed so that the set of the models approximates as best as possible (see section 2.1.4) the set of intended models of **L** according to **K**. The ontology of a language only approximates the conceptualisation because normally it is not convenient to specify an ontology that does not admit other models besides the intended one.

An ontology therefore commits to a conceptualisation if it:

1. is designed to characterise the conceptualisation (not just commit by chance).
2. approximates the conceptualisation, this approximation could be a perfect representation although this is deemed to be close to impossible to attain.

After these formalisations Guarino clarifies Gruber's definition of ontology and ontological commitment as:

> "An ontology is a logical theory accounting for the *intended meaning* of a formal vocabulary, i.e. its *ontological commitment* to a particular *conceptualization* of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models."

Both interpretations of ontological commitment are used throughout, as they are not seen to be opposites, but rather extending each other. Agents use ontologies, and therefore, if the actions of the agent correspond to the actions expected by the ontology these commit to the conceptualisation that the ontology is approximating.

2.1.4 Different types of ontologies

From the above definition it is evident that many ontologies are possible that all commit to the same conceptualisation. Guarino therefore suggests a layered approach which accommodates for different levels of commitment to conceptualisations [7].

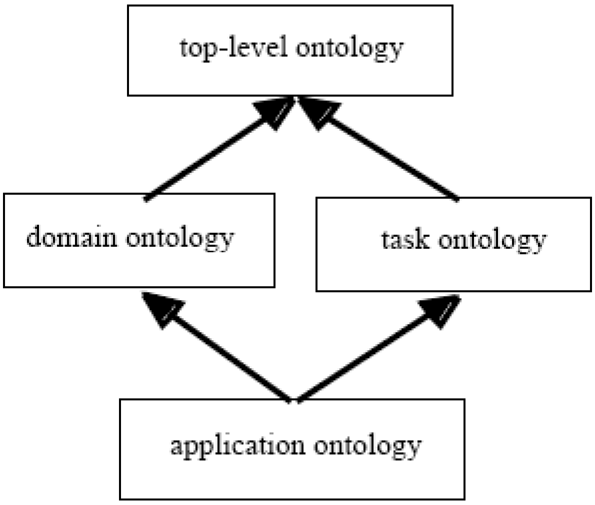


Figure 2.3 Layers of Ontology

- Top-level ontology (also called upper ontology) should describe high level general concepts, which are independent of domain or specific problems. This would typically be concepts like matter, object, event, time, etc. These should allow for large communities of users, thus enabling tools that work across domains and applications of all the users.
- Domain ontologies and task ontologies should define the vocabularies and intensional relations respectively of generic domains (e.g. medicine, competency or education) and generic tasks and activities (e.g. diagnosis, hiring

or reflection). This is achieved by specialising the terms defined in the top-level ontology layer. In practice these two ontology types are often put in a single combined domain and task ontology at this level with one ontology for each domain, as this is easier for the ontology engineer to practically do.

- Application ontologies describe specialised concepts that correspond to roles in the domain performing certain activities. These make use of both the domain and task ontologies.

Gruber [8] also examines how ontologies should be designed, especially where they are designed to have a guaranteed consistent operation between all entities (agents) that commit to them.

He establishes five different design criteria for the ontologies:

1. Clarity – the formalism of ontological languages ensures this.
2. Coherence – there should not be contradictions in the ontology.
3. Extendibility – develop with other uses in mind.
4. Minimal encoding bias – (an encoding bias results when representation choices are made purely for the convenience of notation or implementation.) This enables several computer systems to speak together.
5. Minimal ontological commitment – should only make the assumptions needed for agents to work together leaving the freedom of the agents to create the "worlds" they need internally.

2.2 Ontologies in practice

In practice ontologies are described in languages based on some form of logic (as described in section 2.1.3). There exists a wealth of knowledge representation languages, which can be used to formalise ontologies. The following figure 2.4 shows examples as presented by Martin [14].

Language:	Statement
E:	Tom owns a dog that is not Snoopy.
FE:	Tom is owner of a dog different_from Snoopy.
FCG:	[Tom, owner of: (a dog != Snoopy)]
KIF:	(exists ((?x dog)) (and (owner ?x Tom) (/= ?x Snoopy)))
XOWLL:	<Dog><owner><owl:Thing rdf:about="#Tom"/></owner> <owl:differentFrom rdf:resource="#Snoopy"/></Dog>
N3:	[a :Dog; :owner :Tom; owl:differentFrom :Snoopy].

Figure 2.4 Tom owns a dog [14]

Here "E" is normal English, which is the most expressive, but is not formalised. "FE" stands for formalised English, which is a formalised version of normal English. "FCG" is a language standing for "For Conceptual Graphs". "KIF" is the Knowledge Interchange Format which according to Martin is the most expressive of the languages as it "has been designed and used to specify the logical denotation of many other knowledge representation languages." "XOWLL" is OWL using RDF/XML notation [15] and "N3" is OWL using N3 notation [16].

The focus in this part of the literature review will be on the Web Ontology Language (OWL) which is the ontological language of the Semantic Web.

> The Semantic Web, as defined by W3C in [17], "is about two things. It is about common formats for integration and combination of data drawn from diverse sources, where on the original Web mainly concentrated on the interchange of documents. It is also about language for recording how the data relates to real world objects. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing."

The reason for using OWL instead of the more expressive language KIF is because OWL is a more modern language and has more supporting tools and libraries available. It would be very beneficial if these tools and libraries could be utilised in work within vague domains (e.g. for portability of knowledge and automated inferencing over semantic relationships), something which is not possible if they are just used in their intended way.

2.2.1 OWL

OWL is based upon the Descriptive Logic SHOIN(D), also known as SHOIN [18]. It uses RDF to describe statements and provides the structure of OWL underpinned by RDF schemas (RDFS) as its class structure. It is therefore possible to integrate into Internet data approaches by use of XML [19], although other RDF notations can be used, such as N-Triples and N3. There exist three different types of OWL; OWL-lite, OWL-DL and OWL-full. These are used in a similar fashion, and exist because of different needs of inferences on the ontologies they describe. In this thesis OWL-DL is used, however the differences are minute, the specific differences between them can be found in [15].

2.2.1.1 Class, Individual and Property

According to [15] at the core of OWL is the concept of a *Class* of individuals within the conceptualisation. By using classes it is possible to make inferences and assumptions about the individuals based upon their membership of a *Class*. In this thesis N3 will be used in the examples as it is perceived by the author as the more "human readable" notation. A tutorial by Berners-Lee can be found on the Internet [16]. For instance the following three lines of N3:

```
:Animal         a                owl:Class.
:Dog            a                owl:Class.
:Dog            rdf:subClassOf   :Animal.
```

would declare two OWL classes, one called *:Animal* and one called *:Dog*. (namespaces are left out in the examples and the "a" in N3 is equivalent to rdf:type). Furthermore it is declared that *:Dog* is a sub-class of *:Animal*, hence all individuals that are of type *:Dog* will have the same properties as any other individual of class *:Animal*.

It is now possible to declare individuals of each *Class***:**

```
:Fluffy         a                :Dog.
:Trofast        a                :Dog.
:BlubbyTheFish  a                :Animal.
```

All these individuals are now animals, and it is only *:BlubbyTheFish* which is only known to be an *:Animal*. It might be a *:Dog* or something else (most probably a fish but formally the ontology does not say that). Every individual will belong to the class *owl:Thing* as all classes are subclasses of that class.

This allows for the creation of taxonomies, which can be used to describe and define the possible worlds (**W**) and domain (**D**) of a conceptualisation, however, as is known from section 2.1.2, a conceptualisation also consists of intensional conceptual relations ($\mathfrak{R}$). OWL uses properties to define these relations. Properties can both be used as intensional relations and "normal" relations, since properties can be defined as possible (or absolutely necessary) properties of a specific *Class*, or as an actual property of an individual. There are two different types of properties; *DatatypeProperty* and *ObjectProperty*.

- DatatypeProperty: Properties which relate an individual of a Class to a literal, such as RDF literal or XML schema datatype.
- ObjectProperty: Properties which relate an individual of a Class to an individual of another Class.

It is also possible to define restrictions of properties such as domain, range, cardinality, some or all values of a certain kind of *Class* and a specific value for all individuals of a *Class*.

So the following statements are now possible, which define a property *:hasOwner* with the domain of *:Dog* and the range of *:Person*. *:Person* has not been defined. This is acceptable in OWL, because it is now possible to make the inference that there must be such a *Class* and *:Bo* is a *:Person*:

```
:hasOwner          a                 owl:ObjectProperty;
                   rdfs:domain       :Dog;
                   rdfs:range        :Person.
:numberOfLegs      a                 owl:DatatypeProperty;
                   rdfs:domain       :Animal;
                   rdfs:range        xsd:integer.
:Fluffy            :hasOwner         :Bo.
```

2.2.1.2 Property characteristics, ontology mapping and complex classes

In the previous section the only piece of information that any inference engine could use was connected to the property restrictions (i.e. knowing *:Bo* was a *:Person*). OWL inherently has many additional logical features that enable more “clever” inferencing. These can be grouped into three different types:

- Property characteristics: Definitions of logical characteristics of properties:
  - *TransitiveProperty*:

    F is a *TransitiveProperty* if for any a, b and c.

    F(a, b) and F(b, c) ⇒ F(a, c)
  - *SymmetricProperty*:

    F is a *SymmetricProperty* if for any a and b.

    F(a, b) iff F(b, a)

- *FunctionalProperty*

  F is a *FunctionalProperty* if for any a, b and c.

  F(a, b) and F(a, c) ⇒ b = c

- *InverseFunctionalProperty*

  F is a *InverseFunctionalProperty* if for any a, b and c .

  F(a, c) and F(b, c) ⇒ a = b

- *inverseOf*

  Property F1 is *inverseOf* property of F2 if for any a and b.

  F1(a, b) iff F2(b, a)

- Ontology mapping: Definitions that map different classes and individuals

  - *equivalentClass*:

    A is an *equivalentClass* of B if A and B are both classes and have the exact same individuals.

  - *equivalentProperty*

    F1 is an *equivalentProperty* of F2 if F1 and F2 are both properties and are making the exact same relations.

  - *sameAs*

    I1 *sameAs* I2 declares individual I1 to be the same individual as I2.

  - *differentFrom* and *AllDifferent*

    I1 *differentFrom* I2 declares individual I1 to be different from individual I2.

    By using *AllDifferent* it is possible to create a list of individuals where each member of the list is a distinct individual.

- Complex classes: A *Class* can be defined using other Classes as references
    - Basic set operators:
        - *intersectionOf*

          This defines a new class from the intersection of two other classes.

          The following is an example of all dogs with owners. The class is the intersection of the class :Dog and the anonymous class (the construct “[:a :b]” creates an anonymous resource, which in this case is a class made out of a restriction) that have a property :hasOwner which has a value of something (an individual of class owl:Thing):

          ```
          :houseDog     owl:intersectionOf
          (
                              :Dog
                              [
                              a                   owl:Restriction;
                              owl:OnProperty      :hasOwner;
                              owl:someValuesFrom owl:Thing
                              ]
          ) .
          ```

        - *unionOf*

          This defines a new class from the union of two other classes.

          This example class is all individuals of class :Cat and :Dog:

          ```
          :Pet    owl:unionOf   (:Cat   :Dog) .
          ```

- *complementOf*

  This defines a new class of all individuals that are not members of the class.

  The example is the class of all dogs without owners, which is the intersection of all individuals of class :Dog and the anonymous class which is the complement of :houseDog. If :Dog had been left out everything else but individuals of :Dog, e.g. :Bo, would have been an individual of class :wildDog:

  ```
  :wildDog        owl:intersectionOf
              (
                          :Dog
                          [ owl:complementOf   :houseDog]
              ).
  ```

- *oneOf*

  This defines a class by enumeration, hence a class which only consists of the values presented.

  The example is a class of all lights in a traffic light:

  ```
  :trafficLightColour     owl:oneOf      (:green :yellow :red).
  ```

- *disjointWith*

  This defines a class that guarantees that an individual that is a member of the class cannot simultaneously be an instance of the other specified class(es).

  The example is a further specification of :Dog:

  :Dog owl:disjointWith :Cat ,
  :Crocodile ,
  :Bird ,
  :Fish ,
  :Person .

2.2.1.3 SPARQL

W3C recommends that a user of RDF graphs, and therefore also OWL ontologies, make queries by using SPARQL [20]. SPARQL has lately been standardised by W3C [21], which follows a long period with multiple competing query languages where SPARQL emerged to become the *de facto* standard with support from W3C. It is a comprehensive query language which resembles SQL (Structured Query Language), the query language for relational databases [22], thus this will purely function as an example of what is possible with SPARQL queries. The following are the four different types of query statements in SPARQL with examples of their usage;

- ASK

  A simple query whether a stated graph is present in the RDF graph

  ASK {:Dog a :Cat} returns "no".

- SELECT

Returns values from the RDF graph based on variables in the stated graph.

SELECT ?x {?x a :Dog} returns one list of all individuals of class :Dog.

SELECT ?x ?y {?x a ?y} return one list of all individuals of any class together with their class.

- CONSTRUCT

  Constructs a new graph by finding each query solution of a stated graph and add them to another.

  CONSTRUCT {?x a :housedog} WHERE {?x a :Dog} returns a new graph where all individuals of class :Dog are changed to class :housedog.

- DESCRIBE

  Returns a graph for each query solution of what else is known about the solution.

  DESCRIBE ?x {:x :hasOwner :Bo} returns a graph containing :Fluffy and all relations that are known about :Fluffy, for instance that it is an individual of class :Dog.

2.2.1.4 Inference

Inference is an important aspect of ontology driven applications, and inference has already been mentioned in the previous section when describing the different properties that exists in OWL. OWL is based on the Descriptive Logic SHOIN, thus enabling basic inferences based on the SHOIN rule set.

It is possible, when using N3 notation, to expand this set of rules (section 2.2.1.2) by user defined rules. For instance it is possible to create rules like [23]:

```
{?x   :hasOwner   ?y }   =>   {?y   a   :dogLover}
```

which states that any owner of a :Dog is also a :dogLover.

This example is a very simple example, but this method can be used to describe very complicated “business” rules in the conceptualisation described by the ontology. An OWL tool or library is needed if rules like this are required in standard OWL.

2.3 Using OWL

There exist many libraries and toolsets that support and use OWL due to the fact that it is the ontological description language of the Semantic Web. This section is not an exhaustive list of these tools and libraries, but describes the most common and helpful for this work, which are available at the time of writing.

2.3.1 Model libraries

There are several libraries for using ontology models. Most of them only support RDF models, for instance Sesame [24] and Redland [25], hence leaving inferencing to the application level, as they do not even support RDFS inferences. The reason for these technologies is that inferencing can be very computational intensive, thus it can be more efficient to do all inferencing in the business rules of an application to target the exact level of inference needed.

The most prominent library which supports OWL is Jena (supported by Hewlett-Packard [26]), which itself started out as an application programming interface (API) supporting RDF and RDFS in the Java programming language. It has been extended into the Jena 2 Ontology API which offers full programmatic support for OWL models. Besides allowing programmatic creation of OWL ontologies using Java statements the API also incorporates:

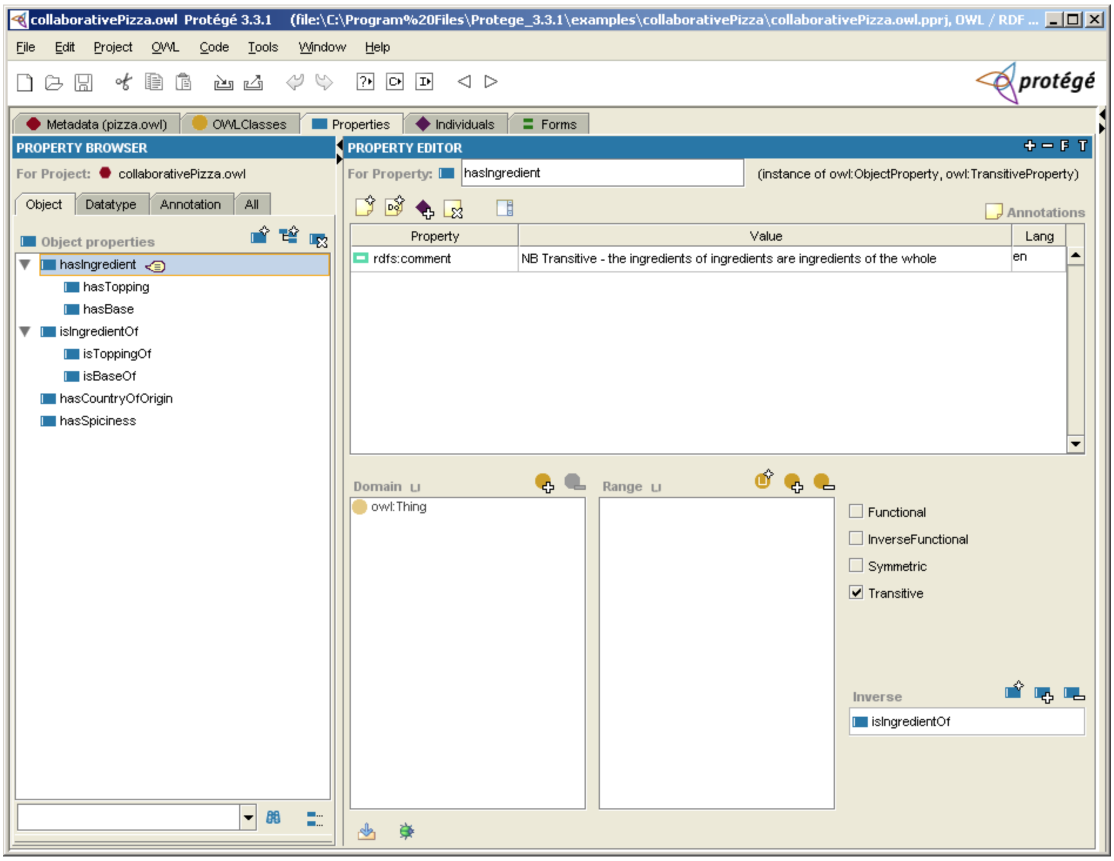
- Different data models for RDF
    - Memory model
    - Persistent database models
- Several inference engines and reasoners
    - Ready-made reasoners supporting different levels of inference, starting with a simple transitive engine all the way to OWL DL inferencing.
    - A rule based inference engine for user defined rules.
    - DIG reasoning support [27]. DIG is an XML based inference protocol for integrating third party reasoners.
- Query support through ARQ which is Jena's SPARQL implementation
    - Joseki is a web server built for Jena queries over the internet via HTTP [28] .

CWM [29] is a Python based general purpose Semantic Web OWL data processor with incorporated inferencing and ontology language translator modules.

2.3.2 Ontology editors

There are many editors with different approaches to the process of ontology development available, ranging from something as simple as notepad, and editor plug-ins to fully fledged applications with graphical user interfaces.

Protégé [30] is a commonly used ontology editor with an easy to use graphical user interface. The data model of Protégé is underpinned by a bespoke frame-based ontology model (OBKC) there is, however, an OWL plug-in which provides full support of OWL ontology creation and maintenance through Protégé. Additionally it supports DIG inferencing for third party inference engines to evaluate ontology consistency.


collaborativePizza.owl Protégé 3.3.1 (file:\C:\Program%20Files\Protege_3.3.1\examples\collaborativePizza\collaborativePizza.owl.pprj, OWL / RDF ...

For Project: collaborativePizza.owl

- hasIngredient
  - hasTopping
  - hasBase
- isIngredientOf
  - isToppingOf
  - isBaseOf
- hasCountryOfOrigin
- hasSpiciness

For Property: hasIngredient (instance of owl:ObjectProperty, owl:TransitiveProperty)

Property	Value	Lang
rdfs:comment	NB Transitive - the ingredients of ingredients are ingredients of the whole	en

Domain: owl:Thing

Range:

☐ Functional
☐ InverseFunctional
☐ Symmetric
☑ Transitive

Inverse: isIngredientOf

Figure 2.5 Protegé screenshot

Knoodl is another ontology editor which integrates a formal ontology specification (using OWL natively) into a WIKI based authoring system for collaborative ontology creation [31].

Another commonly used ontology editor, which is more simplistic, is the text editor XEmacs with a supported OWL plug-in [32].

2.3.3 Ontology Inference engines

There are also many different ontology reasoners / inference engines available. Some of the ontology modelling libraries have, as mentioned in section 2.3.1, integrated inferencing engines in them. It is also possible to use third party inference engines; the reason for this is that these are optimised for performance when inferencing which is a resource demanding and computational heavy task.

RacerPro is a commercial inference engine, which has support up to OWL-DL, with the possibility of user defined extensions. It can be used through DIG and can be integrated into Java and LISP through APIs. It has, additionally, full support of SPARQL queries [33].

Open source alternatives exist in the form of Pellet [34], which is a Java implementation with full Jena API and DIG protocol support, and FaCT++ a C++ implementation of the common lisp FaCT reasoner which also supports the DIG protocol [35].

2.3.4 Application of ontology

There are many examples of ontology in practice. One good example is within biological and medical domains where many ontologies exist to allow scientists to work using common vocabularies. One important project is Snomed [36] which is a standardised set of medical ontologies functioning as an international clinical terminology used by, amongst others, the NHS in the UK, National Institutes of Health in America and non-English speaking countries such as Denmark and the Netherlands. Another example is Open Biomedical Ontologies [37] an umbrella foundation which collects and coordinates ontology work in this area.

Other scientific organisations and companies are also using ontologies, for instance;

- NASA which maintain the “Semantic Web for Earth and Environmental Terminology” [38], which is used in a variety of scientific organisation around the globe.
- General Motors which has developed the “Variation-Reduction Adviser” which is a knowledge system for automotive assembly plants enabled by ontologies [39], which is used to enable ‘intelligent’ searches.
- There are also ontologies of language, for example Cyc, the open source version openCyc [40] and WordNet [41]. All of which classify the English language creating interrelations between the different words, such as synonymy and generalisation.
- The Semantic Web with its tools is, of course, an example of how ontologies could be used on a grander scale to enable better data interoperability, although it is still at an emergent state of development.

2.4 Criticism of ontology

There are diverse opinions around the theory and use of ontologies. The following sections illustrate this by describing two extremes in current academic debate.

2.4.1 “Concept” is too vague

Some ontology researchers believe that the notion of concept, as referred to in conceptualisation, which is used in the traditional definition of ontology by Gruber [9] and Guarino [7] is an astonishing choice. This has, according to these sceptics, lead to several misconceptions in the computer science branch of ontological knowledge representation.

Smith argues against these misconceptions by observing that [42]:

- There is no such thing as objective reality.
- It cannot be know what objective reality is like, so that there is no practical benefit to be gained from the attempt to establish such a correspondence.
- ‘Reality’ is a constructed term built on concepts, therefore any concept-system can claim to be a ‘reality’.

Smith follows this by looking at the reasons why traditional knowledge representation ontologists believe there is an objective reality. He claims that it stems from two wrong assertions made by traditional ontologists:

- Knowledge exists only in the mind of humans, thus it is relative to our minds, and hence our knowledge cannot be of concepts as they are merely relevant to ourselves.
- Knowledge is known to be fallible, and therefore ontologies should take account of fallible information.

Smith argues that these assertions are wrong, firstly humans can describe “universals” (or as he puts it “representation of entities as they exist in reality”), and thus going back to a discipline closer to what Aristotle defined as Ontology. Secondly any knowledge can become fallible when new knowledge arises; this does not mean that knowledge cannot be described.

This is somewhat of an oxymoron as Smith advocates the usage of “universals”, and how can a “universal” be fallible; however in this school of thought “universals” are exactly universal, it is the instances of “universals” which change. [43]

Smith also argues that the usage of Description Logics to define ontologies in knowledge representation transforms the ontology to a mathematical abstraction (which computer science ontologists would call a “precise semantic”), which according to Smith is far removed from normal understanding of semantics (the interplay between terms and meanings) [42].

This criticism suggests that ontologies, as described in the previous sections, are too vague, and that the practitioners ought to make ontologies by describing “universals” as they are in the real world, thereby describing something that scientists can use for advancing their science.

2.4.2 Ontologies will become too precise

A contrasting criticism of ontologies, made by Parry, can be stated in one sentence: “an ontology that perfectly expresses one persons understanding of the world is useless for anyone else with a different view of the world.” [44] So the more that an ontology may be usable for one school of thought, the less usable it probably will be for another. This is a paradox for ontologies as they are meant as a tool for knowledge sharing and allowing consistent sharing between different agents [6]

A similar criticism is made by Cory Doctorow in Metacrap [45], where he states that Schemas (and thus ontologies, as ontologies are a subset of schemas) always will be skewed by the person (or entity) who created them. Additionally there are always several ways of viewing the same thing; sometimes these different ways are quite contradictory, as Doctorow amusingly states it: "No, I'm not watching cartoons! It's cultural anthropology." Requiring people to use the same vocabulary (i.e. an ontology) would result in homogeneity of ideas, something that Doctorow views as unreasonable. Moreover, in a clear attack on the Semantic Web efforts, it is assessed that it is impractical to expect ordinary people to start using ontologies when creating content on the Internet; as Doctorow states it, they are unreliable (liars), lazy and / or stupid.

This criticism attacks the very basis of the criticism described in the previous section, by stating that there is no universal truth and that different people (and even experts) have different viewpoints of the world (and indeed their expertise), hence it is nonsense to be creating ontologies of shared truth. Standard ontological tools deal with truth or not truth (and perhaps unknown), which renders them unusable in many cases.

2.5 Vague Domains

It was presented in section 2.4.1 that some formal ontologists believe that the ontological tools presented in section 2.1 – 2.3 are too vague, hence a change is needed to use universals (statements that are always facts / truths) in ontology design. However is it always possible to do this? What about domains where experts have very divergent ideas of what is and is not true? There are plenty of domains, especially in the life sciences and humanities, where it seems impossible to find two experts who agree on even the simplest domain concepts! Consider a domain such as politics that by its very nature is based on disagreement, would it be impossible to create an ontology based system to support this kind of domain? Even the domain of formal ontology is a very good example of this, as seen in section 2.4.1; everything down to the definition of a formal ontology is disputed. So if an ontology engineer were to create an ontology of formal ontology, it would be difficult to describe it in a consistent way, and would necessarily have to describe the different viewpoints in the ontology.

The notion of ill-defined domains has been used within the AI community, where a well-defined domain (the opposite concept) is defined as "one (domain) in which there exists a systematic way to determine when a proposed solution is acceptable." [46]. The problem that is addressed here is, however, different from this definition, as comparisons within domains are investigated, where people (experts and normal people alike) disagree about how things relate within the domain, even the simplest of relationships. Additionally the problem that is considered here is not necessarily to find solutions to problems, but rather to find and define links within domains in order to enable inferencing and comparisons. Therefore the term ill-defined cannot be used here, thus the term a vague domain is used.

Questions could be asked regarding why it would be beneficial to use ontology tools, clearly created to be used in precise domains, in vague domains. Consider the example where the task of a computer system is to gather different ideas, such as in survey software. Could such a system benefit from the philosophy behind ontologies? If ontology tools could be created for this it might be possible to describe the functionality of the system using a task ontology and then by changing the domain ontology allow the system to work in many different domains. Additionally the wealth of tools and approaches developed to work with ontologies, described in this chapter, could be used with vague domains, giving the benefits that these tools provide. This would, at present, be difficult as the domain, and perhaps even the task, ontology would necessarily have to describe divergent viewpoints, in order to allow the capture of them by the survey. In the following chapters the ways ontology tools can be utilised within vague domains will be investigated and researched.

2.6 Summary

This chapter introduced and defined ontology and the theory of ontologies. Then ontologies and ontological tools where investigated in practice, as well as how these are often used in applications for knowledge representation and inferencing. Ontology editors and inference engines were also introduced.

Two critical views of ontologies were then discussed. The chapter ended with a discussion of vague domains and ontologies concluding that ontologies, in the normal usage of ontologies, cannot be used within vague domains, and concluded that approaches that would enable their usage would be beneficial.

The next chapter will look at alternative technologies and methods that are used within vague domains. Chapter 4 and 6 will introduce two novel approaches that use ontologies and ontological tools within vague domains.

3 Existing Technologies for Supporting Vague Domains

3.1 Introduction

This chapter introduces some approaches which have proven to be effective solutions or show some promising features when applied to vague domains. The aim of this introduction is to highlight the need for an approach based on a standard ontology toolset but which can also support vague domains.

3.1.1 Vague Domain

Vague domains were introduced in section 2.5. Most domains have a certain degree of vagueness attached to them; this is definitely seen to be the case in the life sciences such as psychology, sociology, ethics, humanities, but is also the case in the more “hard” sciences, such as political science, economy, law, computer science and business management where several schools of thoughts are “living” side by side, sometimes in conflict with each other, over even basic concepts of their science. It is not the aim of this thesis to prove the necessity of supporting technologies for vague domains, nor is it the aim to disprove the critiques of Smith in section 2.4.1, as there might be a place for ‘universals’ in ontology design, however it is simply observed that some domains might not have well defined ‘universals’ as domain experts disagree about what these ‘universals’ ought to be.

3.1.2 Technologies

Throughout this chapter several methods and technologies that support vague domains and aid the creation of tools that work with vagueness will be presented. This is not an exhaustive list, however it provides the background evidence to support the fact that there is a need for using traditional ontology tools using modified methods in order to create tools within vague domains.

3.2 Semantic Web

The Semantic Web was briefly introduced in Chapter 2 through some of the ontology technologies which are used in this thesis. It is an approach which enables applications to work on data which is vague in an automated application.

3.2.1 The “Old” Internet

The Internet (or Web) without the Semantic Web layer is, at its core, only for humans to consume information by viewing and listening to the content provided through the browser. Other computer applications cannot easily use the information as the data is formatted for use by humans. It was (and to a large extent still is) merely designed as an information space to be consumed by people. This was not the intention of the inventor of the Web, Tim Berners-Lee; he envisaged an Internet where computers could share and compute using the data to help the exchange of information. [6]

3.2.2 The Data Web

From the realisation that the traditional Internet failed to provide the computer-to-computer data consumption layer, which Berners-Lee also calls the Data Web [47], he started to work towards realising the Data Web. In order to realise such a web, the (at that time) emergent technologies of XML, XML schemas, XML-RDF and RDFS would not be sufficient to allow exchange information between different computer systems, as these technologies would only facilitate exchange of data between disparate computer systems. They do not contain the necessary semantic values in order for computer systems to “know” and be able to use such information consistently. For instance from

Figure 3.1, how would a system know that a "friend" would be the same kind of entity as an "employee" and that "Where" would be the same as "Place"?

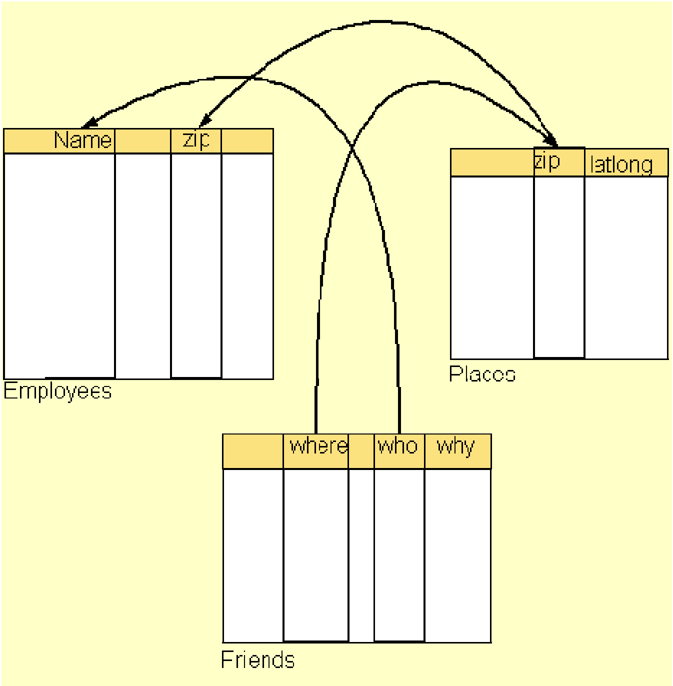


Figure 3.1 Semantic links from [6]

For this kind of semantic inference to take place new web technologies would be needed [6]:

- Ontology Language: OWL (see section 2.3)
    - Conversion between concepts (semantic relations).
    - Logical Layer with rule creation (N3 and bespoke rule creation solutions in inference engines).
- Query Language: SPARQL (see section 2.2.1.3).
- Digital Signature for the system to know categorically who created the assertions and documents to allow assertions about trust and proof: Through "XML Signature Syntax and Processing" RFC 3075 [48]. However none of these are as yet integrated into OWL engines.

3.2.3 Vague Domains – Advantages and Disadvantages

The Semantic Web is, at its core, designed to use ontologies to describe vague domains, and allow automatic processing by computers, and it is a very promising effort in this direction. This is also the reason for choosing the technologies of the Semantic Web as the underpinning technologies for the case studies used in this thesis.

However there are some problems associated with this approach. First and foremost for the Semantic Web tools to be able to handle vague domains, a fully functional digital signature system would be needed as it would enable inferences based on "who said what", in a consistent manner (an alternative identification system based on inferences about user statement could also be envisaged, but this would not rely on pure Semantic Web tools). Unfortunately this layer of the Semantic Web is not yet available and integrated into the stack of tools. As mentioned in the previous section a specification from the W3C has been developed, however there are problems with this approach [49], Loeb rather cynically lists some issues:

- Who controls the verification of the Digital Signatures, and thus controls the system? It is an area which could easily be monopolised, and / or misused by big corporations. Especially by companies who have access to monopolised Operating Systems, browser technology and who have been heavily involved in the creation of the standard.
- Digital Signatures could be stolen; just as credit cards operated by banks should be safe, they could be stolen and misused. It is not possible to say that a digital signature system would ever be safe. Using digital signatures is just redirecting the authentication burden down the chain, and as such is not really a solution.

Even if Digital Signatures were safe, they would only ensure the knowledge of who said something, but this person could potentially own many Digital Signatures, hence skew the data by “spamming”, and therefore potentially harm any automation efforts.

Another issue with digital signatures and the Semantic Web is the question of why people would start using them. At the moment the Semantic Web only makes up a tiny proportion of the Internet, and until many users take up this technology, it does not make sense to start using Digital Signatures. However before people start using them, the real benefit of the Semantic Web will not be achieved.

There are also issues with the actual ontologies which are being developed through the Semantic Web effort, since it is expected that many people will be developing the ontologies, many different versions of ontologies will be arising which fit to the needs of the ontology author. This is probably a good thing, as it would allow the highest level of flexibility, allowing competing ontologies to evolve and diversify the potential of the ontological specifications. Normally there would need to be semantic translation tools between the ontologies using the rule based inference engines of the Semantic Web allowing automatic transformations when needed. However what if the competing ontologies, either unconsciously or on purpose, have been designed so that they cannot be translated automatically?

A simple example of this can be found in something as simple as personal detail ontologies. Several ontologies and methods are available, two commonly used are FOAF (Friend Of A friend) [50] and vCard [51], [52]; even though these two ontologies formally specify the same domain, there are several places where these two specifications diverge, for instance vCard has fields for additional-name, address of a person and several different e-mail types, however FOAF does not have any way of providing additional-names, except as part of full-names; there is only the option to specify where a person lives approximately (using based_near) and there is only one e-mail type specified. On the other hand FOAF can specify many different types of homepages associated with a person, something which vCard does not allow. This makes automatic translation difficult, especially without losing information in the translation process. This is not a problem in this example; however it highlights the core of the problems with vague domains. People will create different ontologies, depending on their view of the domain, and on the intended usage of the ontology. Berners-Lee argues that this is not a problem within the Semantic Web as the Semantic Web technologies work with limited knowledge and will try to use any of the knowledge contained in the metadata, and that semantic knowledge can be retrospectively changed, something that was not possible in Knowledge Representation efforts before the semantic web [53]. The problem with this is that in order for it to work, the users have to be proficient in the use of the technologies to make such modifications.

In [53] Berners-Lee also claims that most webpages would only need simple RDF expressions of access control lists, privacy preferences, and search criteria. However if this is right, how would the elaborated inferences that normally are described as expected outcomes of the Semantic Web be possible? Good inferences need good metadata, and the average web designer cannot be expected to be an ontology engineer. Tools might be designed in the future to support web designers, but at present they are not available. Additionally the statements made by Doctorow in the cynical Metacrap

[45] highlight the problems that the Semantic Web might be facing with bad or wrong metadata.

To conclude, the Semantic Web is a good example to study, when looking at how to utilise ontologies in vague domains, however by allowing everybody access to add and share the ontologies with everybody else, the results of inference will probably be problematic. Additionally web designers cannot be expected to supply the amounts of metadata that would be necessary for the inferences to be useful.

3.2.4 RDFa: Bridging the Human Web and the Data Web

The W3C has worked in parallel with the Semantic Web specifications on a specification called RDFa [54]. The goal of this specification is to allow website creators to embed references into vocabularies and ontologies within XHTML, thus enabling the creator to specify the semantic relationships that form part of the web content. This provides an easy way of adding semantic web information to actual web content allowing both human users and "computer users" easy access to the content.

This approach is, however, not a solution to the problems of vague domains. It is merely a tool for web authors to provide the interface for automation of their content. The authors still rely on the correctness and suitability of the underlying vocabularies and ontologies. Additionally, several, if not all, of the points made by Doctorow [45] still apply to RDFa content, "people still lie, are lazy, are stupid and do not know themselves."

The RDFa standard is, at its core, a useful standard to embed the semantics of ontologies into actual web content, and therefore also vague domains, however it does not solve the problem caused by vague domains nor enable the description or usage of them.

3.3 Statistics

Statistical methods can be used to model diversity and trends within and between domains. They are a very effective way to create a “picture” of a domain for people to analyse it. Furthermore statistical methods can be utilised to make inferences about a “population” through inferential statistics [55] by using a sample of the “population” and taking advantage of probability distributions. For instance if it is known that some event happens on average every 10 minutes, how likely is it then that it will happen 5 times in the next 10 minutes? Or if 10 people out of 100 interviewees believe that X is true, how likely is it that a majority of people in the whole population believes that X is true?

Ontologies and statistics are therefore both used to describe domains and support making inferences about the domains; furthermore statistical methods have a firm base in set theory and logics [56] as do ontology. However while statistics are used to infer knowledge based on uncertain information, traditional ontologies are used to make inferences based on statements that define certainties and facts (or perhaps perceived facts), and this difference make them quite different tools. However it might be possible to combine statistical methods with ontological tools, creating a new basis for inference.

One of the issues when using statistical tools in conjunction with ontologies would be that the specified domains cannot always be statistically surveyed. For instance the domain in question might have a small "population" (10s rather than 1000s), thus making the resulting inferences void or meaningless. Additionally the domain might be artificially constructed, to describe certain aspects of a science for example, thus making it impossible to find samples of the domain, as it does not exist. Furthermore there will always be issues with bias in the statistical tools, e.g. biased selection, design issues of the statistical models and the questions which are asked. However there seems to be promising prospects in combining the use of statistical and ontology methods.

3.4 Folksonomy

Vander Wal [57], who coined the term, defines a folksonomy as: "Folksonomy is the result of personal free tagging of information and objects (anything with a URL) for one's own retrieval. The tagging is done in a social environment (usually shared and open to others). Folksonomy is created from the act of tagging by the person consuming the information." Tagging is a process, where a user is relating a keyword to a resource in some meaningful way (for the user) [58], therefore in other words, a folksonomy is the collective semantic meaning created out of the semantic value-giving activities performed by a cohort of people.

There are many existing folksonomies available on the Internet today, they are one of the products of what is termed Web2.0, which is a term for the Internet services which make content (in the broadest term of the word) creation and sharing easy for the non-tech savvy users, making the web more like a platform of services than a content delivery service [59]. The following list includes examples of services which can be described as folksonomies:

- Del.icio.us (http://delicious.com/)

  A service which enables users to bookmark webpages through the use of tags. The collection of relations between tags and websites provides a basis of semantics which users can traverse.

- Flickr (http://www.flickr.com/)

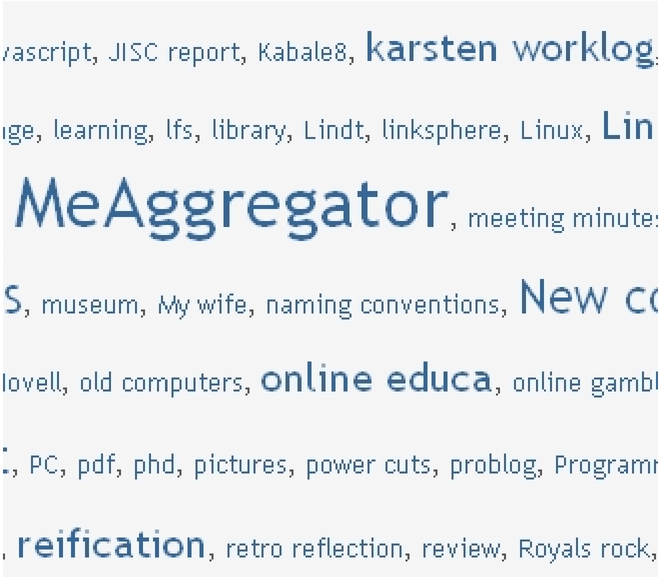
  A tag enabled photo sharing service. The tags can be used to define the semantics of the photo, which can describe the content of the photo, or simply some metadata about it, such as it was taken in 'sept05'.

- Technorati (http://technorati.com/)

  Blog aggregation service, which by collecting tags from blogs provides a valuable collection of recently used tags in the blogosphere.

Folksonomies can, arguably, be described as an ongoing free-format statistical user survey where the consumer decides exactly what statistical questions they want to ask, and from what viewpoint they want to see it, albeit the answers are normally based purely on the number of user interactions (e.g. the number of a specific tags of a particular blog makes it easier to find for the user). The big difference is that the statistical population, the actual data being used to generate data, are gathered because the subjects get something out of using the service (the actual service of the site), thus more and more content is gathered, and the folksonomy automatically becomes better and better. One problem with folksonomies stems from the fact that embedded semantics are produced by the aggregation of many people's "value-adding" actions, which is sometime referred to as the wisdom of the crowd [60]. The question is, whether the aggregation of knowledge by ordinary people will provide valuable or just "wrong" knowledge about a domain. Comparing this to statistical methods, this problem is potentially greater because the naming selection of the tags is completely free format, and has not been designed by anybody with an aim or plan (this can also be viewed as one of the biggest strengths of a folksonomy). On the other hand no expert will have tried to skew the information; if anybody were to do this it would be the users of the site, who might poison data intentionally (spammers, political activists, vandals etc.).

The biggest problem with folksonomies at present is that the aggregated semantic knowledge is only understandable and usable by human users through the different views the disparate services provides to their users. A commonly created view of the semantics contained in a folksonomy is the tag cloud, which is a listing of commonly used tags with differing font sizes, bigger font equalling more usages.



ascript, JISC report, Kabale8, karsten worklog,

ge, learning, lfs, library, Lindt, linksphere, Linux, Lin

MeAggregator, meeting minute

S, museum, My wife, naming conventions, New c

lovell, old computers, online educa, online gambl

, PC, pdf, phd, pictures, power cuts, problog, Programr

, reification, retro reflection, review, Royals rock,

Figure 3.2 Tag cloud from Redgloo 27-05-2010 (http://redgloo.sse.reading.ac.uk)

Other ways of traversing the semantic connections in the folksonomy include manually finding connections by using the tags as the basis of a search query. For instance, if

Petra is interested in the Android mobile phone platform, she can just type Android in, and hope that the users of the folksonomy have made connections between the tag Android and some other content. If many people have done this, there is a greater likelihood that the content that shows up is the most interesting and / or appropriate.

Some of the services have APIs (e.g. Technorati and Del.icio.us) which allow programmatic access to their services; however these provide individual access to each service, and therefore make inter-service relationships hard to assess. It has been proposed (and provisionally examined), by the author, to reify semantic relationships inherent in folksonomies through ontologies, through what is termed Folksonomological Reification [61]. One proposed way of storing the semantic knowledge is through reifications between an ontology and the folksonomy. The different statements in the ontology, which normally would be ‘universals’, are then backed up by reified statements that further specify the individual statements, such as who, when and where they were defined, thus enabling creation of diverging statements within the same semantic specification.

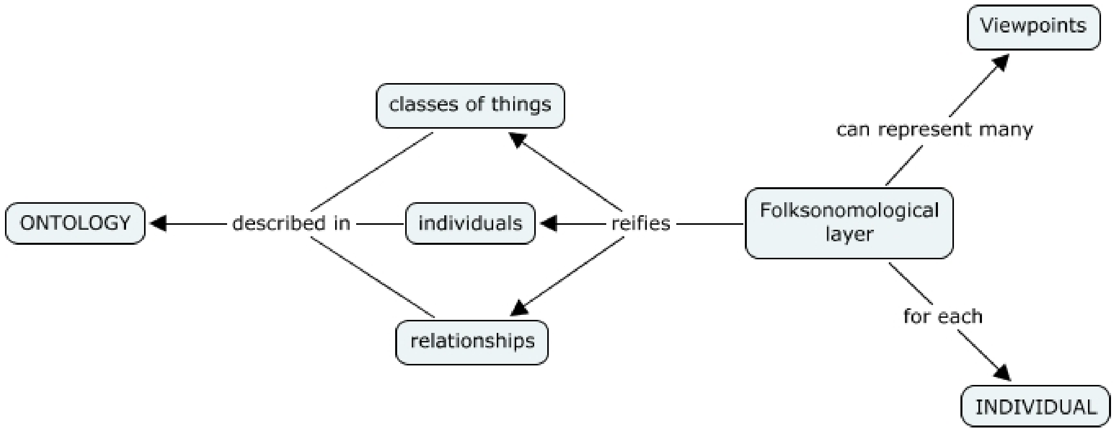


Figure 3.3 Folksonomy represented by one ontology using a reified folksonomological layer

Folksonomological Reification is still an infant technology which cannot be utilised in the case studies of this thesis, and as ordinary semantic knowledge is only accessible for humans, these technologies have to be a future endeavour.

3.5 Alternative Logic

In Chapter 2 we described how the *de facto* ontological language OWL is based on Descriptive Logics, which can be translated into first-order predicate logics, therefore something is either false or true. However, as described earlier, this is a premise which is rarely satisfied, and is hardly ever satisfied in the context of vague domains. There are other kinds of logics which work with degrees of truth.

Fuzzy logic is probably the best known of such logics. It is based on truth statements of different variables which has degree of truths. For instance some sentences could be

IF hot THEN fan faster

IF normal THEN fan stabilise

IF cold THEN fan slower

Several of these IF's could be satisfied to some degree at the same time, because it could be hot to a certain degree and also normal to a certain degree. A logical theory has been created using this kind of truth values, using different truth distributions and different operations on the statements, such as AND and OR usually being implemented using MIN and MAX functions on distribution functions [62]

An alternative kind of ontology, fuzzy ontology, is being proposed and developed based on fuzzy logics [63], [64], [65], which is a promising development in formally defining vague domains. However the problem of several viewpoints is still not fully addressed in this kind of logic, as it only represents the possibility of assigning a truth distribution to the statements within the ontology; thus if two experts disagree about a statement it cannot be fully described.

Another issue is that fuzzy logic only uses a truth distribution, whereas sometimes there is also uncertainty present in the real world. Subjective logic is another theory which introduces a set of logical operations which is based on belief / disbelief (truth) and the uncertainty of statements, which is called the opinion space [66]. Although this theory would not solve the problem of many different views in a vague domain, it would be an interesting development if an ontological theory was developed based on such a logical theory.

3.6 Neural Networks

A completely different approach to knowledge representation can be found within the Artificial Intelligence (AI) domain. This method works by emulating the functionality of brain cells in living creatures, by creating a network interconnected nodes, inspired by the way neurons are connected by synapses.

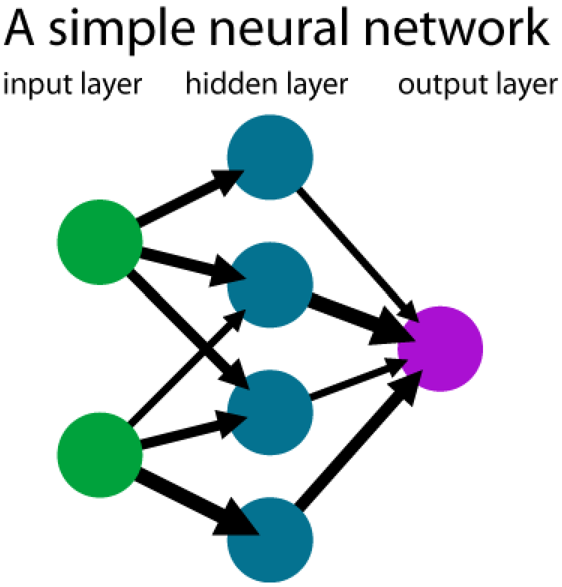


Figure 3.4 Simple neural network (source, Murx, Wikipedia)

The network is designed so that there are a number of input nodes that are connected to the hidden layer(s) of nodes, which end up in what is called the output layer. The output layer is where the "result" of interaction with the network is found. They can be used, for instance, to control decisions of another application or piece of hardware.

Before the network is of any use it must be trained. This requires a training set of inputs and expected outcomes. The inputs are applied to the input layer and the result of the output layers are found, and depending on the result and the training method used the strength of the synapses (connections) between the nodes are modified. By doing this many times the network is trained and will be able to take new inputs and predict what a proper output should be. [67]

Neural networks have traditionally been utilised in many control operations, but have also been used for selection applications such as face recognition [68] and natural language processing [69]. Good examples of natural language processing utilising neural networks are the chatbots at the AI Research (http://a-i.com/) [69]. These are based on combining machine learning techniques and stochastic language models [70], each chatbot needs to be “trained up” to work for each required area, which make them close to impossible for humans to change manually (as required by, for instance, domain changes).

Neural networks can therefore also be used to model vague domains, as at least natural language processing can be described as vague domain. For instance if the training set about semantic relationships between many different concepts was collected from many different users, then the network could be trained to predict what a given semantic relation should be between different concepts. The resulting neural network would, however, be highly dependent on the design of the training material, and would therefore be highly dependent on the designing experts. The training could obviously be continuous, but the different types of material being processed would always skew the resulting network.

3.6.1 Kohonen Networks (Self-organising maps)

An alternative way of utilising neural networks stems from the work of Kohonen [71], where a network is created usually in a 2D grid, where while under training the neurons are competing for the right to represent different kinds of data. The synapses are modified using a winner-takes-all algorithm, which strengthens the connections for the particular node which represents the given piece of data. After thousands or millions of iterations of training the network will have created a self organised map of the concepts represented by the given data set.

This map can potentially be used to find closely related concepts as these will be positioned closely together in the vector space of the network. Unfortunately the semantics of why two concepts are closely related is not always evident or useful, this is due to the fact that the maps are self-organised. For instance the author of this thesis developed an application to organise different English words, in the hope that it would be possible to cluster the words in meaningful semantics. The first iteration of this application indeed did cluster words, however the property that governed what was clustered was the number of letters in each word, a matter which could be easily mapped without the help of a neural network. Care is therefore needed when designing the data set of concepts, so that the mapping of the concepts will make sense given the needs of the application.

The biggest problem for the use of Kohonen networks in applications that make comparisons in vague domains is that the actual semantic relationship which is being mapped might be counter-productive, as in the above example, or could even be unknown. This would not help the utility of the network, and would probably make it impractical in applications.

3.6.2 Bayesian Networks

Bayesian networks [72] are a specialised form of networks where the synapses between the nodes are based on probability distributions. The distributions can be changed through machine learning by presenting a learning set. The resulting network can be used to predict the probability of a new concept matching (or not matching) the presented learning set.

For instance a set of emails which are considered to be spam can be used as the learning set for a Bayesian network. After this the network can be used to find the probability that a given new email is spam or not.

Bayesian networks could potentially be used to map semantic relationships between different concepts postulated by many different people. The network could then be used to predict whether two new concepts were semantically related. The problem with this would be, that for each kind of relationship which connected the two concepts, there would be a need for a new Bayesian network, otherwise it would merely be a ’yes’ they are related or ‘no’ they are not. For example if it is important for a user to know whether a spam email represents a phishing attack or not, a new phishing Bayesian network can be created to catch these. The old spam network would still find the phishing mail, however would not identify it as a phishing attack.

3.6.3 Issues with Neural Networks

There are several issues with the application of neural networks in the context of vague domains. Because these networks need to be trained in order to work, there are definitely issues concerned with this process. For instance who decides on what training material is used? The answer to this simple question can skew the resulting network drastically, making it potentially unworkable for other users. A concern that was not introduced in the previous introduction is the actual learning algorithms which are used when training. There are several settings in any of the above methodologies, which can change the behaviour of the resulting network.

The end result of the learning process should be a functional neural network. This statement is a problematic one, as there is virtually no way of verifying that it works or not. The only way to verify its functionality is by observation, which in many applications simply is not robust enough. Additionally if the domain which is covered by the initial training set changes then the network will not work anymore. The only thing that can be done is re-train the network, train a completely new network on the new domain presented or have ongoing training of the network, which could result in "random" spurious results.

Ultimately these techniques fail due to the multifaceted nature of vague domain. Neural networks are good at working with such domains, however they are not good at calculating results for a given person with a certain view in the network, especially if this view or domain is rapidly changing. It might be possible to train a system that identified the views of a person classified in a Kohonen network, and then use this result as an input for the input layer node of a Bayesian network producing a probable score of what this person believed about a given semantic relationship. However nobody would be able to verify the system, and the actual semantic relationship might change over time, thus creating a need for a new network.

3.7 Summary

This chapter investigated a number of alternative methodologies and technologies which either attempt at or could be used to provide automated comparisons within vague domains. These all had downfalls, which can be concluded as:

- Semantic web
    - There is a need for digital signature to verify who said what, which is not in place.
    - Content creators probably would not make OWL ontologies.
    - And if they make them they might not make them correctly, i.e. cheat to gain an advantage.
- Statistics
    - Vague domains cannot always provide enough data for statistical tools, and the data is always in danger of being skewed.
    - Ontologies are used to inference on certainties and fact (possibly based on experts opinions), whereas statistical tools infer knowledge based on uncertain information. This could potentially provide a basis for new inferences.
- Folksonomy
    - Aggregated semantic knowledge is at present only understandable by human users, because they do not contain easily computable data. Folksonomological reification techniques could be developed to provide future basis for semantic computation.

- Alternative logics
    - Logics such as fuzzy logics are used to make decisions within vague domains. These could be utilised in new ontology technologies to cater for vague domains.
- Neural networks
    - Are a completely different way of providing a data model of vague domains. This approach fails, however, because of the difficulties of training them correctly and verifying the correctness of the networks.

The next chapters will look at the two case studies and how ontologies and ontological tools can be used in novel ways within vague domains, and therefore provide a solution with the features of ontologies (extendible, understandable by humans etc.), yet provide the functionality needed to work within vague domains.

4 Development of Conversational Agents (Chatbots)

The first case study is in the area of conversational agents (chatbot). A conversational agent is an application which emulates people in chats, instant messaging boards and forums [73]. It is an application which is working within vague domains. In this chapter two traditional ways of solving the inherent natural language processing problems will be reviewed. Furthermore a solution which uses ontologies in a novel way will be presented. This will be an ontological method which can be generally utilised within vague domains.

4.1 Parsing algorithms and Artificial Intelligence

The first traditional way of solving natural language processing within chatbots is by following a traditional parsing approach where the text is parsed using a parser similar to a computer language parser added with more look-aheads, using feature enriched grammars (e.g. tense resolution) but supplemented with (or in conjunction with free-standing) statistical methods based on relevant word corpuses to resolve ambiguities. World knowledge is usually supplied in the form of predefined ontologies which are used to determine the proper responses [74]. This can be done using the Natural Language Tool Kit (NTLK) which provides developers with an API in Python [75] enabling easy creation of natural language parsers and statistical parsing tools that work with big word corpuses to establish the most likely parsing option out of several to negate ambiguities.

This approach is complicated to build and maintain, because the pure parsing processes involved in natural language processing are much more demanding than, for instance, those in computer language parsing. Also the lack of word corpuses that reflect the expected natural language utterances appropriately will increase the error rate of the ambiguity resolution performed by the statistical methods. This makes the resulting product quite prone to conversational errors and misunderstandings. Chat is notoriously known for very short sentences with abbreviations and specialised lingo, and it was difficult (if not impossible within time and resource constraints) to acquire appropriate grammatically specified corpuses when this solution was reviewed.

On the other hand, as the world knowledge is represented by using ontologies, it is relatively simple to change the world knowledge of the chatbot without having to change any other parts of it, as long as the domain knowledge of the knowledgebase follows the ontological commitment specified.

4.2 Chat structures

Another common approach is to explore the Zipf curve distribution of words and sentences in natural language, where it is realised that the way natural language is used is not randomly distributed but rather having a small number of common responses to each utterance [76]. The chatbot developer then tries to cover as many potential responses by the user to each chatbot output as possible, by building up chat trees using a descriptive chatbot language such as the Artificial Intelligence Meta Language (AIML) [1]. This is a trial and error approach where the developer constantly monitors the chat logs to see if new chat patterns have been discovered that needs to be added to the chat trees. It is comparatively easy to create conversational trees and results can readily be tested with users to enhance the capabilities of the chatbot. However the world and domain knowledge are incorporated into the chat trees to ease the developmental phase for non-technical developers and avoid ambiguities. The result of this, depending on the size of the conversational tree, is that it is either impossible or at least incredibly difficult to transform the chatbot into another functionality or world domain. For example, if a chatbot has been developed for a FAQ (Frequently Asked Questions) section of a specific webpage, it would be difficult to transform this chatbot into another FAQ chatbot for another site without creating errors; it would normally be just as easy to start from scratch with a new chatbot.

There are many examples of this type of chatbot. For instance Weizenbaum‘s Eliza the classic conversational agent psychiatrist, which utilised the basic structure of psychiatrist sessions [77]. A.L.I.C.E [2], which has won the Turing test competition, the Loebner Prize [78], for several years is another classic example. Pandorabots is a web service dedicated to this type of chatbot, made using AIML [3].

4.3 The Ontology Driven Conversational Agent (Chatbot)

The first case study which forms the basis for the novel approaches explored in this thesis is an ontology driven conversation agent system. The theory behind the conversational interface system is to use the Zipf curve distribution to build chat structures and sentences without any world knowledge (chat patterns), but integrating ontological queries into the structures retrieving the world knowledge on demand when the responses are created. Thus the method benefits from both classical approaches and avoids their pitfalls.

This functionality has been achieved by developing a bespoke ontology query language OwlLang which is used within the interpreted AIML language. Figure 4.1 is a graphical representation of the complete system. At the heart of the system is the OwlLang scripting language, which is used to acquire the ontological knowledgebases to retrieve the domain knowledge, or in other words to fill in the gaps of the static chat patterns. The Language has been developed in Java using the compiler tool JavaCC [79] and the ontology tool Jena [26]. The language itself will be described more in detail in the following section.

The chat patterns are described using AIML and are interpreted by a standard AIML interpreter ProgramD [80]. This application has been slightly modified to allow incorporation of the OwlLang scripts into AIML and call the OwlLang interpreter to perform the retrieval of information from the ontological knowledgebases. The only deviation from the standard AIML language in the static structures has been the inclusion of a grammar check performed from within OwlLang instead of using the usual singular and plural resolution to decide whether an “is” or “are” should be inserted into the response. This is needed because the AIML tags would normally have the actual known value of the subject of the sentence when run in ProgramD, but this value will not be known by ProgramD when using OwlLang as it is obtained at runtime. This grammar check was implemented using Link grammar 4.1b [81] which is an open-source grammar checker that is often used in applications using the parsing approach for natural language processing.

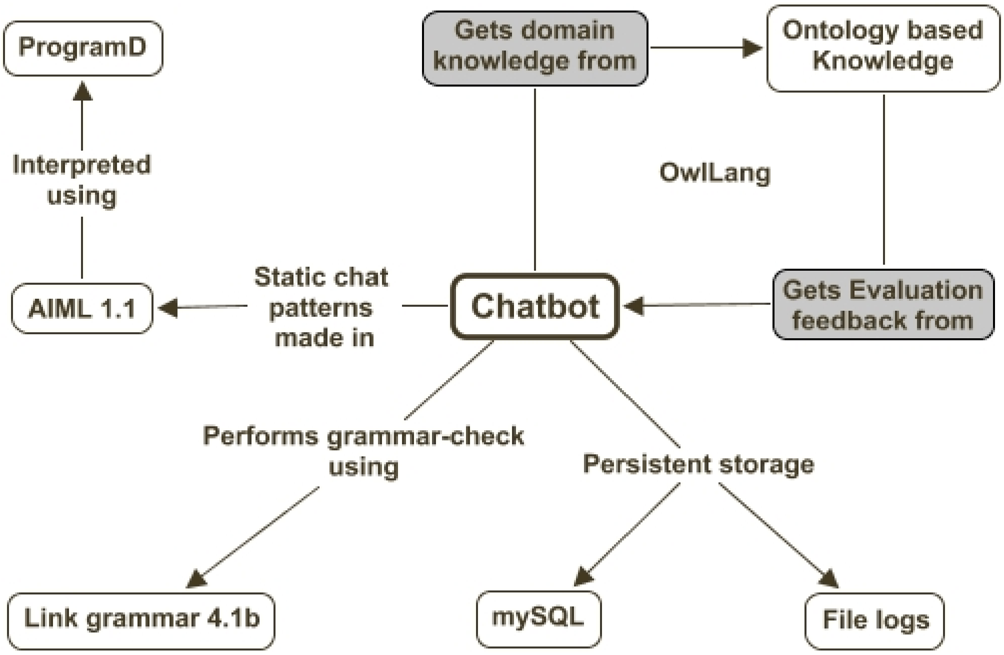
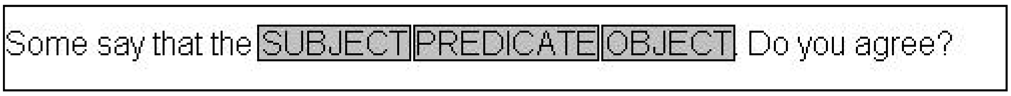


Figure 4.1 The Complete Conversational System

Persistent storage of ProgramD internal data structures is done in a mySQL database (mySQL 2006). Chats, ontology and the complete knowledgebases are stored in file logs, enabling easier readability for the chatbot developer; in future, however, this should be stored in the mySQL database for easier maintenance.



Some say that the SUBJECT PREDICATE OBJECT. Do you agree?

Figure 4.2 Chat structure with integrated ontology query.

Figure 4.2 is an actual example taken from the chat structure of the conversational interface of a surveying chatbot. It asks questions of people and inspects the answers to find meaning. Many different questions can be asked dependent on the values of subject, predicate and object. However the expected responses generally will not depend on the different values, and where they do depend on them the words are usually part of the response, allowing the chatbot systems to infer the meaning. If no meaning can be inferred the system has a series of escape questions which will assist in finding the opinion of the surveyed person.

The complete surveying chatbot system has 100 different question of a similar structure in its chat structure tree all of these have appropriate response trees. The Zipf curve has also been applied on the response trees as most responses to questions follow the same basic pattern. The specific responses are then interpreted by the system and matched with ten different states with different responses and actions. The states are listed below:

- Yes
- Definite no
- No, but verification needed whether the user means
    - Definite no
    - Unknown
- Unknown
- Unsure
- Unsure, but a certain (maybe unknown) value is given
- Evidence of a valid question
- No understandable answer given

When the user response has been fully explored with follow up questions a new question will be asked, or the evaluation will end depending on the number of questions that is required by the system to complete a full survey. Following is an actual example of a complete cycle of one question taken from an evaluation session of the on-line Blackboard system at The University of Reading [82]. The user initially answers with a ‘no’ that needs verification resulting in a follow up question that is answered by a ‘definite no’:

chatbot>

When asked, some people claim that the discussion board in programming is easy to navigate. Do you support that view?

user>

no

chatbot>

Do you say that because you have not got an opinion?

user>

i have got an opinion, but frankly, it's not a positive one

chatbot>

Really!

4.3.1 OwlLang

The main functionality of OwlLang is to retrieve a string from the ontological knowledgebases depending on specified operands and logics; if several strings match the specifications then a random string from that list of strings will be retrieved. The following example is not intended to be a comprehensive instruction guide for OwlLang, but rather an introduction to the capabilities of the language.

There exist three main top-level string retrieval commands, which are;

- ONTIND – gets the string from a known individual within the ontology.
- ONTPROP – gets the string from a known property within the ontology.
- ONTCLASS – gets the string from a known class of the ontology.

The default string that will be returned is the URI (Universal Resource Identifier) of the found entity; however alternative strings can be specified following the command. So for instance

ONTIND(something) :[rdfs:label]@'en'

would return the rdfs:label of an individual that satisfies '*something*' from within the ontology.

There are three main commands that can be used within the brackets of the top-level string retrieval commands which determine what kinds of sets are used in the selection, they are;

- ind – individuals
- prop - properties
- cls – classes

So for instance if ‘*cls*’ is used with ONTIND the string will come from one of the individuals of the matching classes, on the other hand if ‘*ind*’ is used within ONTCLS the string will come from one of the classes in which the matching individuals are defined.

The matching operators are equal (==) and not equal (!=) and different commands can be logically joined together with AND, OR and XOR. Again here the default string to match with is the URI, which can be changed by following the command with the URI of the predicate that points to the string that should be used in the match instead, e.g. [rdsf:label] (‘[‘ and ‘]’ are used instead of the standard URI characters ‘<’ and ‘>’ as these characters are not allowed inside XML elements.)

The following is an example of a query to find an individual within the knowledgebase, which is a claim made by somebody other than the actual user. Claims are reifications of statements made within the ontology, so the subject, predicate and object part of that statement is also returned to AIML:

```
$OWLLANG(
ONTIND(
        cls==[seequel:Claim] AND
        prop==[seequel:madeBy]>>!=[:<get name="name"/>])
ONTIND(ind==#1):[seequel:subject]
ONTIND(ind==#1):[seequel:predicate]
ONTIND(ind==#1):[seequel:object]
ONTIND(ind==#2):[rdfs:label]@'en'
ONTPRO(prop==#3):[rdfs:label]@'en'
ONTIND(ind==#4):[rdfs:label]@'en'
) THAT='INDPROIND'
```

The results of the individual commands are added to a queue, which can be referenced by following commands using the ‘#’ character.

The ‘THAT’ part is connected to the AIML language. The way that OwlLang has been integrated into AIML means that results from OwlLang are treated as an utterance from the user, just with the difference that the OwlLang can change the ‘THAT’ part of the matching stack of AIML. The ‘THAT’ part is the last sentence that AIML sent to the user. The result is that the following piece of AIML code will match and process the previous OwlLang command (if it has been properly included in an AIML <category> itself.):

```
<category>
<pattern>
OWLLANG ONTIND * ONTIND * ONTIND * ONTIND * ONTIND * ONTPRO *
ONTIND *
</pattern>
    <that>
        INDPROIND
    </that>
    <template>
        <think>
            <set name="subject"><star index="5"/></set>
            <set name="predicate"><star index="6"/></set>
            <set name="object"><star index="7"/></set>
            <set name="subjectURI"><star index="2"/></set>
            <set name="predicateURI"><star index="3"/></set>
            <set name="objectURI"><star index="4"/></set>
            <srai>CHECKPREDICATEGRAMMAR</srai>
        </think>
        <random>
            <li>
                <think>
                    <set name="topic">QUESTION
                    DOYOU</set>
```

```
                        </think>
                        <srai>ASKSOMEBELIEVES</srai>
                  </li>
                  <li>
                        <think>
                              <set name="topic"> QUESTION
                              WOULDYOUAGREE</set>
                        </think>
                        <srai>WOULDYOUAGREE</srai>
                  </li>
            </random>
      </template>
</category>
```

The string in the <pattern> element is the pattern being matched, i.e. the response from OwlLang; hence it can be observed that OwlLang returns all results in one long string where each sub-result string follows its appropriate command name. The ‘*’ is a wildcard character in which the value can be accessed using the AIML command <star> (with an index specifying which number ‘*’ is referenced). This design enables flexibility within the AIML language where the OwlLang code has full access to all aspects of AIML, and the only non-AIML behaviour is the change of the ‘THAT’ part in the matching stack. This was allowed for, because this design would allow looping behaviour previously not part of AIML.

One of the more peculiar design decisions made initially was to stay away from an XML based approach even though AIML is using such an approach. The reason for this is that by doing so the language would be more compact, easier to read, and it would allow the interpretation of OwlLang to be performed after the AIML code had been processed but still integrated into the response of AIML, hence allowing the use of all the AIML tags within the code of OwlLang. If OwlLang had been in XML, AIML would simply skip other AIML code within the OwlLang tags. To exemplify with a piece of actual code, where the URI of the inverse predicate of the predicate being used by AIML presently is being found and returned.

```
ONTIND(
cls==[seequel:Claim] AND
prop==[seequel:madeBy]>>!=[:<get name="name"/>])
```

The AIML code <get name=”name”> would not have been processed had the ONTIND been a XML element, resulting in a situation with very limited inter-functionality between AIML and OwlLang.

Another design issue which is especially important for more experienced semantic web developers is that the knowledgebase and ontologies are considered to be a closed world rather than the usual open world assumption. Open world means that if something cannot be inferred true then it does not mean that it is false, it can still be true [15]. This assumption is an intelligent assumption when looking at the World Wide Web, but when retrieving strings contained within an ontology and an ontological knowledgebase then it simply is more effective to allow a closed world similar to what would be expected in an ordinary object oriented programming environment. This is the main reason for not using a standard semantic web query language such as SPARQL [20].

Beside string retrieval the language also has some other functionality for convenience, these are:

- Regular expression string comparison.
- Storage of knowledge into the ontological knowledgebase.
- Boolean check whether a given URI is a resource known in the knowledgebase or not.
- Unique URI creation based on already known URI's in the knowledgebase.
- Grammar check.

All of the commands within OwlLang facilitate the creation of ontology driven conversational interfaces. As described, the most important feature of OwlLang is that it provides an easy way to query ontologies and find strings from the knowledgebase contained within them. Any class, predicate and individual contained in the ontologies can be found and used in this process. The responses from the 'chatters' can furthermore be stored within the ontologies allowing a growing base of knowledge. The other functionalities are made to support these two functionalities.

4.3.2 The ontologies

The surveying system uses three different ontologies as described in figure 4.3; one that describes quality in eLearning, one that describes teaching products/procedure and lastly one, also called a knowledgebase, which states what claims and facts are. By using these ontologies the eLearning product/procedure that is to be evaluated can be described and questions can be developed. Then new claims and facts can be made about it from the responses of the surveyed people. The actual answers to these questions are added to the knowledgebase as claims, which extend the pool of information that the questions are taken from.

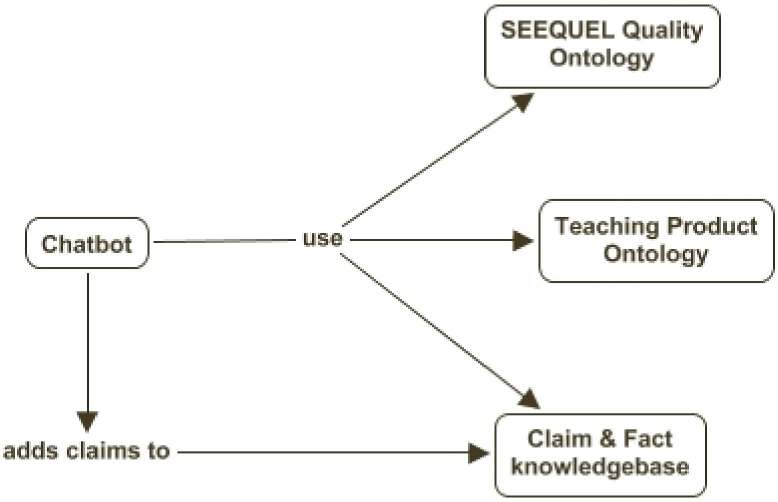


Figure 4.3 Ontologies used by the System

The SEEQUEL Quality ontology is an ontology based on the internal findings of the SEEQUEL project [83]. In this project a framework for quality in eLearning was built, the ontology is based on an internal SEEQUEL ontology, which was created to better understand the framework. For what is developed here it has been transformed using the web ontology language OWL [15] in order to allow the use of Jena to manipulate it programmatically. The ontology describes different quality descriptors and how they interrelate. The following are some of the descriptors:

- Interesting
- Fun
- Boring

- Too long
- Off topic
- Engaging.

These descriptors form the basis of the chatbot's vocabulary when it needs to use quality terms, however if other terms are needed in other evaluation tasks the open nature of ontologies would allow the user to dynamically add the appropriate terms to the ontology. The following code is an example of one of the inter-relational facts which are in the ontology.

```
:boring          :impliesNot       :interesting .
:fact103         a                 :Fact;
                 :madeBy           :seequel;
                 :object           :interesting;
                 :predicate        :impliesNot;
                 :subject          :boring .
```

These OWL lines describe that if something is boring then it is implied that it is not interesting. It is also reified as a "fact" in the ontology made by SEEQUEL. This "fact" can then be used by the chatbot to infer additional questions to ask the users.

The Teaching Product Ontology (figure 4.3) is a simple ontology, which enables developers to describe any teaching product enabling them to establish claims about it using the SEEQUEL Quality Ontology. This ontology is based on three core entities; teaching product, units of a teaching product and functionality of teaching products.

The ontology defines the two inverse properties “is” and “is not” and the property “no relation”, which can be used to describe a defined entity using a quality descriptor from the SEEQUEL Ontology. Using these ontologies the chatbot developer can then define the entities which make up the actual eLearning product, which needs to be evaluated. After this he can add the claims which form the basis of the claims knowledgebase. The knowledgebase is an ontology containing individual claims and facts. It stores all the claims and facts within the surveying system and these are used to find the domain knowledge when the conversational interface creates another question. The following code shows one such claim, here it is claimed by a user named “Karsten” that the functionally called “discussion boards on Blackboard” is “easy to use”.

```
:bbDB              :is                seequel:easyToUse .

:claim1            a                  seequel:Claim ;
                   :object            seequel:easyToUse ;
                   :predicate         :is ;
                   :subject           :bbDB ;
                   :madeBy            seequel:Karsten .

:bbDB              a                  :DiscussionBoard ;
                   rdfs:label         "discussion boards on Blackboard"@en .

:DiscussionBoard   rdfs:subClassOf    :Functionality ;
                   rdfs:label         "discussion board"@en .
```

This illustrates how easy and fast it is to add additional knowledge to the complete system allowing it to ask new evaluative questions to the users. This allows the conversational agent to be transformable to other systems. Similarly it is feasible to change the quality terms used in the chatbot system by removing the SEEQUEL Quality Ontology and adding another quality ontology with different quality terms.

4.3.3 Results of Using the Chatbot in Surveying eLearning

The logic behind utilising chatbots for carrying out surveys is that it is normally extremely difficult to get user responses to traditional questionnaire-based surveys, unless the test people are forced in some way to complete the survey, which has a tendency to distort the results. For instance a survey carried out by CEDEFOP in the European Training Village produced a less than 1.5% response rate [84]. It was also anticipated that the responses would produce more varied and informative responses with added value compared to tradition rating of topics on a scale from 1 – 5.

It was decided not to make participation in the survey compulsory for the students, which is the traditional approach in evaluation surveying at the University, so in order for the students to participate the chatbot was advertised in one of the lectures, and a link to the chatbot was added from within the programming module on Blackboard. The chatbot had 15 students out of 203 registered students on the course (7.3% response rate) chatting, this is lower than anticipated, even though it was remarkably higher than the CEDEFOP participation percentage.

It is no surprise that the ratio was higher than the CEDEFOP survey, as the author had closer relationship to the target audience through the supervisor, and therefore had a direct contact with the students, however the reason for expecting an even higher involvement ratio was that the students hopefully would see the use of the chatbot as an opportunity to influence the usage of Blackboard. Unfortunately the traditional evaluation period at the end of term is a time where students are focussing on finishing the term and starting their holidays. For that reason the chatbot had a more difficult task of attracting attention than it would have had if introduced to the students in the middle of term. Another was that the chatbot was not directly accessible from outside the university firewall, so the students had to access through a terminal server service provided by the school, and seamless integration into Blackboard was impossible to achieve for the pilot.

When examining the quality of the feedback rather than the quantity, the best way to illustrate the chatbot working is by showing a complete example:

chatbot*>*

Hello!

I will be asking you a series of questions about the use of Blackboard in the Programming course at Reading University. It is important for me that you answer the questions as sincerely as you can, as we will be using the results to improve the usage of the product at the university.

This will also help my creator in his research of chat, which in turn will help with my ability to talk to people.

Well enough chit chat, here we go!

Would you agree with this statement, the content of the discussion boards in programming are off topic?

user>

No

chatbot >

Did you answer like that because you have not made up you mind?

user>

No

chatbot>

Really! Would you agree that the discussion board on Blackboard is not easy to navigate?

user>

A little bit

chatbot >

I didn't get that, could you say that in another way, please?

Would you agree that the discussion board on Blackboard is not easy to navigate?

user>

It's not very easy

chatbot >

I see! The discussion board on Blackboard is not easy to navigate. I'll remember that.

When asked, some people claim that the content of the assignments in programming is not boring. Do you support that sentence?

user>

Mostly

chatbot >

The content of the assignments in programming is not boring. Is that a true statement more often than not?

user>

yes

chatbot >

So you agree. Would you agree with this statement, the content of the assignments in programming is challenging?

user>

It is challenging enough

chatbot >

It is challenging enough, so what would you say to this? Would you say that the content of the assignments in programming is challenging more often than not?

user>

yes

chatbot >

I'll remember that.

This is all for this time. Thanks for your help in our research. Have a nice day.

user>

and you

When analysing all the chat logs there is one common trend throughout; the user takes a few introductory statements to reply with anything other than yes or no, just like in the previous example, and for that reason the chats probably ought to have been made longer than in pilot. The cause of this is probably that the users need time to familiarise themselves with the interface of the chatbot, however when the student starts to communicate more with the chatbot, more values are being retrieved out of the evaluation process than is normally the case. For instance in the above example where the user is asked if she agrees that the content of the assignments is challenging, the response is that they are challenging enough. Here the chatbot wants to clarify whether that means they are challenging more often than not, which they are. This kind of communication shows the human evaluator that not only does this user believe they are challenging, which a multiple choice question would elucidate too, but also that they are challenging the user to the limit, as this user clearly is stretched in the assignments. During a normal evaluation process this kind of knowledge would be difficult to determine, as a grading system would only show that the user finds that they were difficult with some score.

It could be argued that the chatbot just works as an automated interviewer, and a human evaluator still has to read through all of the logs, which is true, however not entirely, because the chatbot automatically stores what it infers from the communication using the knowledge in the ontologies. What the human evaluator needs to do is extract the gist of the value-added information from within the chat, as this is information which would be extremely difficult to automate.

An important issue when examining if this is a viable evaluation method is to see if what the chatbot infers is correct. There are, in principle, two different mistakes that can be made by the chatbot; firstly it can infer something wrongly from the chat and secondly it could overlook something within the chat, hence not inferring an intended meaning. Clearly of the two different mistakes the first is the one with the severest consequences as the statistics resulting from the chatbot would be wrong, whereas the second would leave out information which possibly could be rectified by the human evaluator. In our pilot a total of 60 questions were asked, out of these 58 were understood correctly leaving a 96.7% success rate. Both of the incorrectly inferred answers were due to misunderstood communication between the chatbot and the human (the user did not understand the chat correctly), and they were both stored with a wrong interpretation. Although 96.7% is a very high success rate it would be valuable to look at the two misinterpreted answers to see if or how it would possible to change the system, so that it would infer them correctly.

The first misinterpreted answer is a situation where the user clearly (when analysing it as a human) does not have an opinion. The chatbot tries to elaborate by asking further questions and in the end the user answers by saying how the user believes it should be. This is an excellent example of the value added information that comes through when using the evaluation chatbot; however an answer indicating how it should be, inferred to 'how it is', does skew the results:

chatbot>

It has been observed that the content of the assignments in programming is challenging. Do you support that view?

user>

depends on the particular assinment

chatbot>

In your opinion would it be true that the content of the assignments in programming is challenging more often than not?

user>

hard to say

chatbot>

My mind was in another place please say it in another way? In your opinion would it be true that the content of the assignments in programming is challenging more often than not?

user>

I suppose i would say yes because it should be challenging

chatbot>

Ok!

This kind of misinterpretation is probably difficult to rectify, as the value added information provided by the user clearly states that she believes the assignments ought to be (challenging). However she is causing confusion to the inference process, because she does not know whether they are challenging or not. She could have answered that she did not know, and that would have been inferred correctly, however possibly her answer now provides us with more information, even though the statistical result is skewed, as we now know that this student wants challenging assignments, and that they sometimes are that but not always. This should be taken into account when designing the assignments delivered in Blackboard.

The other example where the answer is misinterpreted is a situation where the user apparently has misread the question, answering with an agreement although stating that he disagrees:

chatbot>

Let's try a new question. It has been observed that the content of the discussion boards in programming are not off topic. Do you support that sentence?

user>

No they are on topic.

chatbot>

The content of the discussion boards in programming are off topic! I'll remember that.

Ambiguities in natural language are probably one of the most serious and difficult problems within natural language processing [74] and it certainly does not help when the human response is based on a human error. Because the human seemingly disagrees with the statement (starting with a no), the chatbot infers that he disagrees, without checking that the user is actually disagreeing with the opposite of he what was asked (the human error). What can be learned though from this example is that the structure of the sentences could be kept simpler than they are. For instance having double negatives is not a good practice, as they might add to the confusion of the users, so by changing “off topic” to “on topic” in the knowledge base would probably enhance the usability and inference capabilities of the chatbot. The results in this section are reused from [82], which describe the original work of the author.

4.3.4 Changing the Ontology

To test the versatility of the ontological approach, another ontology was designed, which described generic quality of webpages. This new ontology was easily integrated into the old chat tree structure (it took only 3-4 hours), and was then used to evaluate the website of EFQUEL [85]. The chatbot was functional for a period of a month in 2006.

The following is an extract of the webpage ontology:

```
# Webpage, structures of webpages and functionality

:Webpage     a               owl:Class ;
             rdfs:label      "web page"@en ;
             rdfs:comment    "A web page is an Internet site with an
                             associated URL"@en .
:Structure   a               owl:Class ;
             rdfs:label      "structure on a web page"@en ;
             rdfs:comment    "A structure which is part of an web page"@en .


:partOf      a               owl:ObjectProperty ;
             rdfs:domain     :Structure ;
             rdfs:range      :Webpage ;
             rdfs:label      "is part of"@en ;
             rdfs:comment    "The relation that a structure is
                             part of a web page."@en .
```

```
:WebFunctionality    a                 owl:Class ;
                     rdfs:label        "web functionality"@en ;
                     rdfs:comment      "A functionality of a web page
or web structure."@en .
:functionalityOf     a                 owl:ObjectProperty ;
                     rdfs:domain       :Relation ;
                     rdfs:range        [ owl:unionOf ( :Structure :Webpage ) ] ;
                     rdfs:label        "functionality of"@en ;
                     rdfs:comment      "A functionality of a web page
                                       or structure of a webpage."@en .
```

By using this simple ontology of webpages the actual EFQUEL website was the described:

```
# EFQUEL page structures
efq:home             a                 :Webpage ;
                     rdfs:label        "Homepage of EFQUEL"@en ;
                     rdfs:comment      "The front page of
                                       EFQUELs web presence."@en ;
                     dc:identifier     "" .


efq:eLQual           a                 :Structure ;
                     rdfs:label        "eLearning Quality Tab"@en ;
                     rdfs:comment      "The eLearning Quality tab of
```

```
                                        EFQUELs web presence."@en ;
                    dc:identifier       "" .

efq:news            a                   :Structure ;
                    rdfs:label          "News and Events Section"@en ;
                    rdfs:comment        "The new and events tab of
                                        EFQUELs web presence."@en ;
                    dc:identifier       "" .

efq:community       a                   :Structure ;
                    rdfs:label          "EFQUEL Community"@en ;
                    rdfs:comment        "The Community tab of
                                        EFQUELs web presence."@en ;
                    dc:identifier       "" .

efq:about           a                   :Structure ;
                    rdfs:label          "About Section"@en ;
                    rdfs:comment        "The eLearning Quality tab
                                        of EFQUELs web presence."@en ;
                    dc:identifier       "" .
```

```
efq:myEFQ         a                  :Structure ;
                  rdfs:label         "My EFQUEL Section"@en ;
                  rdfs:comment       "The my EFQUEL tab of EFQUELs web
presence."@en ;
                  dc:identifier      "" .
efq:wiki          a                  :Webpage ;
                  rdfs:label         "EFQUEL WIKI Page"@en ;
                  rdfs:comment       "The Quality WIKIpedia
                                     run by EFQUEL"@en ;
                  dc:identifier      "http://www.qualityfoundation.org/
                                     quality-wiki/index.php/Main_Page" .


efq:layout        a                  :WebFunctionality ;
                  rdfs:label         "layout of the pages"@en ;
                  rdfs:comment       "The layout of all of the pages contained
                                     in EFQUELs web presence"@en .
```

Like the eLearning Quality survey chatbot some initial claims were made seeding the chatbot, and the chatbot was finalised.

The actual results of this chatbot, were unfortunately quite disappointing. Only four users decided to try the chatbot, and an additional user tried to crash the chatbot by looping chats back into the chatbot system, creating an infinite loop. The lesson from this is that surveys probably are not suited to completely open environments such as a webpage, where people are not expecting to be surveyed.

More importantly, the exercise showed that the approach allows easy exchange of domain knowledge through the use of the domain ontologies.

4.4 Reification Statements Used to Overcome the Problem of Vague Domains

The problem of the vague domains in this chatbot environment is inherent since several people will have differing opinions about the domain that is being surveyed. Therefore the knowledge ontology is going to have opposing statements about the domain. For instance, some users of Blackboard found the forums usable, and others had a completely different view on them. Both of these statements would then become part of underlying ontology. This created problems with the inference engine resulting in inferences that would both indicate usability and the opposite, thus always resulting in a positive answer when queried.

This problem would obviously become even more evident if a chatbot was created which would grow the ontology. For instance if an expert system domain developer chatbot was created, the chatbot would be asking experts about their domain, enabling creation of classes and properties within the domain. If two experts then differed in opinion about a certain class or property then the ontology would become unusable from an inferencing viewpoint as the inference engine would not know which statement would be correct.

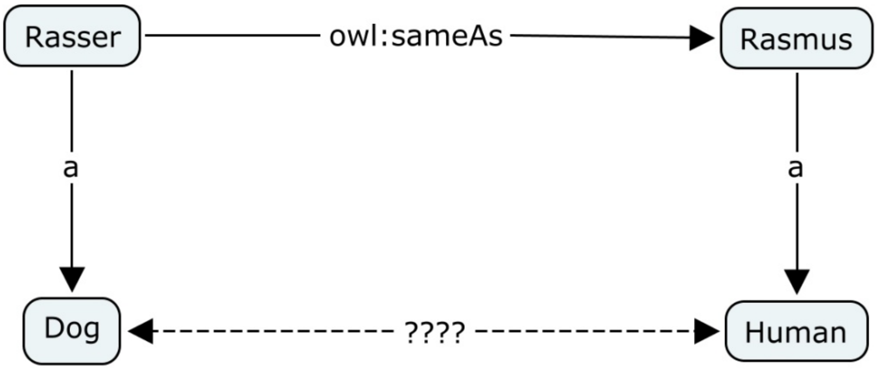


Figure 4.4 Problem of Standard Ontology

Figure 4.4 to figure 4.6 is a simplified example of a vague domain where there are two disagreeing “experts”. The experts would have, in a standard discrete logic based ontology, to agree on all the statements within the ontology. In figure 4.4 diverging semantic statements have been put into a standard ontology. This results in a semantic problem, because “Rasser” is declared as being the same entity as “Rasmus” then because “Rasser” is a “Dog” and “Rasmus” is a “Human” it can be inferred that “Dog” is either the same class as, a sub-class of or a super-class of “Human”.

:Rasmus	*rdf:type*	*:Human .*
:Rasser	*rdf:type*	*:Dog .*
:Rasser	*owl:sameAs*	*:Rasmus .*

Inferred

:Rasmus	*rdf:type*	*:Dog .*
:Rasser	*rdf:type*	*:Human .*

The meaning of Dog could obviously be a promiscuous person rather than the animal, and then there evidently would not be a problem, however in this example “Dog” is meant to be the four legged animal. The above inference would clearly only be a problem for the integrity of the ontology (neither Rasser nor Rasmus can both be a dog and a human) if there where statements somewhere in the ontology declaring that humans and dogs are not related, such as a simple owl:disjointWith declaration or an elaborated class hierarchy of biology. With no such declarations existing, an "open world" inference engine would infer that they shared the same class extensions, thus having some relationship between "human" and "Dog", as both "Rasmus" and "Rasser" would both be "Human" and "Dog".

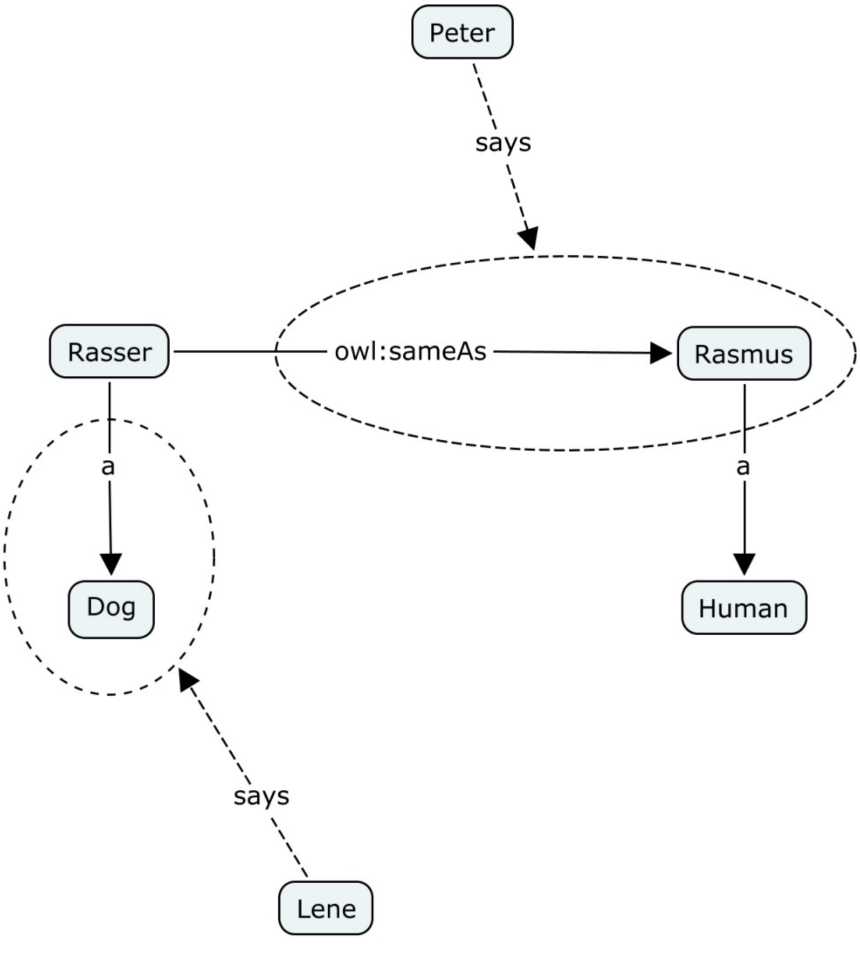


Figure 4.5 Two "Experts"

As stated earlier the whole problem stems from the fact that two experts are disagreeing about the facts of the domain. It is evident that the two experts Lene and Peter in figure 4.5 probably would be able to come to an agreement that "Rasser" is two different entities, one a "Dog" and the other a "Person" also known as "Rasmus", and then declare two different entities for "Rasser", however the situation is undoubtedly not always as easy as that. What if the ontology is distributed, for instance as described with the help of a chatbot, and the experts never meet? Or what if some part of the domain simply is in conflict? Finally, what if the system is going to be used as a data model for a folksonomy, where all the users are experts in their own right, and they never communicate while creating the content?

4.4.1 The Solution

The method developed in the chatbot system is to utilise reifications of statements. A reification is at its core a statement about a statement. When creating a reified statement using OWL a token (e.g. a URI) is created, which points conclusively to the statement which needs to be reified. This token is then used as a subject in other statements to declare new statements. For instance the following declares that there is a statement that Rasmus is a human, and that this statement was made by Peter:

```
:Rasmus        rdf:type         :Human .
:Reif          rdf:type         rdf:Statement ;
               rdf:subject      :Rasmus ;
               rdf:predicate    rdf:type ;
               rdf:object       :Human .
:Reif          :madeBy          :Peter .
```

This is all declared in standard RDF and OWL, as reifications are part of standard RDF, thus enabling the use of the standard Semantic Web toolsets such as Jena. Figure 4.6 represents graphically the simple example using reified statements.

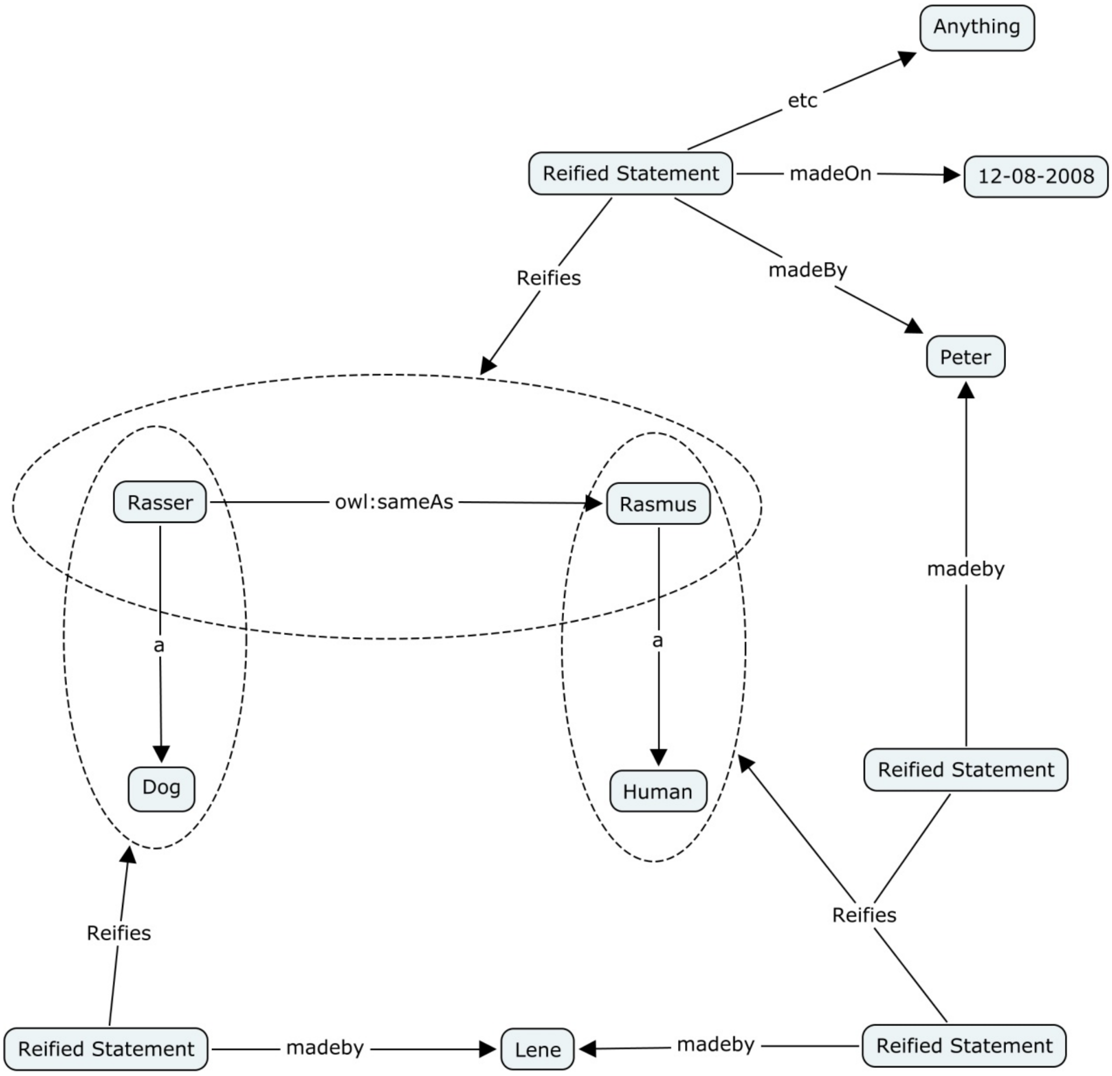


Figure 4.6 Reified Ontology

In figure 4.6 both Lene and Peter are have been able to make completely different statements about Rasser and Rasmus. Peter believes Rasser and Rasmus are the same Person, whereas Lene believes that Rasser is a dog. This illustrates two different viewpoints of the same domain (the very simple domain of Rasser and Rasmus).

This solution could be compared to the digital signatures which will become part of the Semantic Web toolset, however it is not the same methodology. Digital signatures are supposed to be used at the level of different ontologies, thus enabling the users of different ontologies to identify and verify who has made the complete ontology. Therefore this proposed solution allows a more granular functionality in describing “who declared what”, as each single statement effectively could be “signed”, thus provide a much more granular expressivity.

4.4.2 Inference

At the core, inferencing within a reified ontology is similar. They are both underpinned by the logics provided through OWL. The difference of a reified ontology inference capabilities, are that a few new inferencing opportunities are available:

- Viewpoints can be explored.

  The different viewpoints provided within the reified statements can be explored by all statements made by a single user (i.e. by using SparQL) and further inferred by comparing/contrasting (clustering) different user viewpoints. These clusters of viewpoints could then be utilised to make predictions, based on what is known about the clusters, of other similar kinds of views.

  - Statistical methods can be utilised to compare and contrast different viewpoints.
  - Fuzzy methods can potentially be used to represent aggregated points of view.
- Disagreements between users can be explored.
  - The actual difference of view can by specified as the disagreeing statements can be found within the same domain ontologies.

    - They can be utilised to compare and contrast different clusters of disagreement. Similarly they can be used to identify patterns of similarities between differences, possibly with the opportunity to aid serendipity.
- History of ontology development.
    - As the ontology grows, rather than being created, the "history" of the ontology can be used to make new types of inferences. Trends from the past could be identified easily and these might be possible to utilise to make predictions of future ontology changes, both general changes and changes from individual viewpoints.

The method used in the chatbot system to enable inferencing is to utilise the above methodology of reifications. The user base of the survey chatbot has, however, been too small to provide meaningful inferences between clusters of views, other than those which normal statistical tools would have been able to find. However the different viewpoints are contained in the reified layer of the ontology, and it should become possible to make inferences on a personal level. Statements like "People like you normally believe that..." and "Others who have said this are also saying..." are then possible to make. Unfortunately normal inference engines (such as Pellet and FaCT++) are not developed to work in this kind of logic environment. They only use the logic provided through OWL, but they can be used to some extent by first making SPARQL queries that extract the different viewpoint and then run the inference engine on the results. However it would be beneficial if tools could compare and contrast different viewpoints without this intermediate step.

4.5 Summary

This chapter started with introducing two different methods of developing conversational agents. The main emphasis of the chapter described a solution, where an ontological scripting language was introduced to allow full access to knowledgebases contained within ontologies and adding knowledge based on user interaction, thus allowing a growing knowledge base. By utilising ontologies, quick changes of knowledge between domains are now possible and this is not possible with the other investigated methods.

The solution provided a novel methodology of storing diverging knowledge within the same ontology by utilising reification statements, this has enabled a new layer of knowledge within the knowledge base, which can be utilised when inferencing between different points of view.

A generic survey chatbot was developed using these technologies. Two different sets of ontologies were created for the chatbot enabling it to survey two different scenarios, surveying both students about their views of the virtual learning environment Blackboard, and users of a website.

The next two chapters will be investigating the second case study domain of competencies and competency frameworks.

5 Competency, Competency Frameworks and Representation

In this chapter the domain of competency and competency frameworks will be introduced, which will form the basis for the second case study in chapter 6. Competency has grown into an important concept in many domains, especially in categorisation of education [86], [87] and human resource [88], [89], [90]. The European Union and most of the member states are also showing their interest in competencies especially by investing heavily in the development of competency frameworks and qualification frameworks. Traditionally these have been paper based documents or standalone documents on web pages, however it has been a perceived need [88] to electronically manipulate and share both competencies and competency frameworks, hence the need for electronic representation is increasing.

5.1 Definition of Competency

The term competency has been the root of much debate and confusion; this is probably due to the "artificial" nature of the concept of competency. The concept has even been called a "fuzzy concept" by Boon et al, and recognised as a ″useful term, bridging the gap between education and job requirements." [91] It has been created by people to represent something that is not evident in the world, and it is therefore a reification of some aspect or attribute of humans or agents, thus there is no easy way of defining the term. There is even confusion about the difference or similarity of the terms competence and competency. Mostly they are used as synonyms, but some researchers and competency practitioners apply subtle differences between the two words. Take for example the Columbia Guide to Standard American English; "Competence means both "a sufficient amount to live on, to meet one's needs" and "having legal or practical ability to perform.” Competency means the same things but is less frequently used, except in educational argot, where competencies are the various skills pupils are to be taught and teachers are to be prepared to teach. The plural competences occur infrequently.” [92]. Another example is that “some authors consistently use competency when referring to occupational competence.” [93]. In this thesis competency and competence will be used as synonyms.

When competency is being defined it usually includes the concepts knowledge and skills and then “something else”.

For example:

> “'competence' is defined here as a combination of knowledge, skills and attitudes appropriate to a particular situation.” [87]

> “'competence' means the proven ability to use knowledge, skills and personal, social and/or methodological abilities, in work or study situations and in professional and/or personal development. In the European Qualifications Framework, competence is described in terms of responsibility and autonomy.” [86]

> “then, if intellectual capabilities are required to develop knowledge and operationalising knowledge is part of developing skills, all are prerequisites to developing competence, along with other social and attitudinal factors.” [93]

The Qualifications and Curriculum Authority in the UK (formerly known as National Council for Vocational Qualifications defines competence: “the ability (knowledge, skill and understanding) to perform to standards required in employment. It includes problem solving and the flexibility necessary to adapt to changing circumstances.” [94]

5.2 Competency Frameworks

There is a multitude of competency frameworks. The reason for this diversity is that there are many reasons for developing and using the frameworks, for instance because they:

- Enable comparisons between competency frameworks.
- Define different sectors in the work force.
- Allow users to view regional (both national and internationally) issues.
- Define different domains of target users (Corporate world, Education, HR, government, etc.)
- Accommodate different purposes (e.g. enable easy transition between educational institutes, or between “world of education” to “world of work”.
- Have been developed using different methodologies (e.g. task based or functional analysis).

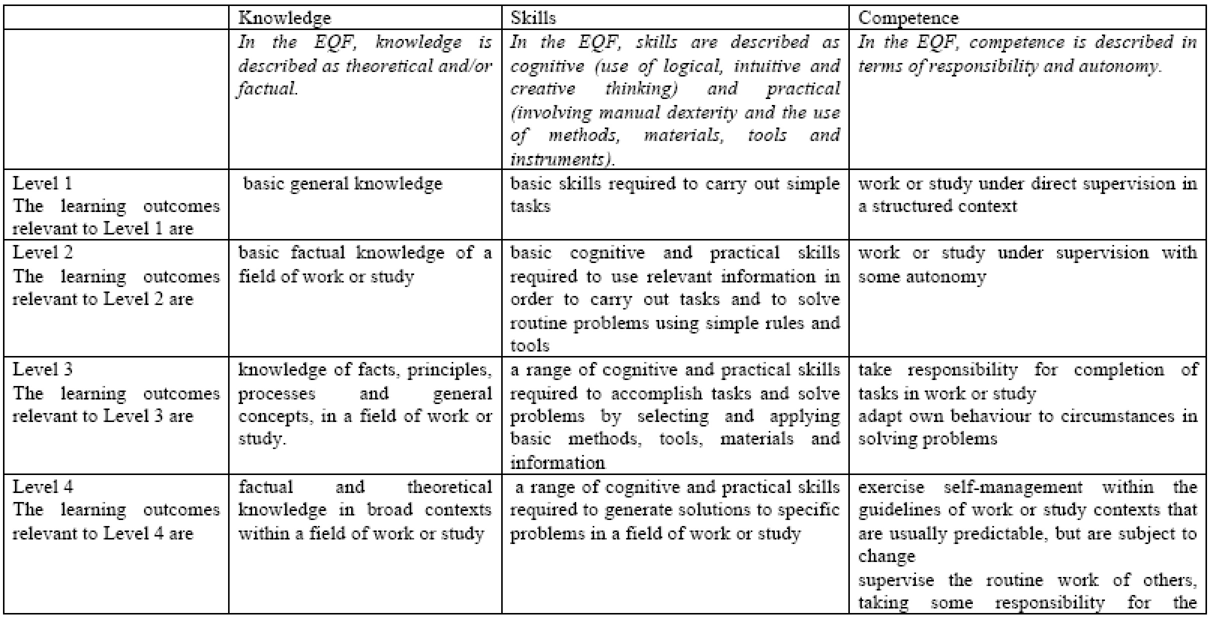
This thesis will present some different frameworks to illustrate the vast variety that exists. The frameworks have been chosen both to show the diversity and also to illustrate the appropriateness they have for the process of comparison between different competency frameworks. The reason for this is that the competency comparison research which is presented in chapter 6 will be focussing on such comparisons.

5.2.1 EQF and ICT Skills Meta-framework

At the “highest level” of competency frameworks are the Meta frameworks, which aim at being frameworks by which other frameworks can be understood and referenced, thus made to enable comparison between different competency frameworks.

The European Qualification Framework specifies as the “main purpose … is to act as a translation device and neutral reference point for comparing qualifications across different education and training systems and to strengthen co-operation and mutual trust between the relevant stakeholders. This will increase transparency, facilitate the transfer and use of qualifications across different education and training systems and levels.” [86]. It could be said that because the EQF is a qualification framework it does not relate to competencies. However in the EQF learning outcomes are being related to knowledge, skills and competence, which are defined as the ability to use knowledge and skill (see section 5.1) within the scope of EQF. In [86] it is emphasised that “learning outcomes - in the EQF understood as the statements of what a learner knows, understands and is able to do on completion of a learning process. … In the EQF learning outcomes are defined by a combination of knowledge, skills and competence.”

The EQF defines 8 levels of knowledges, skills and competences, which should be used as reference points by which learning outcomes from the different member states can reference the learning outcomes of their education system.



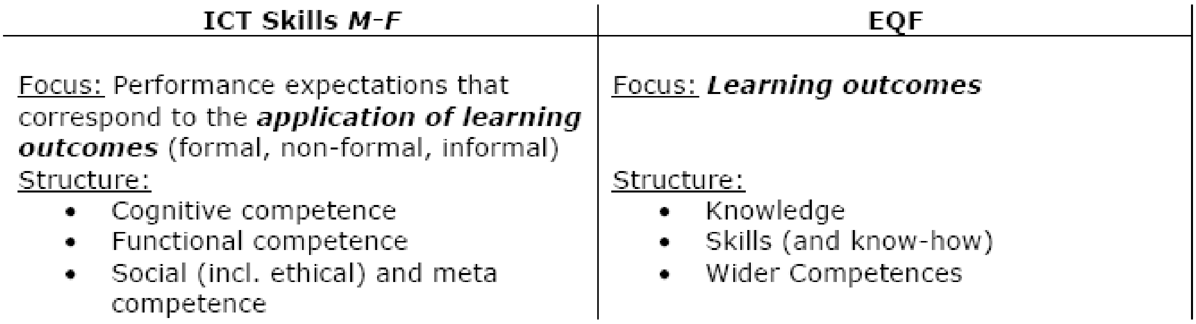
	Knowledge	Skills	Competence
	In the EQF, knowledge is described as theoretical and/or factual.	*In the EQF, skills are described as cognitive (use of logical, intuitive and creative thinking) and practical (involving manual dexterity and the use of methods, materials, tools and instruments).*	*In the EQF, competence is described in terms of responsibility and autonomy.*
Level 1 The learning outcomes relevant to Level 1 are	basic general knowledge	basic skills required to carry out simple tasks	work or study under direct supervision in a structured context
Level 2 The learning outcomes relevant to Level 2 are	basic factual knowledge of a field of work or study	basic cognitive and practical skills required to use relevant information in order to carry out tasks and to solve routine problems using simple rules and tools	work or study under supervision with some autonomy
Level 3 The learning outcomes relevant to Level 3 are	knowledge of facts, principles, processes and general concepts, in a field of work or study.	a range of cognitive and practical skills required to accomplish tasks and solve problems by selecting and applying basic methods, tools, materials and information	take responsibility for completion of tasks in work or study adapt own behaviour to circumstances in solving problems
Level 4 The learning outcomes relevant to Level 4 are	factual and theoretical knowledge in broad contexts within a field of work or study	a range of cognitive and practical skills required to generate solutions to specific problems in a field of work or study	exercise self-management within the guidelines of work or study contexts that are usually predictable, but are subject to change supervise the routine work of others, taking some responsibility for the

Figure 5.1 four of the EQF levels [86]

The purpose of the ICT Skills Meta framework (M-F) is “to promote better understanding within the European Union about the nature and structure of the ICT Practitioner Skills required by employers.” [95]. It is somewhat related to the EQF, as it also aims at creating further understanding of a domain (ICT practitioners both on the employer and employee side). In [95] it is discussed how M-F relates to EQF. Basically the EQF relates to the supply side of competencies with no specific target domain, whereas M-F is more on the demand side (employees can also demand competencies) with a specified domain in mind, but still developed with the same end result of being able to provide a common ground of its users.

There are several similarities between EQF and M-F so therefore symbiosis between them is possible. For instance they relate the structure of EQF to the M-F structure, figure 5.2 shows how similar that the basic structure is, although the structure of the M-F is further specified in sub-categories due to the specified nature of the domain.

Furthermore the M-F directly aligns its competency levels to the EQF competency levels even to the point that M-F does not have levels 1, 2 and 8 because the work group behind the M-F deemed that these levels would not be needed in the domain that is covered [95].



ICT Skills *M-F*	EQF
Focus: Performance expectations that correspond to the ***application of learning outcomes*** (formal, non-formal, informal) Structure: • Cognitive competence • Functional competence • Social (incl. ethical) and meta competence	Focus: ***Learning outcomes*** Structure: • Knowledge • Skills (and know-how) • Wider Competences

Figure 5.2 Basic structures of EQF and M-F [95]

5.2.2 National and Sectoral Competency Frameworks

There is an abundance of competency frameworks, and there almost exists a “Babel’s Tower” situation in the European Community, not just because of languages, but also because of the inability to communicate across borders both nationally and across sectors. The following listing of frameworks with very short explanations is functioning more as a validation of the variety and quantity than anything else. The list is only an extract which was compiled by the author to form the basis of a survey in TRACE project report [94] of the competency frameworks in the British Isles but the complete report showed that the situation is similar all over Europe.

- National Vocational Qualification (NVQ)

  Detailed description and levelling of vocational employment activities linked to sectors in England and Wales.

  http://www.direct.gov.uk/en/EducationAndLearning/QualificationsExplained/DG_10039029

- Scottish Vocational Qualification (SVQ)

  Similar to NVQ, but for Scotland.

  http://www.sqa.org.uk/sqa/2.html

- Skills Framework for the Information Age (SFIA)

  The Skills Framework for the Information Age (SFIA) provides a common reference model for the identification of the skills needed to develop effective Information Systems (IS) making use of Information & Communications Technology (ICT) in the UK.

  http://www.sfia.org.uk

- UK-Spec

  UK Standard for Professional Engineering Competence.

  http://www.engc.org.uk/UKSPEC/default.aspx

- BCS Accreditation

  This is the British Computer Society implementation of the Engineering Council above. They also accredit for Chartered Scientist and Chartered IT Professional.

  http://www.bcs.org/

- European Computer Driving License (ECDL)

  Vocation computer user skills framework from Ireland.

  http://www.ecdl.com

- Competency Framework for Managing Change through Partnership

  Framework for managing changing organisations to achieve higher performance based in Ireland.

  http://www.ncpp.ie/inside.asp?catid=77&zoneId=1

- BECTA Technical Competency Framework

  Competency Framework for IT technicians in schools in the UK, BECTA has been discontinued as an institution early 2010, so the future of this framework is unclear.

  http://schools.becta.org.uk/index.php?section=re&catcode=ss_res_tec_man_03 &rid=10033

The problem is that there probably is a need for each of the competency frameworks, however at the same time there is a need for a means to communicate across the communities of practice that the frameworks create. The Meta frameworks try to occupy the middle ground, but there is a perception that the Meta frameworks are good at expressing overarching high level issues and only usable for analysis by humans.

5.2.3 O*NET

The O*NET [96] is not a competency framework in the traditional sense. It is an occupational database of all the occupations in the US economy defining and organising data from the "world of work". At the core it provides a taxonomy of competency components (see next paragraph) and data is continuously collected from actual work places specifying what levels the employees in U.S. work places in different occupations scored in each individual competency component. It is therefore a statistic tool which can be used to explore the vastly (vague) domain of the U.S. job market. It is accessible to everybody through on-line tools and an application programming interface allowing a valuable insight into the composition of most U.S. jobs.

The O*NET is interesting in the context of competency frameworks because of the competency component taxonomy which organises occupation by knowledges, skills, abilities and various other categories (known as KSAO). Therefore as Ostyn suggested [97] and Brown in TRACE project meetings elaborated upon [98], it could provide a large taxonomy of knowledges, skills and other kinds of information valuable in the creation of other competency frameworks. These could be the building blocks within the competency mappings, and together with formally defined logical relationships could provide a powerful way of defining and specifying competencies.

5.3 Representation of Competency

When exchanging competency data between applications it is important to able to do this in a standardised manner which enables interoperability.

5.3.1 Reusable Competency Definition

There exists only one agreed standard to represent competencies. The Reusable Competency Definition (RCD) [99] is an IEEE standard which functions as a syntactic standard, so enabling interoperability between data systems, enabling each system to establish what parts of received data is a competency and ascertain which part of the competency data is the title, description etc. The standard originates from IMS Reusable Definitions for Competencies and Educational Objectives (RDCEO) [100], [101] 'Reusable Competency Definition' (RCD) is used to present some of the syntactical elements of competencies.

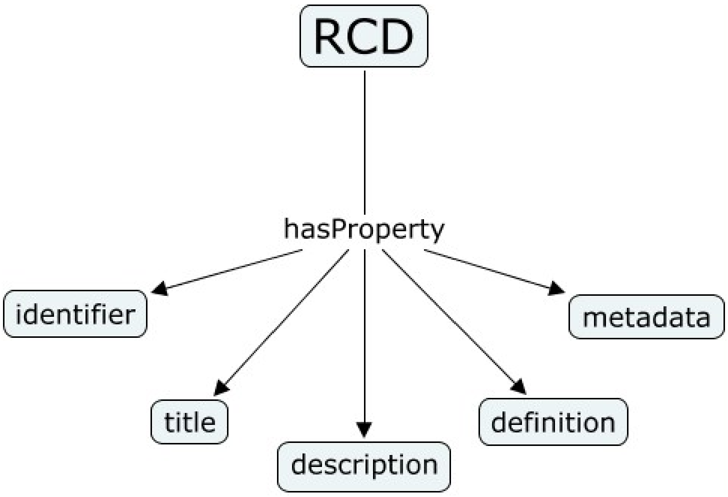


Figure 5.3 RCD

The RCD (Figure 5.3) for a specific competency should contain in natural language [99]:

- A unique identifier
- A title

Optionally it could also have

- A description (natural language)
- A definition (a reference to another repository or definition)
- Metadata (further information about a particular competency, this is not limited , it can be any size or format)

The main problem with this standard is that the main parts (title, description and definition) are in human readable form, so if any semantic meaning is to be made available for computers there must be additional knowledge, e.g. attached in the metadata part, connections to other RCDs with metadata or external bindings to other data structures such as ontologies. Furthermore RCDs are only a partial representation of competencies as they are only supposed to define competencies; the evidence, context, dimensions etc. are not included. Evidence is an especially important issue for many competency descriptions, and the RCD therefore needs to be “backed” up by some other material to be able to validate the competencies.

An example of how to use RCDs (especially with the view of evidencing) can be found in the domain of electronic portfolios (e-Portfolio), which are "a collection of authentic and diverse evidence, drawn from a larger archive, that represents what a person or organization has learned over time, on which the person or organization has reflected, designed for presentation to one or more audiences for a particular rhetorical purpose." [102]

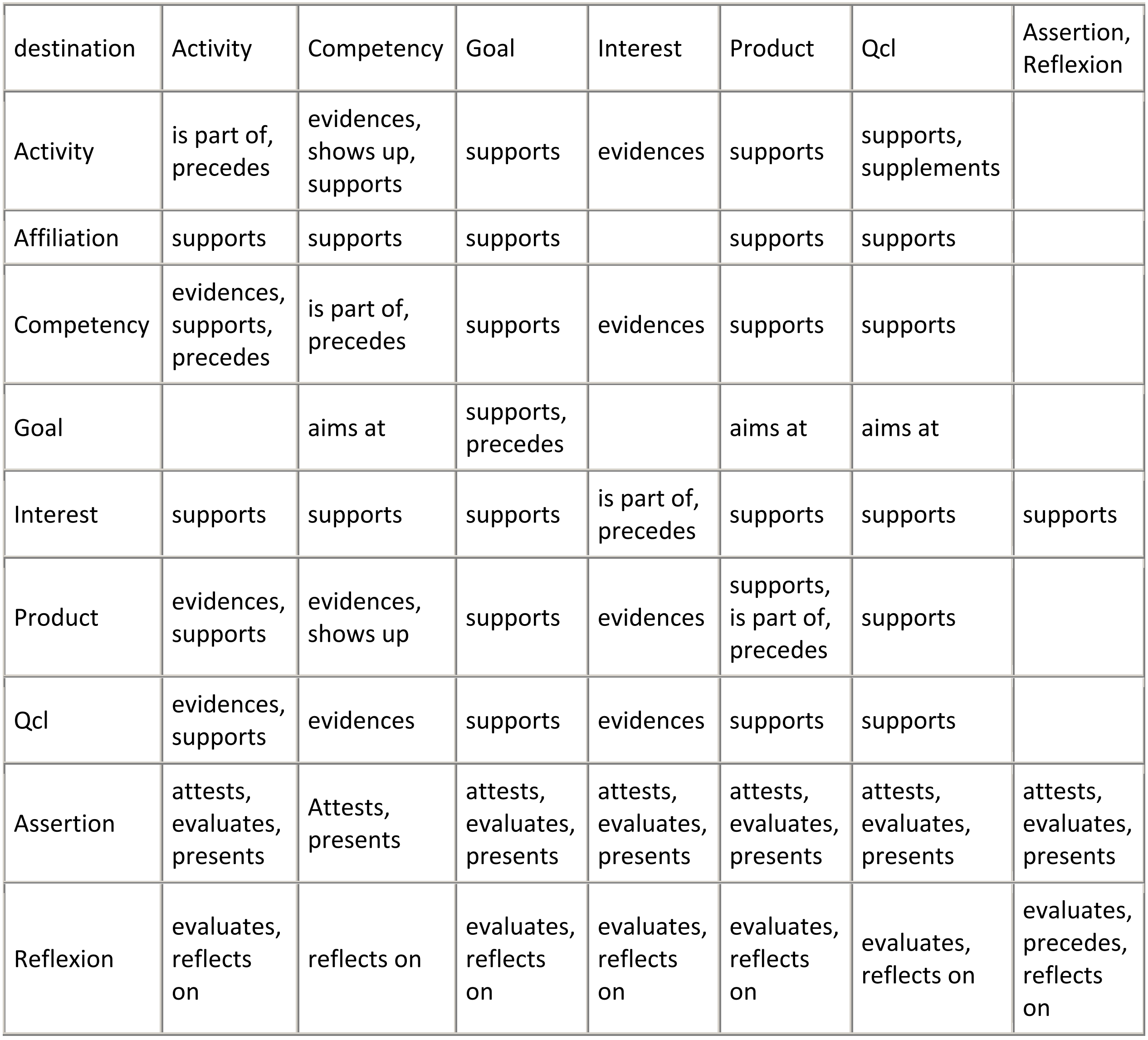
There are several reasons or objectives for making an e-Portfolio:

- Assessment
- Presentation
- Learning (document, guide, advance and reflect)
- Personal development
- Multiple users working in teams

Usually the objective will be a mix of the reasons stated above.

E-Portfolios could be used to present and provide evidence of competencies by mapping the different parts of the e-Portfolios to RCDs, hence establish the evidence and context of the RCD (competency mapping).

The IMS ePortfolio Best Practice and Implementation Guide provides guidance on how an ePortfolio ought to connect competencies in the form of RCDs to other types of material in an ePortfolio, figure 5.4 shows these relationships. For example a “Product” can evidence that somebody has a “Competency”, furthermore a “Competency” can support a “Goal” that an individual wants to reach. The purpose of these is to enable automated inferences between the different aspects of ePortfolios. However even when given and using these predefined relationships there is the inherent problem that RCDs are in natural language, hence automated computation and comparisons between them is not possible unless taking advantage of artificial intelligence methods.



destination	Activity	Competency	Goal	Interest	Product	Qcl	Assertion, Reflexion
Activity	is part of, precedes	evidences, shows up, supports	supports	evidences	supports	supports, supplements	
Affiliation	supports	supports	supports		supports	supports	
Competency	evidences, supports, precedes	is part of, precedes	supports	evidences	supports	supports	
Goal		aims at	supports, precedes		aims at	aims at	
Interest	supports	supports	supports	is part of, precedes	supports	supports	supports
Product	evidences, supports	evidences, shows up	supports	evidences	supports, is part of, precedes	supports	
Qcl	evidences, supports	evidences	supports	evidences	supports	supports	
Assertion	attests, evaluates, presents	Attests, presents	attests, evaluates, presents	attests, evaluates, presents	attests, evaluates, presents	attests, evaluates, presents	attests, evaluates, presents
Reflexion	evaluates, reflects on	reflects on	evaluates, reflects on	evaluates, reflects on	evaluates, reflects on	evaluates, reflects on	evaluates, precedes, reflects on

Figure 5.4 Relationships between elements in ePortfolio systems [102]

5.3.2 Competency Mappings

It was observed while creating a bespoke ePortfolio that inherent semantic and logical relationships between competencies are needed when creating a picture or representation of a person's competency [103], and as such the RCD standard lacks the descriptive power to transfer such knowledge in the exchange processes.

Ostyn has also identified this problem and proposed a standard in the IEEE draft standard "Simple Reusable Competency Mappings" (SRCM) [5]. A SRCM like the RCD consists of

- A unique identifier
- A title

Optionally it can also have

- A description
- Metadata (further information about a particular competency, this is not limited)

But in addition the SRCM has a directed acyclic graph (DAG) of nodes with attached competencies. A DAG is a "directed graph with no path that starts and ends at the same vertex." [104] The graph must have at least one entry node (the default entry node). Note SRCM does not include the definition part which RCDs have, because the graph provides the equivalent (above and beyond) functionality, as the mapping enables computational representation definitions of the actual competencies, which is the whole purpose of the RCD's definition part.

Each node in the DAG can have several related properties, for instance:

- Some competency (RCD or another SRCM)
- Proficiency scores (Required or Desired)
- Relationship to other nodes within the graph (Parents and Children), where the relationship could be defined with a score or some kind of logical relationship.

When analysing the SRCM standard, it becomes obvious that it is very much a draft standard. For instance, it is not possible to attach proficiency scores other than required and desired score. There is no way, for instance, for people to represent that they have a competency with a proficiency score. Additionally the logic relationships within the

SCRMs are not based on any formal logic which, if present, could help automated understanding the actual logics behind them. An alternative competency mapping will be introduced in section 6.2.

5.4 Summary

This chapter introduced the vague domain of competency and competency frameworks. It began by examining different definitions of the terms, which highlighted the vagueness of the domain. Several competency frameworks were introduced on the basis of their functionality and purpose. The chapter ended by presenting the computer standards, which are available. The next chapter will describe the novel competency system the “Competency Suite” which enables comparison of competency using ontologies.

6 Competency Comparison

6.1 Introduction

The competency domain is, as described in the previous section, an extremely vague domain, with plenty of divergent opinions, and different ways of describing the same concepts. This chapter discusses the approach taken to describe and compare competency and competency mappings by utilising ontologies when there is significant vagueness in the domain. This work was part of the European Union (EU) Leonardo sponsored project TRACE [4]. Essentially the project was concerned with a European Competency Framework system referred to as E*NET. At the core of the work a competency inference engine was developed together with an application programming interface (API) and a set of tools based on this API which was referred to as the “Competency Suite”. The approach to the representation of competency, competency frameworks and links between competencies, together with the ontology based inference engine and toolset go beyond what was previously available. They form the basis of a new approach to competency representation that is the main subject of this thesis.

6.2 Initial Considerations

Early on in the TRACE project, it was realised by the partnership of TRACE that even though the domain of Competency Frameworks is a vague domain, there were some similarities shared by domain experts. As described in chapter 5 these “universal” concepts are knowledge, skill and other attributes (KSO), sometimes with abilities being included separately (KSA or KSAO). From this it was concluded, following Ostyn’s and Brown’s suggestions [97], [98], that it would be possible to develop an “upper ontology” of competency (see section 2.1.4 Different types of ontologies).

Figure 6.1 is an early diagrammatic overview of the proposed representation system including some foreseen applications that would utilise E*NET (made by the author). It illustrates how E*NET will be able to provide the middle ground for different competency suppliers (i.e. frameworks) and demanders (i.e. ePortfolio and job descriptions) and, provided adequate inference, seamlessly make comparisons between the different domains competency usage. To get to this level of inference the semantics of the upper ontology would need to be specified, rules based on this upper ontology between different frameworks would be needed (domain and task ontology), and tools would be needed at the application level to make the links between the different kind of usages that would arise from using E*NET (i.e. evidencing and requirements).

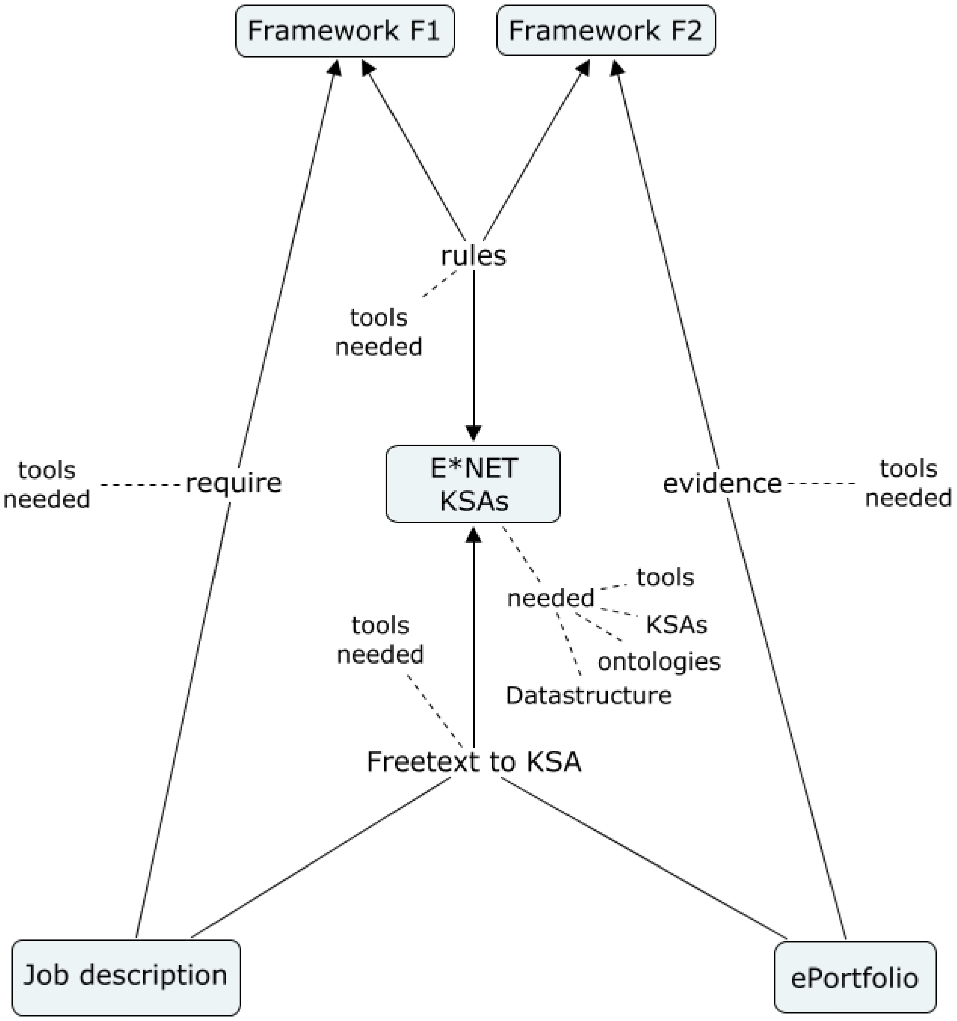


Figure 6.1 Proposed System

The flexibility for users of the system would then lie in the extensions they could make to existing statements of E*NET and other extensions, by utilising logical semantic relationships (taken from linguistic logic) such as synonymy, antonymy (opposites), meronymy (being part of), hyponymy (specialisation) etc. [105]. This would allow the creation of user specified semantic trees of bespoke knowledge. Figure 6.2 is an early example of such a tree.

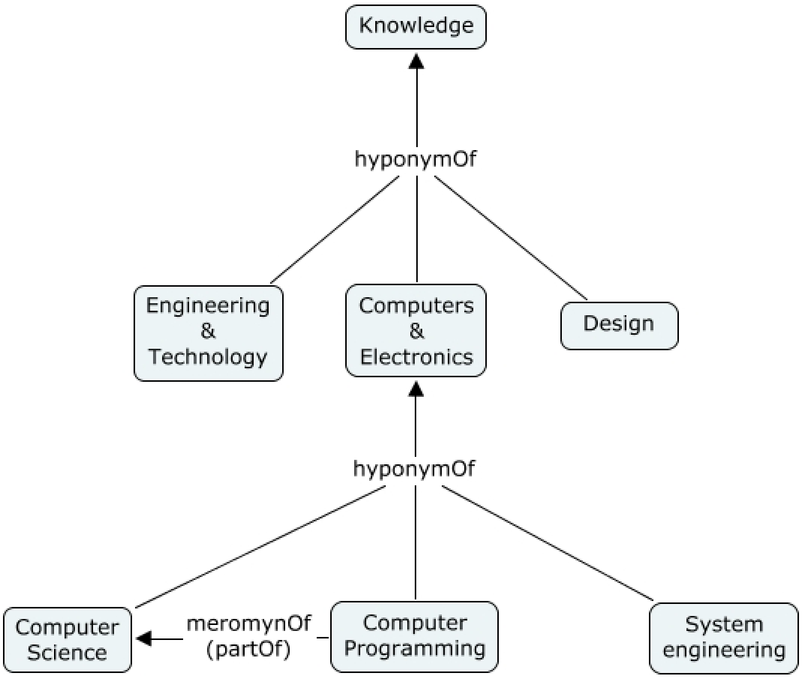


Figure 6.2: Knowledge tree

It was necessary for the TRACE project to determine whether the O*NET could be used as the basis for supplying descriptors to be used within E*NET as a starting point for a competency domain ontology. This was done by a series of feasibility tests. These tests concluded that it would be feasible as it is a comprehensive taxonomy, although appropriate changes would be needed for usage on a broader scale as parts of the O*NET are very "American". [106], [107], [108], [109]. This approach can be criticised; the O*NET classification is not intended as a formal domain ontology of competencies, and using it as one might pose some problems. For instance the individual descriptions are written in non-formalised English, and could therefore introduce inconsistencies. Furthermore the structure of the O*NET classification is not always completely clear. However the usage of O*NET merely constituted a starting point for the research, obviously by using ontological tools this starting point could be changed once the approach is standardised, and for this research a prototype was needed to prove the concept.

Additionally by utilising the emergent competency standards the semantics defined in, and used through, the “upper ontology” would be sharable between many different applications. The development of RCDs were finalised by IEEE in the life time of the TRACE project, and thus used throughout the project in an XML variant (Appendix C), as the container for each “unit” of competency, for instance writing, driving or mathematics. The standardisation process of SRCM, however, is still ongoing at the time of writing, and we believe there are serious flaws in the standard. For instance the important aspect of being able to describe somebody having a proficiency in a competency. In the proposed standard this would be impossible, only allowing “required” and “desired” proficiency. Therefore a bespoke standard called Very Simple Reusable Competency Mapping (VSRCM) was developed based on the SCRM [5] with a few alterations.

The author is participating in the IEEE Working Group for standardisation of Competency, and is therefore both following and shaping the developments of a standard of Competency Mappings.

We define VSRCMs like RCDs consists of

- A unique identifier
- A title

Optionally it could also have

- A description
- Metadata (further information about a particular competency, this is not limited)

Note VSRCM does not have a definition section as RCD's have, the graph provides an improved equivalent functionality.

Additionally the VSRCM has a graph of nodes with attached competencies. The graph must have at least one entry node (the default entry node).

Each node has properties:

- Competency
  - RCD

    or
  - VSRCM (note this could be recursive)
- Proficiency (levelling can be user-defined, with support for ontological definitions)
  - Required
  - Desired
  - Current (has)
- Relationship to other nodes within the graph
  - All

    That is, all the proficiencies of the competencies of the "sub-nodes" need to be "fulfilled" for this relationship to be successful
  - Any

    That is, one or more of the proficiencies of the competencies of the "sub-nodes" need to be "fulfilled" for this relationship to be successful.

- If
    - True
    - False

  This is used to represent alternate proficiencies of competencies, for example a taxi driver based in London is required to have specific knowledge of the area, while a taxi driver elsewhere may only require general map reading.

Figure 6.3 is a graphical representation of the different parts of RCD and VSRCM and how they interrelate. Used to describe a complete competency profile the graph structure of the VSRCM allows the semantic relationships between competencies to be presented in a logical syntactic way, by building up a graph of how competencies are related to each other in the competency map of the profile. Furthermore different mappings can be compared and contrasted automatically by traversing the graphs of the VSCRM using the logical rules throughout the process, however it is not truly a semantic representation of the competencies, because there is no ability to represent relationships between individual competencies within the nodes of the graph. This is achieved in this work by utilising ontologies.

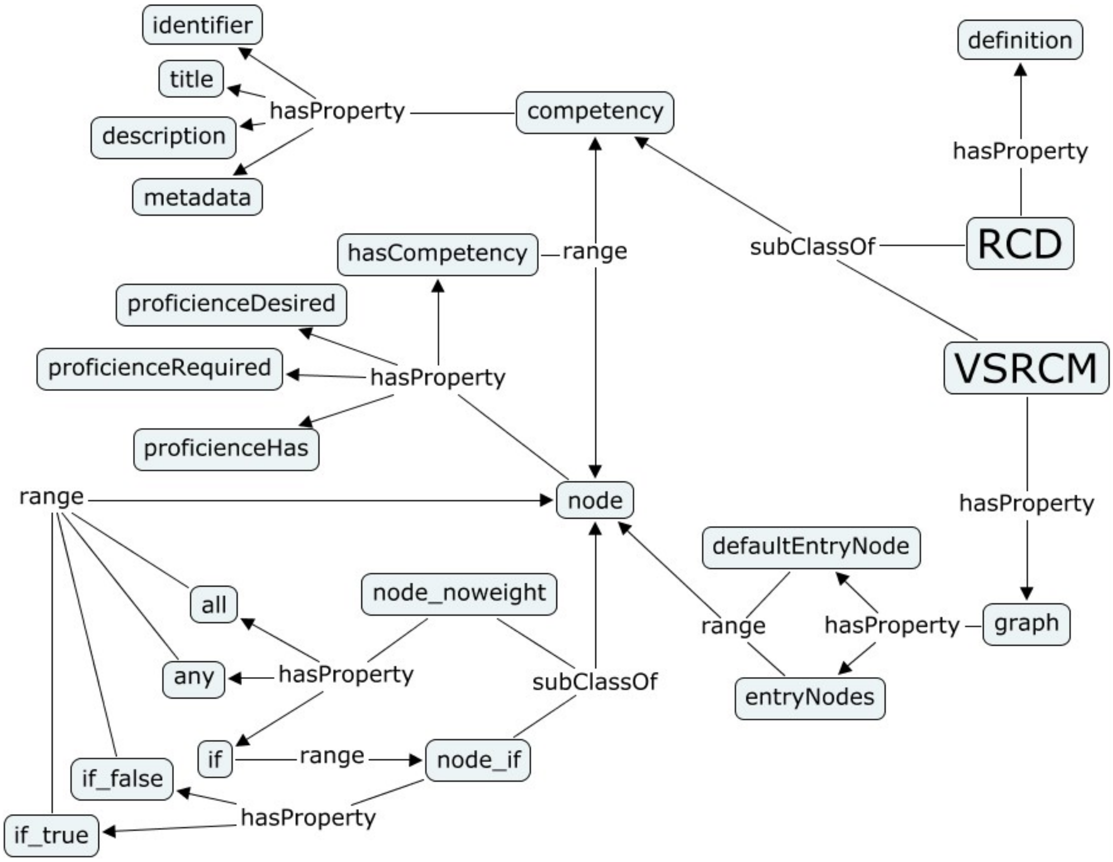


Figure 6.3: Relationship between RCD and VSRCM

6.3 The Upper Competency Ontology

It was decided to utilise ontologies as an underpinning approach to allow for and verify the increased interoperability between competency systems when performing comparisons [103]. Chapter 3 listed several alternatives to ontologies, however the expressiveness of ontological languages, the fact that humans can "understand" them and modify them, and that the results of inferencing engines are verifiable meant that this method would be the best for this vague domain.

There are, as described in chapter 2 many ways to describe Ontological knowledge. OWL was chosen as the underlying ontology language because it is used in the Semantic Web approach, and therefore has many technologies that support its features.

The diagram in figure 6.4 represents a simplified view of the ontology underpinning the system (the upper ontology). Appendix C is a listing of the OWL language files which make up the internal ontologies in TRACE.

The classes with the TRACE Comparison Grid ontology are:

- Single Competency (RCD)
- Competence Profiles (VSRCM)

RCDs are further defined in to sub-classes:

- Knowledge
- Skill
- Others

The top-left of the figure shows the different semantic relationships that are specified within the upper ontology to interrelate all the different competencies. These can be of type single competency (RCD) or competency profile (VSCRM), which incorporates a graph of logically related competencies. The single competency components can be of type skill, knowledge or “other”.

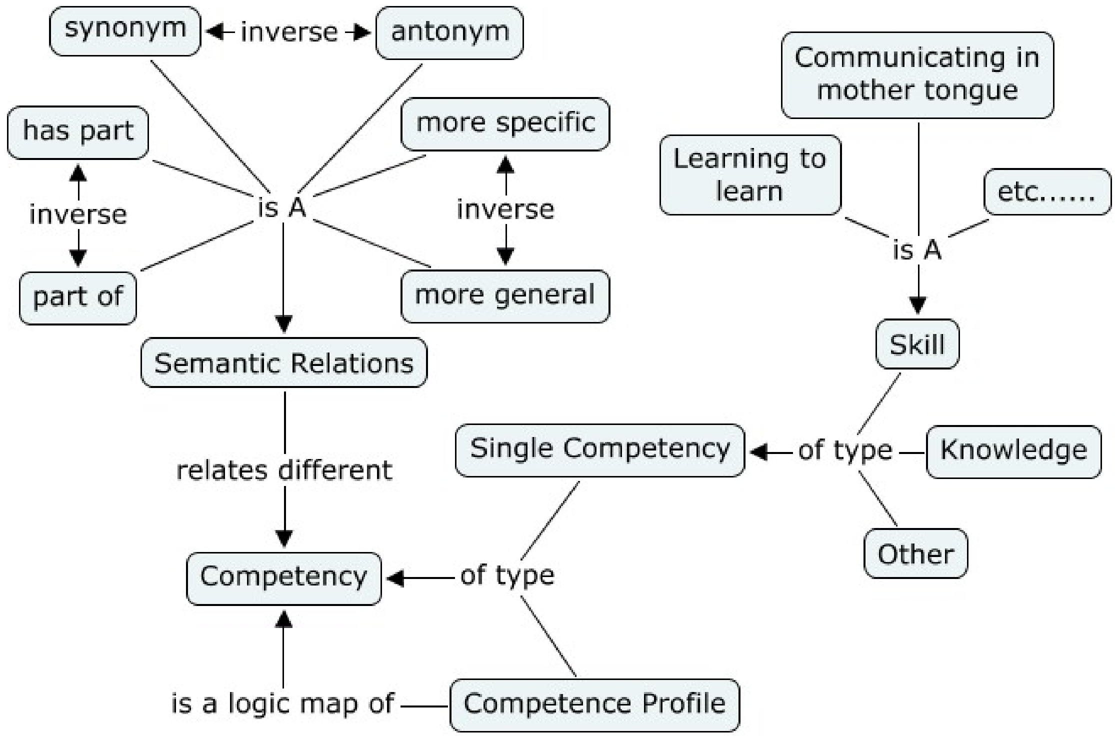


Figure 6.4: Representation of the ontology

The individual competencies used by each of the sub-classes knowledge and skills are based on the O*NET classification [96] with small alterations. As described in the previous section this was chosen as a reasonable starting point for the prototype E*NET through feasibility tests. A bespoke European KSO specification would require standardisation and consensus building on a grander scale, and for the purpose of this research a prototype that could validate the approach was needed.

The semantic relationships, specified within the upper ontology, which therefore can be used to create new ontological specifications between individual competencies (both RCDs and VSRCMs) are based on semantic relationships from linguistics, that have been widely used in knowledge representations, for instance in the electronic lexical database for the English language WordNet [105]. A semantic relationship is represented as a property:

- Alternates
    - Synonym

      Competencies which mean the same; for example different terms used for the same competency across frameworks.
    - Antonym

      Competencies which mean the opposite; for example the competencies: empathy and impartial. This could be useful for inferences.
- Part, either:
    - A has part B (holonym)

      That is the relationship that competency A intrinsically includes B; for instance drive has part that is use of brakes.
    - A part of B (meronym)

      That is the relationship that competency A intrinsically included in B for instance drive is part of competency to be taxi driver.
- Generality
    - A is more general than B (hypernym)

      A includes all the meaning of B, but B includes more detail.

      For example driving is more general than driving a lorry.
    - A is more specific than B (hyponym)

      B includes all the meaning of A, but B includes less detail.

      For example lorry driving is more specific than driving.

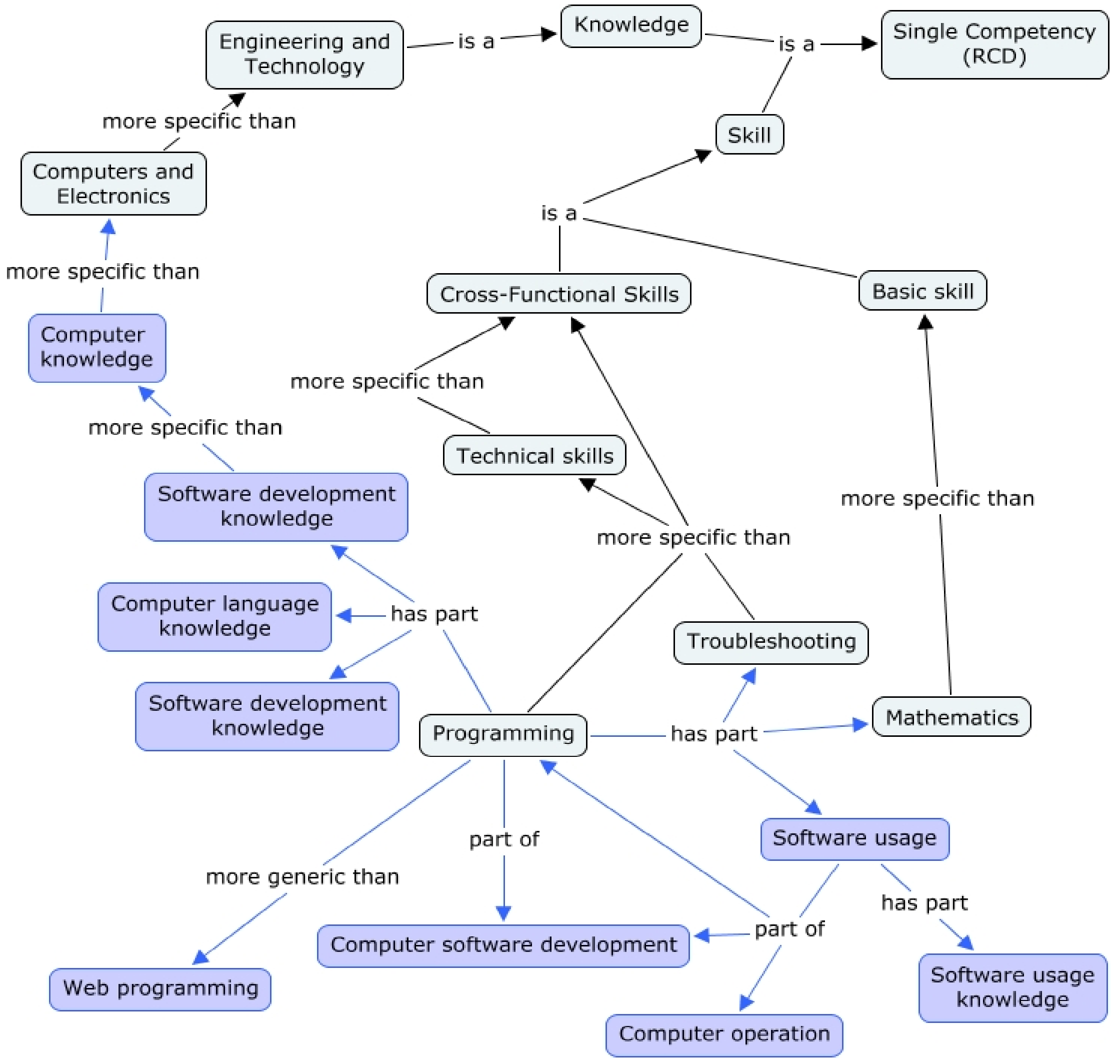


Figure 6.5 Grid of semantic links between E*NET RCDs and bespoke Computer Science Domain

By using the specified knowledges and skills from O*NET and the defined semantic relationships it is now possible to extend the complete knowledge base in a consistent manner, which allows for further inferences on the additions made by others across domains and applications. Figure 6.5 is an example of a competency mapping which has had knowledges and skills added from a small sample of Computer Science descriptions (dark shaded). Using the predefined semantic relationships, it is possible to define relationships and add them into the pre-defined knowledgebase (light shaded). The pre-defined knowledge base is, in this instance, knowledge from E*NET. The user (knowledge engineer), who has added the new information, has done so according to his own belief. It is important that the people, who need the system to work for their situation, have the total freedom to specify any semantic links between competency components, likewise that they can remove any knowledge from the base of the inference, so that the semantic system will provide the user with inferences that make sense in their work. All this is possible, if the user uses the upper ontology provided through the system. The API provided through the system has been designed so that knowledge can easily be taken in and out of the inferencing engine. The idea behind this is that in order to avoid ambiguities within the vague domain each user of the system should be allowed to use any knowledge they need and add any new information to the system by adding new models (of semantic knowledge) to it, but in small pieces at a time. Different users can then use the inference engine of the system to inspect these small models of other users to see if they are appropriate to use in their situation. This follows to a certain extent how the Semantic Web is designed to operate, where the multitudes of available ontologies are referenced and brought into the inferencing engine as they are used and referenced. Anybody can make these references, and therefore the information in the ontologies is prone to misuse (as discussed in section 3.4). The differences here are that:

- There is a defined upper ontology which users have to reference, which makes it easier to evaluate new ontological information and identify good information contra misinformation.
- There is an emphasis on making small ontology models, which makes it simple for the users to choose the appropriate (for their situation) ontologies.

- The linguistic semantics, which are not normally part of ontology, make it possible for users to create further statements about existing competencies, which are not as formally specified as traditional ontological statements.
- A reificated inferencing system as described in Chapter 4 could be used in the future to allow an even better base for inference enabling different views of competencies to be present as the basis of the same ontological inferences.

Later in this chapter these additions (plus several others) will be used to make comparisons between different competency profiles.

Competency profiles can now be created using all of the RCD's in a total model including the user defined additions. Figure 6.6 is a representation of a competency profile developed from the specification of Technician A in Desktop & Application Support (TECH A) in the BECTA ICT Competence Framework [110], it is important to emphasise that the process of making the profile is a subjective engineering process, which is obvious due to the vagueness of the domain.

The profile has been made by analysing the competency framework specifications and assessing how the words could be mapped to the competency definitions in the E*NET System. This is a non-trivial process which hardly can be performed without an understanding of the domain that the framework specifies.

According to the BECTA ICT Framework specification the activities of a TECH A are [110];

- Connect up and check hardware for normal operation.
- Set up a suitable desktop environment for users of a standalone PC.
- Install simple software.

The knowledge needed is;

- Confident user of common hardware and OS.
- Confident user of common desktop application software.

The following are some examples of what they do;

- Check floppy disk and CD-ROM drives are working.
- Help a user save a file to shared area.
- Arrange desktop icons and create class work areas on a classroom PC.
- Set up a program for children to use on a class computer.

The knowledge engineer has then gone through an assessment of each of these activities, knowledges and examples matching them with the KSOs (contained in RCDs) of E*NET and other RCDs and VSCRMs within the full model of knowledge (for instance the bespoke computer science domain from figure 6.5) connecting them using the logical links within competency mappings.

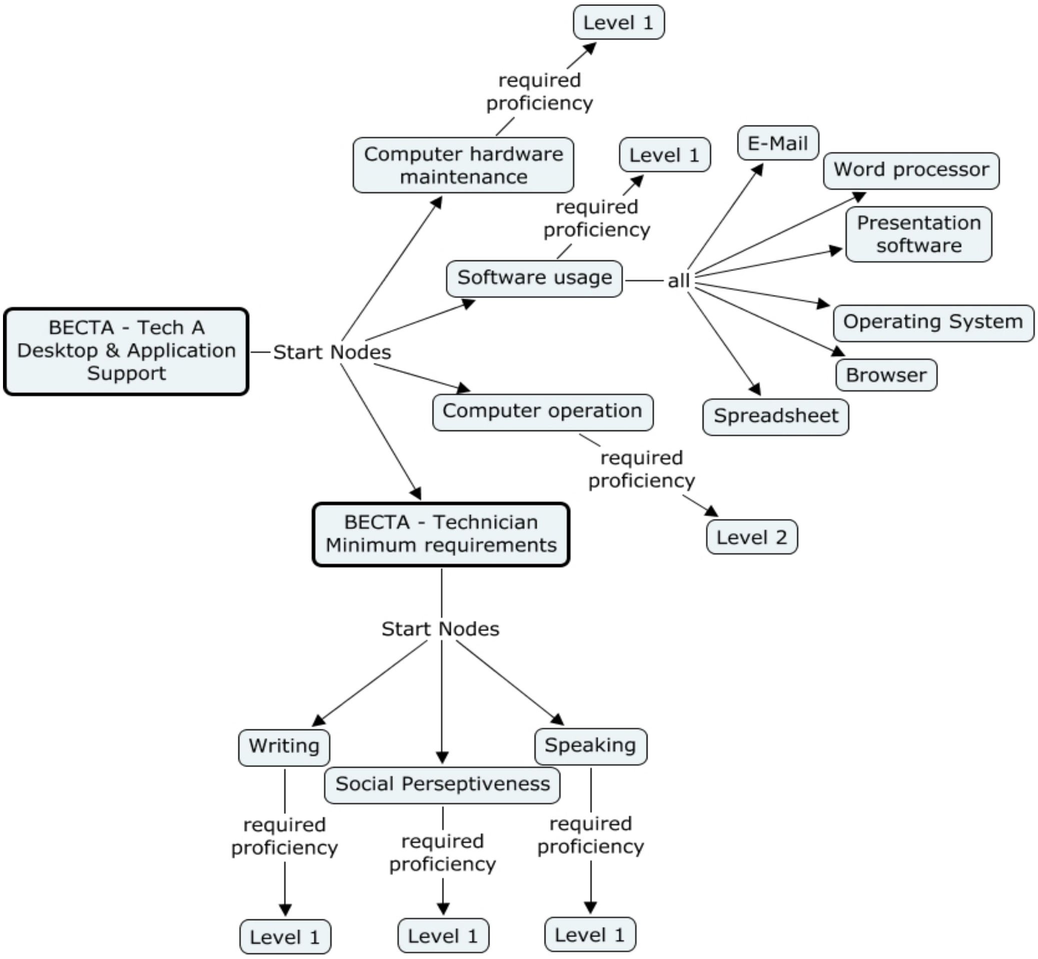


Figure 6.6 An example competency profile

The profile is made up of a graph with four start nodes, which are all necessary requirements for a person who fulfils the TECH A level of competency (given the definition of start nodes within VSCRMs);

- BECTA – Technician Minimum requirements
- Computer operation
- Software usage
- Computer hardware maintenance

The node (BECTA – Technician Minimum requirements) is another competency profile which describes the assessed minimum requirements of any person working in a job described through the competency framework; the knowledge engineer developed this competency profile as part of the competency analysis, having identified that this use of competencies was repeatedly used throughout the BECTA ICT Framework. Everybody working as IT technicians in schools should have some level of proficiency in writing, speaking and social perceptiveness. For instance "Social Perception" within O*NET (and therefore E*NET) is defined as "Being aware of others' reactions and understanding why they react as they do." [111] It would be hard to imagine any person without this skill (at least a low level) working in a school environment.

Additionally this framework level (TECH A) specifies that computer hardware maintenance, computer operation and software usage are necessary to fulfil this level, albeit at a quite low-level. These are necessary components of the profile because a TECH A person needs to be able to "Connect up and check hardware for normal operation", "setup a suitable desktop environment for users of a standalone PC" and "Help a user save a file to shared area". Furthermore under software usage the contexts e-mail, word processor, presentation software, operating system, browser and spreadsheets must be satisfied, because TECH must be a "confident user of common desktop application software".

Many of these analyses have been performed by the author, and it is evident that it is better to perform these with a group of experts, that can discuss the meaning of the frameworks, as seemingly the best results have been obtained in group settings. It is also apparent that further research is necessary to establish the best practices for these processes especially to decrease subjectivity. Human Resource research has studied how to perform analysis of skills in education and work based scenarios e.g. using psychology and sociology methods [89], and their practices could benefit the analysis necessary to build these competency profiles.

6.4 Implementation of the System

In order to make an easy development platform for developers who are not used to traditional ontology development the system was created with a core API using Jena [26]. Some of the classes from the system can be found in Appendix D.

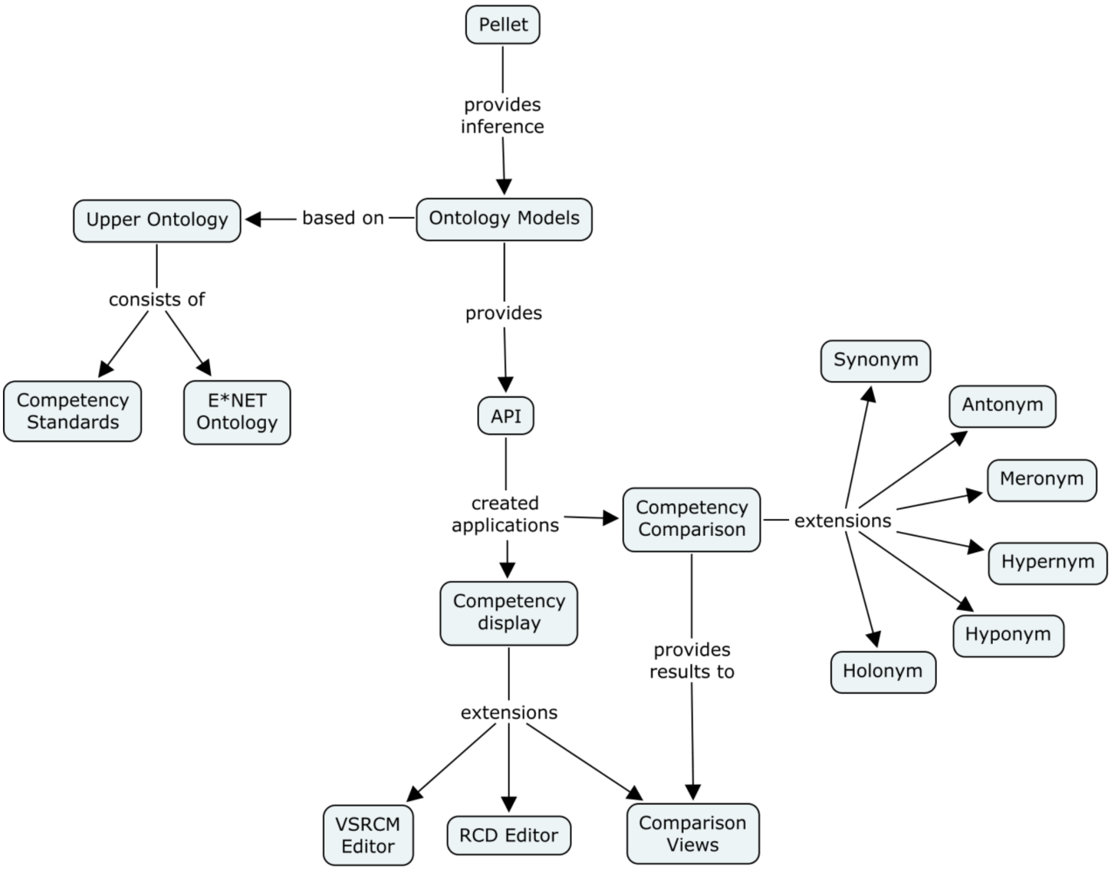


Figure 6.7 System overview

The E*NET API (figure 6.7) system allows the users to pick and chose the needed domain ontology models, as well as building extensions, and swop them in and out of the full ontology model (figure 6.8), which is used for inference. This can be done as needed depending on which models are required to perform the “right” kind of inferences. Remember this is a choice made by the user of the system depending on the ontologies which are to hand and their appropriateness to the inferences that are necessary to perform. The system consists of four levels:

- The E*NET Internal Ontology level, which is always part of the system.
- Userspace ontology model, which is the model that can be modified through the API.
- Outside Userspace ontology models, which are all models that are regarded as knowledge within the system, thus the system is “aware” of them, while performing inference.
- Ontology models, which are available to the system, but not in use.

This allows the users of the system, to minimise the effects of vagueness by disregarding the parts of any ontologies that they disagree with. The user can simply remove the parts that are disagreed with in a particular model, however the reification system from the chatbot system allowing multi-user statements with disagreements is not utilised in this.

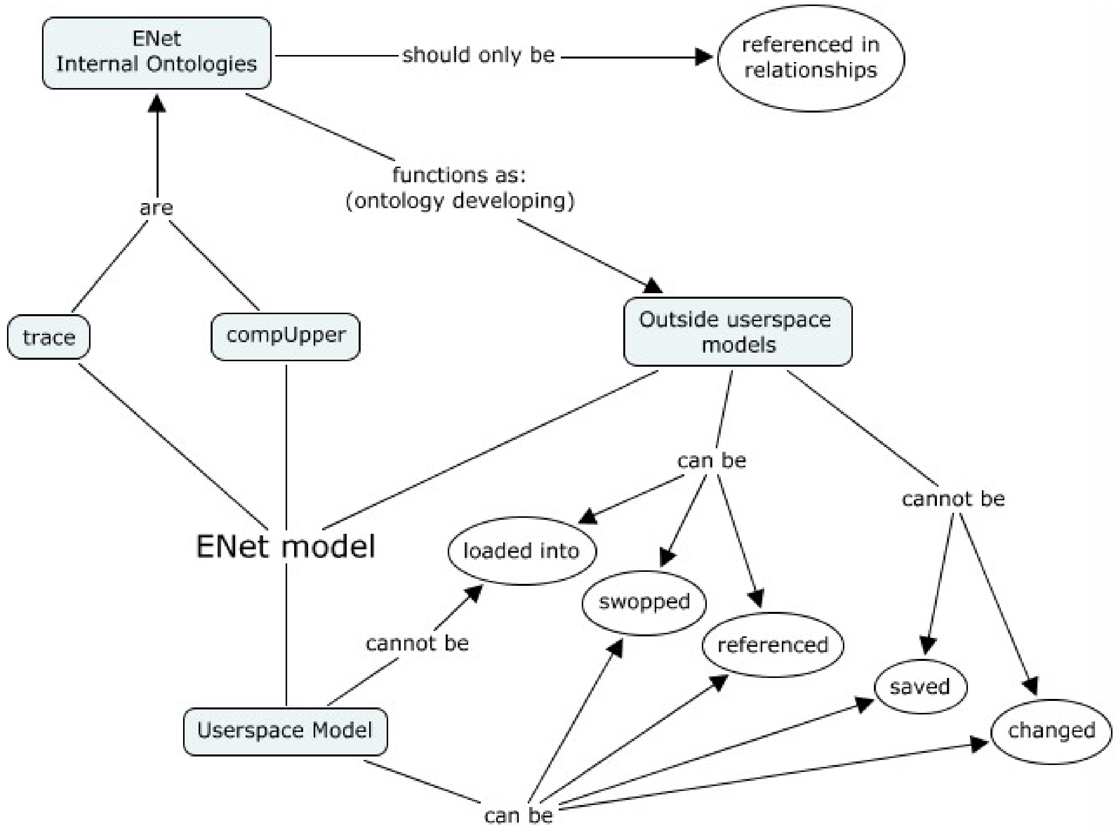


Figure 6.8 User defined ontology (models) can be swopped in and out of memory

The trace and compUpper models are the E*NET internal ontologies which function as upper ontologies. They function just like any other model except that they cannot be swopped in and out of the complete model, but, as upper ontologies, are always part of the complete inferencing. The system has API calls to perform the loading, swopping, referencing and removal of models from the complete model. For instance, a base model with upper ontologies and a user model ready to develop new competencies and competency profiles, can be set up with one line of code:

```
this.model = new ENet("http://www.oster-lundqvist.com/test");
```

To remove the model which is in userspace and load a new from a file can be done by:

```
this.model.removeUserModel();
this.model.loadModel(“/home/karsten/models/test.owl);
```

Using the API a prototype Competency Suite with a graphical user interface (GUI) was developed which allows the users to:

- create new and edit existing single competencies (RCD).

  Enables adding descriptions to all the different parts of RCDs.

- create new and edit existing competency profiles (VSRCM).

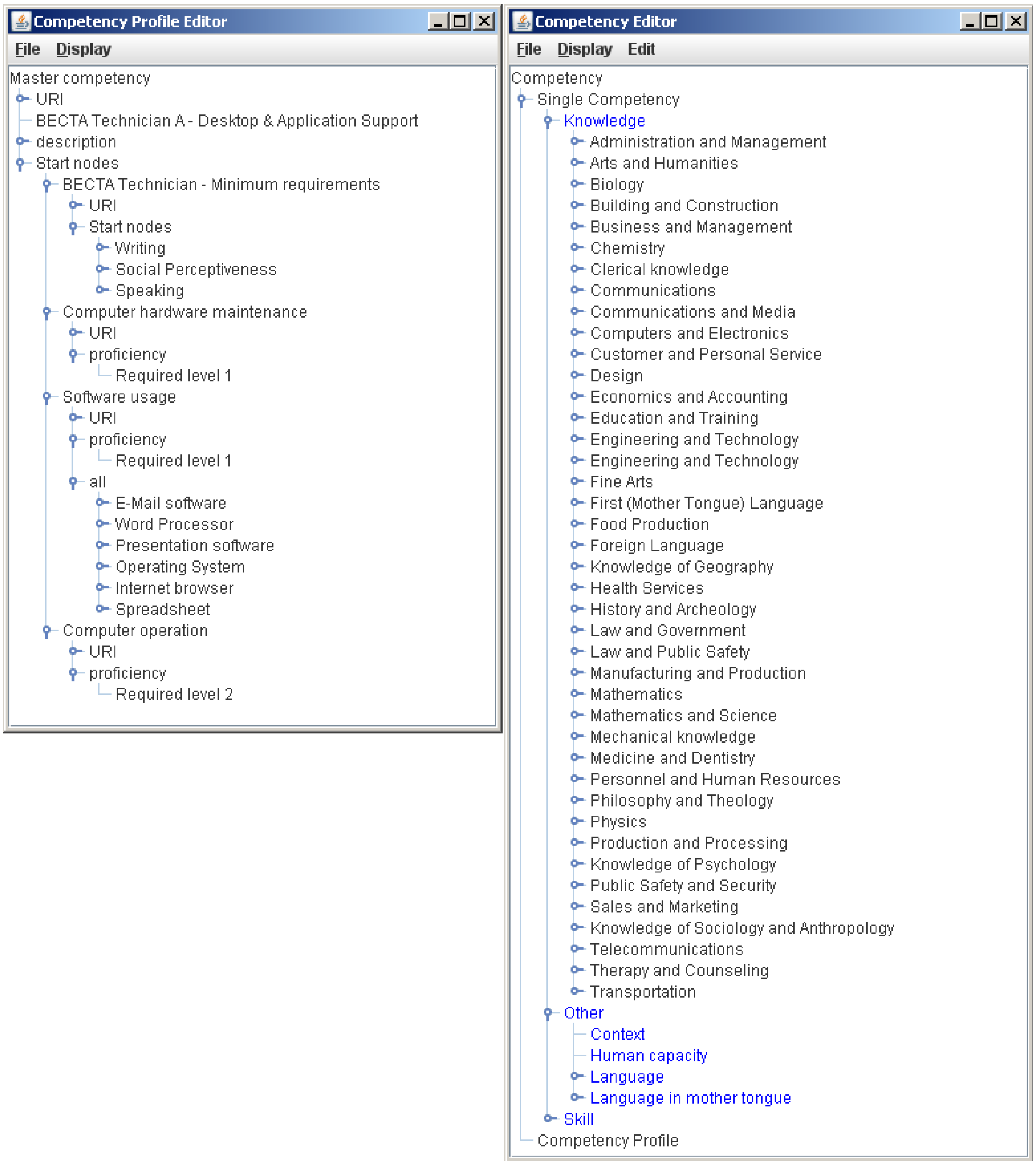
  Like RCDs enables user interaction to all parts of the VSCRM specification including creation of the graph of competencies. This is done by linking RCDs and VSCRMs together using the logic relationships specified in the VSCRM specs.

- add and remove linguistic semantic relationships between single competencies and competency profiles.

- perform comparisons between two existing competency profiles.

  This is achieved in a separate application purely created for comparisons. The user selects a competency which needs to be compared to another competency.

Figure 6.9 are screenshots of the BECTA – Technician A Desktop & Application Support level (TECH A) and the list of predefined knowledges in the E*Net System. The emphasis of these prototype tools has been on demonstrating the concepts of the TRACE API, therefore the design of this toolset is experimental and most attention has been on functionality rather than usability for the end users. The window to the left shows the competency profile for TECH A, with the four different starting nodes as discussed in the previous section. Furthermore the minimum requirements competency profile (the first starting node in TECH A) can be traversed through the GUI. The window to the right is the selector window, where the knowledge engineer can select from all the complete set of competencies (RCDs and VSRCMs) and add them to the competency profile on the left through drop down menus in the GUI. The only seeming difference between the directed acyclic graph (DAG) of the VSRCM and the graph displayed in the prototype editor is that it is shown as a “tree” view, therefore the same node might be shown several times if several nodes have semantic links pointing to the node. This does not change anything about the DAG of the VSRCM, only how it is displayed; the internal links of the ontology are following the specification.



- Master competency
  - URI
  - BECTA Technician A - Desktop & Application Support
  - description
  - Start nodes
    - BECTA Technician - Minimum requirements
      - URI
      - Start nodes
        - Writing
        - Social Perceptiveness
        - Speaking
    - Computer hardware maintenance
      - URI
      - proficiency
        - Required level 1
    - Software usage
      - URI
      - proficiency
        - Required level 1
      - all
        - E-Mail software
        - Word Processor
        - Presentation software
        - Operating System
        - Internet browser
        - Spreadsheet
    - Computer operation
      - URI
      - proficiency
        - Required level 2

- Competency
  - Single Competency
    - Knowledge
      - Administration and Management
      - Arts and Humanities
      - Biology
      - Building and Construction
      - Business and Management
      - Chemistry
      - Clerical knowledge
      - Communications
      - Communications and Media
      - Computers and Electronics
      - Customer and Personal Service
      - Design
      - Economics and Accounting
      - Education and Training
      - Engineering and Technology
      - Engineering and Technology
      - Fine Arts
      - First (Mother Tongue) Language
      - Food Production
      - Foreign Language
      - Knowledge of Geography
      - Health Services
      - History and Archeology
      - Law and Government
      - Law and Public Safety
      - Manufacturing and Production
      - Mathematics
      - Mathematics and Science
      - Mechanical knowledge
      - Medicine and Dentistry
      - Personnel and Human Resources
      - Philosophy and Theology
      - Physics
      - Production and Processing
      - Knowledge of Psychology
      - Public Safety and Security
      - Sales and Marketing
      - Knowledge of Sociology and Anthropology
      - Telecommunications
      - Therapy and Counseling
      - Transportation
    - Other
      - Context
      - Human capacity
      - Language
      - Language in mother tongue
    - Skill
  - Competency Profile

Figure 6.9 Screenshots of the competency editor from the "Competency Suite"

A prototype based on the developed API and underlying theory and methodology was developed by the TRACE partner BitMedia which shows the functionality of the E*Net System in a HR (Human Resources) scenario with emphasis on matching jobs in a job database to job seeker profiles [112]. Additionally an agent based system utilising the API has been proposed and is under development where the different functionalities are performed by agents to speed up the processes [113].

6.5 Comparison of Competency Profiles

A comparison tool was implemented to be able to compare and contrast different competency mappings, and included in the E*Net API System. The purpose of the comparison is to analyse whether a competency profile A (profA is "covered" or "matched" by competency profile B (profB). By "covered" it is meant that profB is describing an agent (e.g. a person) who has (equal to or over and beyond) the required (and possibly also desired) proficiency levels of competencies contained in the graph of profA. By "matched" it is meant that profB describes a competency profile (e.g. a job profile or a competency framework level) which has (equal to or over and beyond) the same required (and possibly also desired) proficiency levels of the competencies contained in the graph of profA. The comparison tool returns the results from the complete graph of profA, meaning that the result of each node has been examined. Such a comparison is obviously not symmetric:

$$\text{Compare(profA, profB)} \neq \text{Compare(profB, profA)}$$

For instance a very simple profile which only has one node with only ‘level 1’ required proficiency score in ‘writing’, can easily be matched by the "BECTA – Technician A Desktop & Application Support" level, because the minimum requirements of the competency profile for this job function states the requirement of ‘writing’ at ‘level 1’. On the other hand the simple profile would not match any node other than writing in the “BECTA – Technician A Desktop & Application Support” level, hence it would not match the full graph.

Without the semantic links between the different competencies the comparison of the competency of each node would be very simplistic. It would either be the same RCD with the same identifier and match, or it would be two different RCDs and hence not match each other. This could easily lead to the situation in Figure 6.10. Two different VSCRMs (the triangles) are being compared. The arrows going out of the VSCRMs are linked to different starting nodes of the DAG, and the two VSCRM are clearly pointing at different RCDs (the circles), and therefore the two different profiles seemingly do not match, but by introducing semantic relationships through the ontological mappings (illustrated by the brown lines in the lower part of the illustration) in the ontological description a match can be inferred between the competency profiles, because there are semantic links between the different RCDs used by the two different VSCRMs.

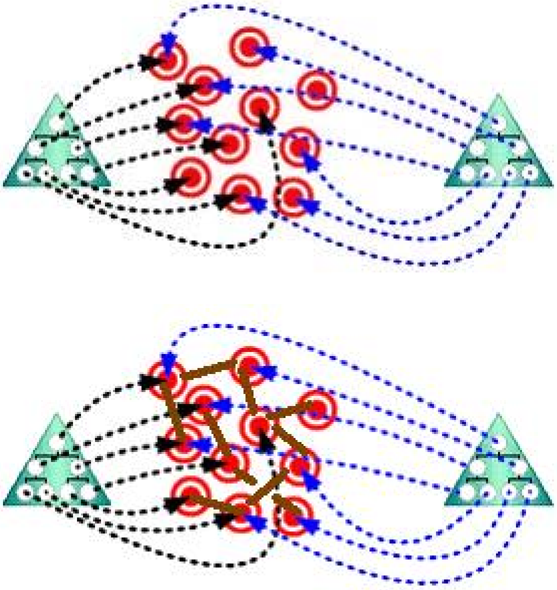


Figure 6.10: Competency profiles and the Comparison Grid

The rules for performing the ontological semantic inferences are as follow; competency A match competency B (these can both be RCDs and VSRCMs) if A can find a semantic link to B using the following ontological rules:

- A equals B
- A synonymous B
- A has part B
- A is more general than B
- A has part C and C satisfies B

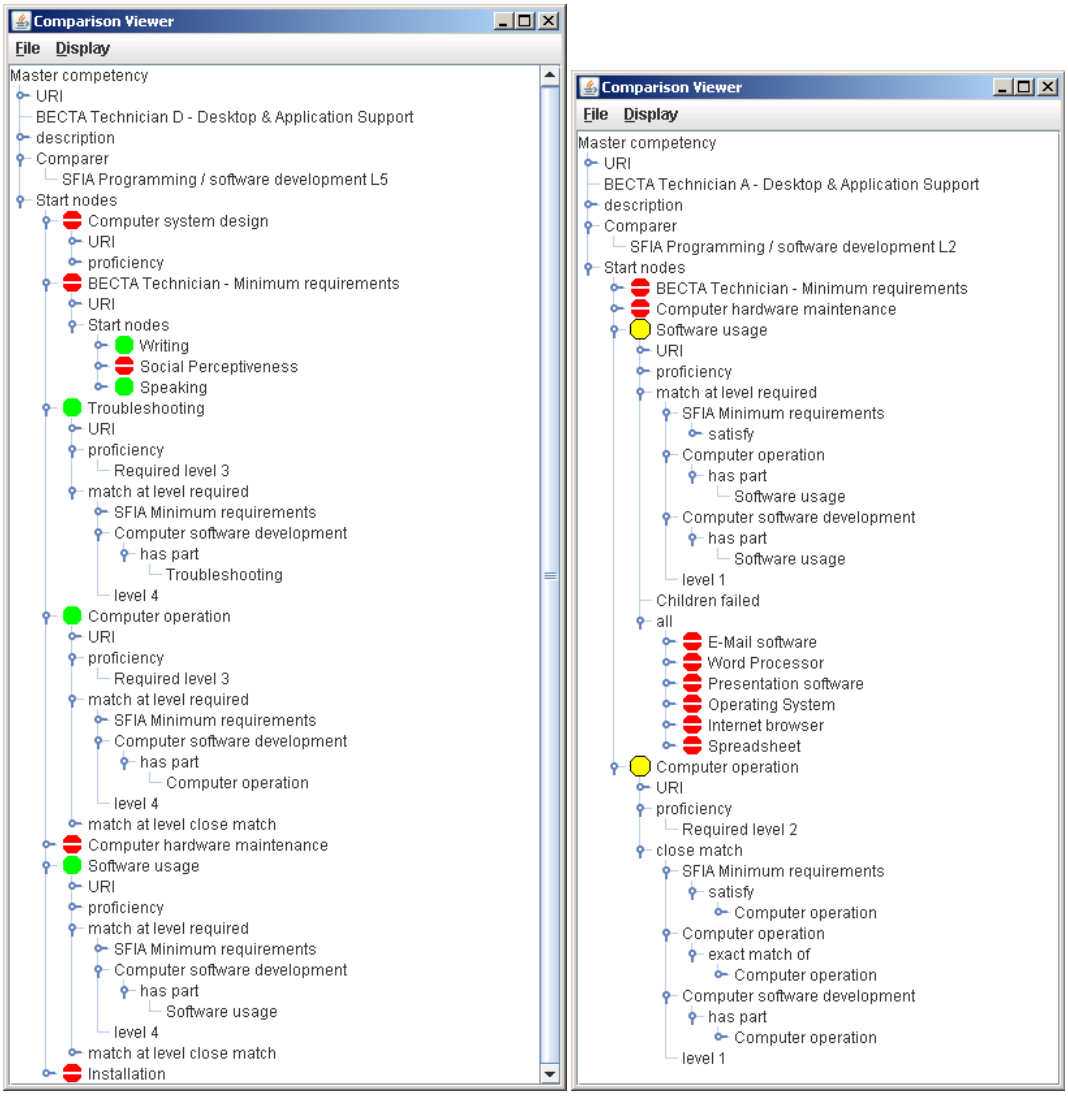
Furthermore competency A indicates a possible match if:

- A is more specific than B
- A is part of B
- A matches B but proficiency levels don’t match

Additionally, the “child” nodes of the node must be matched according to the logical relationship by which they have been related to the node. The following logic relationships are supported by the prototype:

- All: All child nodes must be satisfied (logical and).
- Any: At least one child node must be satisfied (logical or).
- If: (only partly implemented) If the child node is satisfied (using the above rules) then the if_true grandchild node must be satisfied. Otherwise the “grandchild” connected to the child node using the if_false property must be satisfied.

Comparison can now be performed, using the rules, by traversing the nodes of graph, following the logical links from start nodes through to all the child node of the competence profile, trying to find matching nodes in the graph of the other competency profile. Figure 6.11 shows examples of such comparisons from the Competency Suite (above). The green icon indicates a complete match, yellow a possible or close match and red no match.



```
Comparison Viewer
File  Display
Master competency
  URI
    BECTA Technician D - Desktop & Application Support
  description
  Comparer
    SFIA Programming / software development L5
  Start nodes
    ⛔ Computer system design
      URI
      proficiency
    ⛔ BECTA Technician - Minimum requirements
      URI
      Start nodes
        🟢 Writing
        ⛔ Social Perceptiveness
        🟢 Speaking
    🟢 Troubleshooting
      URI
      proficiency
        Required level 3
      match at level required
        SFIA Minimum requirements
        Computer software development
          has part
            Troubleshooting
        level 4
    🟢 Computer operation
      URI
      proficiency
        Required level 3
      match at level required
        SFIA Minimum requirements
        Computer software development
          has part
            Computer operation
        level 4
      match at level close match
    ⛔ Computer hardware maintenance
    🟢 Software usage
      URI
      proficiency
      match at level required
        SFIA Minimum requirements
        Computer software development
          has part
            Software usage
        level 4
      match at level close match
    ⛔ Installation
```

```
Comparison Viewer
File  Display
Master competency
  URI
    BECTA Technician A - Desktop & Application Support
  description
  Comparer
    SFIA Programming / software development L2
  Start nodes
    ⛔ BECTA Technician - Minimum requirements
    ⛔ Computer hardware maintenance
    🟡 Software usage
      URI
      proficiency
      match at level required
        SFIA Minimum requirements
          satisfy
        Computer operation
          has part
            Software usage
        Computer software development
          has part
            Software usage
        level 1
      Children failed
      all
        ⛔ E-Mail software
        ⛔ Word Processor
        ⛔ Presentation software
        ⛔ Operating System
        ⛔ Internet browser
        ⛔ Spreadsheet
    🟡 Computer operation
      URI
      proficiency
        Required level 2
      close match
        SFIA Minimum requirements
          satisfy
            Computer operation
        Computer operation
          exact match of
            Computer operation
        Computer software development
          has part
            Computer operation
        level 1
```

Figure 6.11: Examples of competency profile comparisons

In figure 6.11 to the left the VSCRM “BECTA Technician D – Desktop & Application Support” (TECH D) has been compared with the VSRCM “SFIA Programming / software development L5” (L5). The comparison can be verified by looking at each individual node of the VRSCM graph (on the left). The comparison started at each starting node of TECH D trying to find matches within L5. If a start node is a VSCRM (like the first start node) this profile is also traversed. Therefore this start node is not compared to satisfaction because ‘Social Perceptiveness’ is not part of L5. On the other hand the node which has the competency “Troubleshooting” attached has been matched by the SFIA node with the competency “Computer software development”, because “Computer software development” has been ontological related with the relationship “has part” to “Troubleshooting” in the comparison graph.

To the right is the result of comparing “BECTA Technician A – Desktop & Application Support” with “SFIA Programming / software development L2”.

Note that the comparison process obviously does not changed the actual graphs of the different competency profiles, but merely returns the results of the comparison of each node within the graph being compared.

6.6 Managing the problem of Vague Domains

The problem of vagueness in the competency domain was highlighted in chapter 5. The TRACE application is a system that allows a single user or organisation to manage these issues of differences between competency frameworks, differences in definitions of the competency domains and levels of proficiency, and differences in usage of competency descriptions. This is a major step in the direction of overcoming the problems of vagueness within the competency domain. However it is developed using ontology representation technology, and therefore does not remove the issues completely but enables users to manage them through a stack of small ontological commitments. At the core is the model that enables the description of competencies using the standards of competency as defined by IEEE (or in ontological terms the upper ontology). The next ontological level is the first application ontology using O*NET which was used as a referential starting point, and this is also included as an upper ontology within the system but is likely to be replaced at a later stage after a thorough industrial / standards organisation lead standardisation process has finalised. By using the standards and referencing the O*NET competencies new models can be built. These models can easily be exchanged in and out by the user, thus enabling the user to take out knowledge or change knowledge with which they disagree, thus allowing them to commit to models with which they agree (as described in section 6.4 Implementation of the System).

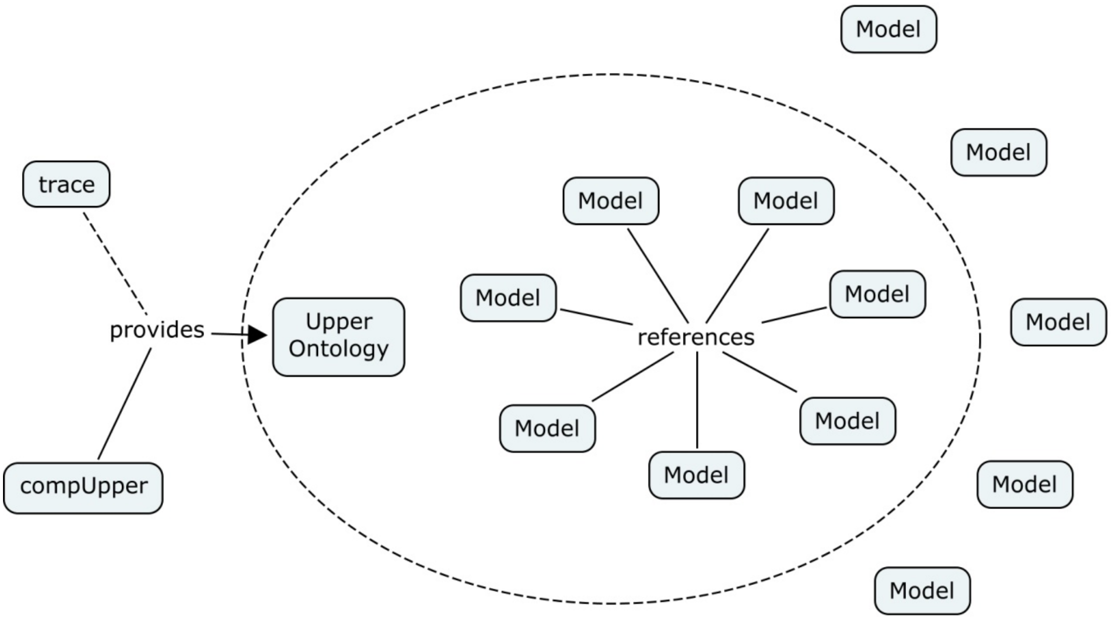


Figure 6.12 Managing vagueness

The commitment to the ontological upper level competency model is necessary in the developed application in order to show the practical application of the suite. The reference starting point used (E*NET) with the strong ties to the U.S. O*NET might be a commitment too far for some users, and it is possible to exchange this model with another starting point and then create semantic relationships using the upper ontology keeping the models intact. This way of coping and managing the vague domain is very similar to the approaches proposed within the semantic web, where each website should have coded up the semantic relationships to several other ontologies. It is however easier to handle within this application, as the standards create a basis on which people can commit to one single upper ontology. There might be problems using the tool if doubts or disagreements arise as to which models should be used in a single organisation. This is obviously true, because the application helps individuals and organisations to manage the vague domain of competencies within what they need to do, it does not solve the problem, and somebody (within the organisation or the individual user) needs to decide how models are created and which models are used within an organisation.

6.7 Future of Competency System

There are several issues, which have been listed throughout this chapter that needs future work in order for the system to gain widespread take up.

- Further design of the upper ontology.

  The standardisation process of competency is not finalised, and this must happen before the upper ontology can be finalised.

- Commitment / re-development of ontological starting point.

  The starting point (E*NET) based on O*NET either needs to be committed to by the user base, or a process has to be started where Europe creates its own O*NET equivalent.

- Community of Practice [114] around the system.

  Several of the processes, especially the knowledge engineering and development of competency descriptions, needs to be further examined. There are some communities who are already performing similar examinations in the world of work (for instance psychologists), but the best practice of creating ontological competency descriptions are non-existent. These need to be developed for wider take up of ontologically based competency systems.

- Large number of users needed.

  The more users the better the ontologies would become. An ontology which is only used by one entity has a very limited value. It is not, for instance, validated by a diverse user base with different purposes of usage, which is a big part of ontology development. Following Gruber's recommendations, the (limited) user base would never know if the ontology would suite many different agent's commitment, thus having a portable semantic value for different applications. The more users who commit to an ontology, the more likely it is that others can make the same commitment.

The future of this application could be a solution which would allow folksonomological reification of data through the gathering of data from many competency contributors. This could be developed by merging the method presented in this chapter with the chatbot reified ontology methods of chapter 4. For instance if organisations created models of competency within the IT domain, these could be reified with statements defining who made them. With many different organisations doing this, it would build up a grand competency framework with reified viewpoint of each organisation. Users could then look at the different viewpoints within the reified ontology and make decisions about appropriateness of semantic links based on clusters of viewpoints rather than a single viewpoint of the domain.

6.8 Summary

In this chapter a solution was explained for making comparisons of competencies based on standards enabling comparison through utilisation of a network of ontologies.

This comprised of several upper ontologies formalising competency standards and competency components such as “knowledge, skills and others”, which could be used to describe domain specific competencies. The system works by allowing small pieces of knowledge in the form of ontologies to be swapped in and out of the system, enabling comparisons that fit the purpose of the user. This was tested on several competencies and comparisons between them were performed. Furthermore a “relaxed” logic, linguistic logic, was utilised to make semantic links between the different competencies within the system. This supported the comparison engine (inference engine) in making semantic links in the vague domain providing non-binary results. The system is extendable through an API. The chapter concluded that the methodologies used within the chatbot tool could be used to enhance the competency system in the future.

7 Future developments in Vague Domains

This chapter will discuss how the techniques provided in the previous chapters could be combined and modified to cope better with the description and inference of semantic information in vague domains, especially by focussing on possible further developments in the vague domains which are the subject of the previous chapters. Each section will discuss this from different viewpoints, first discussing how competency work could be enriched by further work in this field, then a discussion of how social networks could benefit from further use of a reified layer of ontologies, and then ending by examine how chatbots could be further enriched and used both in competency description processes and in more general ontology work.

7.1 Folksonomy of Competency

The greatest problem with the Competency Suite solution described in chapter 6 is that it is providing a solution for managing the vague domain of competencies for a single user or single organisations perspective. It provides a solution where users are able to share competency models and change the knowledge contained in them as necessary only from the single perspective. It would, however, be much more powerful if the knowledge could "grow" incrementally (in very small steps) out of the work of individuals by utilising the power of the Folksonomy as described in section 3.4. Folksonomies combined with competencies, as was suggested in the previous chapter as an approach which could also provide several viewpoints of inference within the same ontology models. This has been suggested by several individuals in the IEEE Competency standardisation working group [115] and the e-portfolio standardisation work under the JISC [116].

Technologically this would be feasible by implementing a reified layer (chapter 4) on the competency system (chapter 6), thus enabling the data model to reference;

- who made what competency and when.
- who “had” specific competencies.
- who created the semantic links between two competencies.
- etc.

This meta-knowledge could then be used in the inferencing engine to provide the different views of the people using the folksonomy, this will be further discussed in the section 7.2. The following sections will provide two possible ways of implementing and organising such systems and discuss some of the implications.

7.1.1 e-Portfolio

e-Portfolios were introduced in section 5.2.1. They can be defined as "a collection of authentic and diverse evidence, drawn from a larger archive, that represents what a person or organization has learned over time, on which the person or organization has reflected, designed for presentation to one or more audiences for a particular rhetorical purpose." [102]. e-Portfolios are traditionally used in education to support learners in evidencing and reflecting on their learning. Some companies are also using them to support their employees in their growth and lifelong learning, for instance Nedcar in the Netherlands [117]. It is also apparent that they are being used to some extent to describe and list the competencies of their owners. There are many different kinds of e-Portfolio systems; they can be individuals writing a blog or maintaining a private homepage, through to multiuser systems with facilities for sharing information between individuals and groups. Although many single user systems in theory can be used to gather competencies it is probably more appropriate to implement them within the multiuser systems. If provisions are made where users can describe and reflect upon competencies and skills that they have or would like to have, in a structured way, these could then be formalised and re-purposed into RCDs and competency mappings and this would, in a multiuser environment, constitute a folksonomy of competencies.

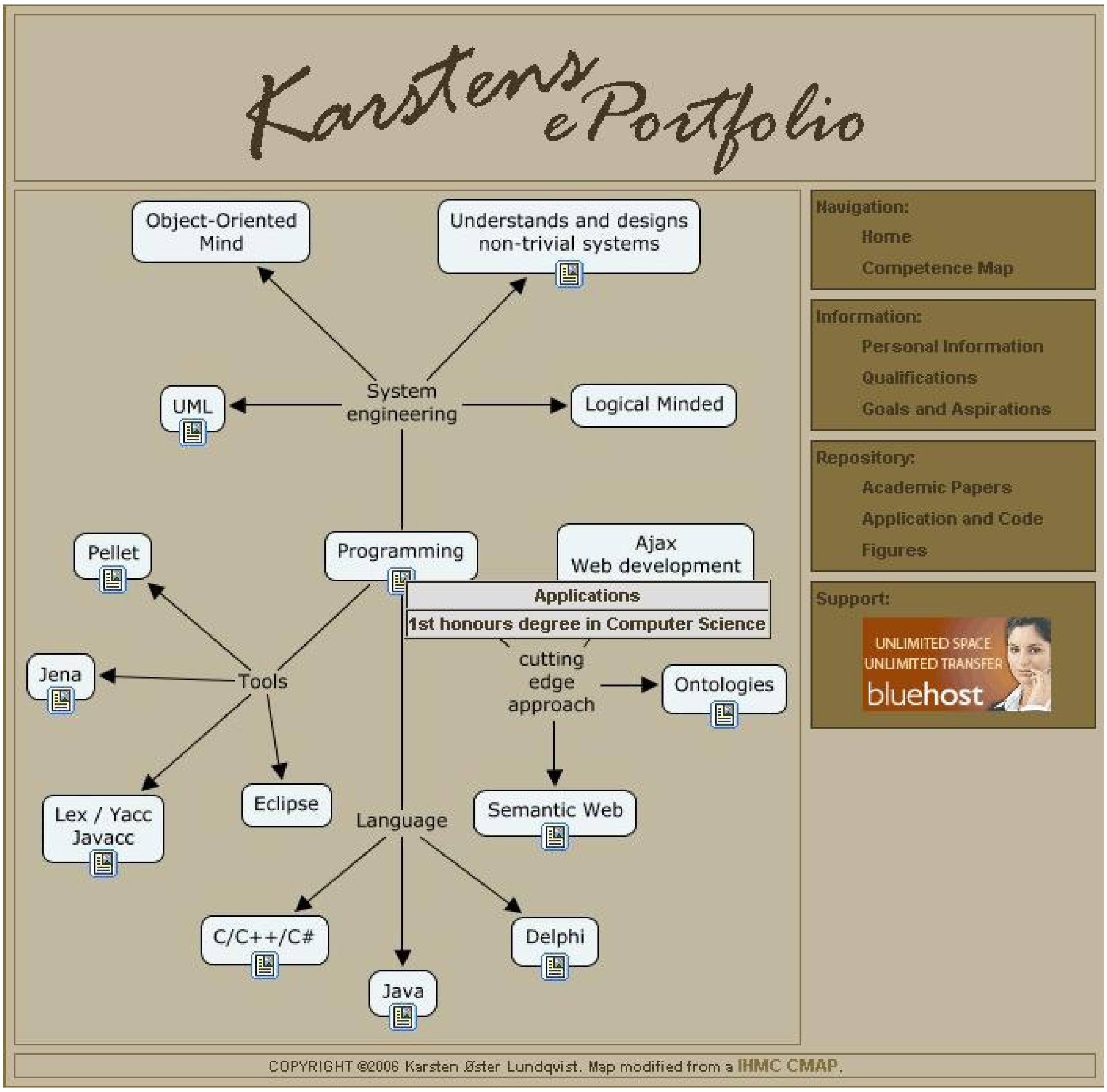
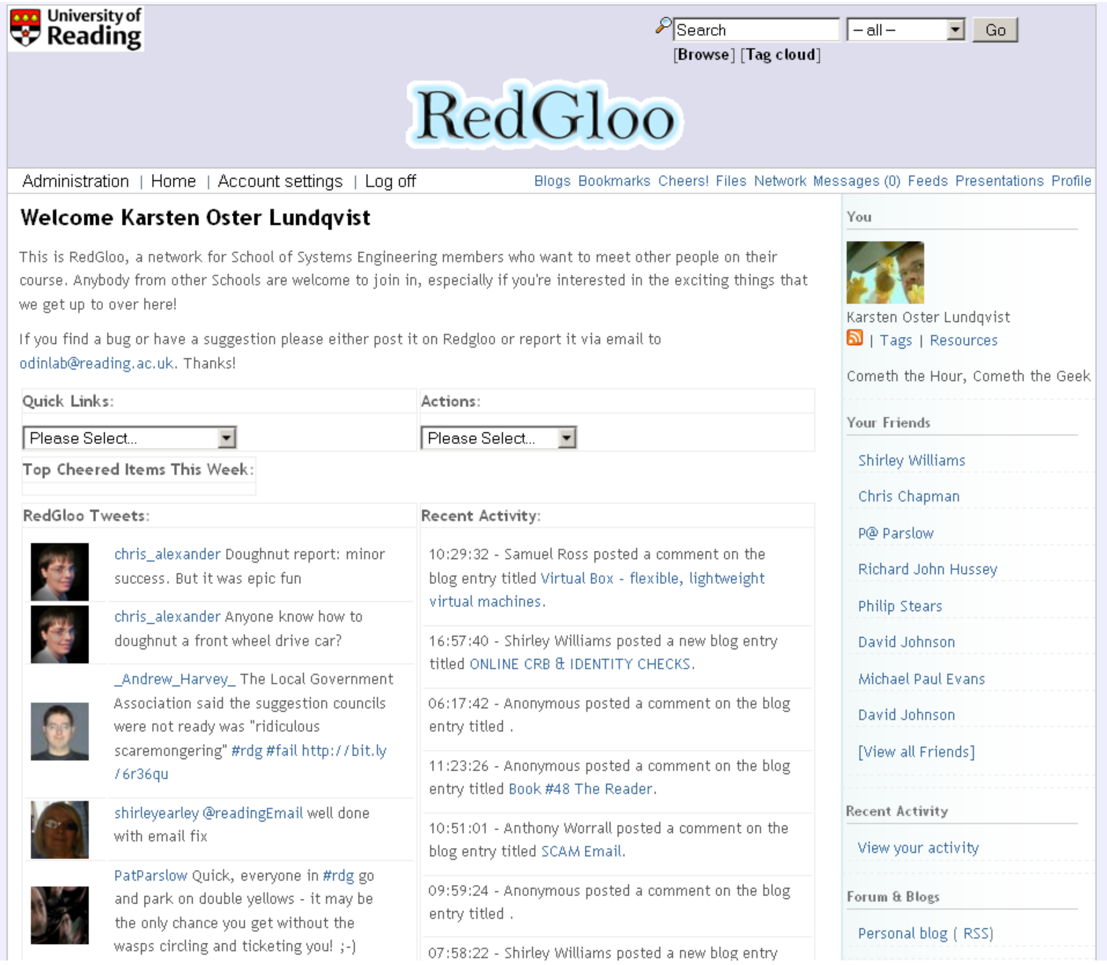


Figure 7.1 A personal competency mapping (incomplete)

Figure 7.1 is a bespoke (incomplete) competency mapping created by the author, which shows how a tool could be made where competencies could be built and interrelated by an individual learner. Within a multi-user system it would be possible to point to other user’s competency descriptions in a multiuser environment, thus create links between competencies which would form the basis of a folksonomy of competencies. This would be possible with a reified layer of semantics introduced in chapter 4.

7.1.2 Social Networking

Several educational organisations have recently started to focus on the use of social networking sites and services in their education environment, for instance by providing students with sites based on ELGG [118], [119], [120]. There are also indications that commercial social networks such as Facebook [121] are being used by students in their learning [122]. Social networking sites have in the past been the source of several folksonomies, for instance by creating tag clouds from individual tags and searchable links between users.



Welcome Karsten Oster Lundqvist

This is RedGloo, a network for School of Systems Engineering members who want to meet other people on their course. Anybody from other Schools are welcome to join in, especially if you're interested in the exciting things that we get up to over here!

If you find a bug or have a suggestion please either post it on Redgloo or report it via email to odinlab@reading.ac.uk. Thanks!

chris_alexander Doughnut report: minor success. But it was epic fun

chris_alexander Anyone know how to doughnut a front wheel drive car?

_Andrew_Harvey_ The Local Government Association said the suggestion councils were not ready was "ridiculous scaremongering" #rdg #fail http://bit.ly/6r36qu

shirleyearley @readingEmail well done with email fix

PatParslow Quick, everyone in #rdg go and park on double yellows - it may be the only chance you get without the wasps circling and ticketing you! ;-)

10:29:32 - Samuel Ross posted a comment on the blog entry titled Virtual Box - flexible, lightweight virtual machines.

16:57:40 - Shirley Williams posted a new blog entry titled ONLINE CRB & IDENTITY CHECKS.

06:17:42 - Anonymous posted a comment on the blog entry titled .

11:23:26 - Anonymous posted a comment on the blog entry titled Book #48 The Reader.

10:51:01 - Anthony Worrall posted a comment on the blog entry titled SCAM Email.

09:59:24 - Anonymous posted a comment on the blog entry titled .

07:58:22 - Shirley Williams posted a new blog entry

Karsten Oster Lundqvist
| Tags | Resources

Cometh the Hour, Cometh the Geek

Shirley Williams
Chris Chapman
P@ Parslow
Richard John Hussey
Philip Stears
David Johnson
Michael Paul Evans
David Johnson
[View all Friends]

Figure 7.2 Example of a Social Network for Learning (RedGloo)

Potentially this could also happen with competencies. The Competency Suite with an extended view like the tool in figure 7.1 could be integrated into a social network, for instance by using a plug-in system as found in Facebook, or by implementing a bespoke module in the educational open source social network system ELGG system. A tool of this type would benefit from the connections that users had already formed, such as friendships, groups, modules, communities, and users could create links between the different competencies of other users. They could give each other competencies or provide evidences for competencies that each other had acquired through assignments, reflections, activities, learning outcomes, etc. These evidences could then in turn form the basis of the creation of new competencies which would then provide more semantic knowledge in the folksonomy.

In a social network based in an educational setting the users could even suggest modules taking learning outcomes based on competencies into account. These would be modified and become a folksonomy of learning modules as more students and lecturers would be creating and providing competencies. To allow granularity of the viewpoints and the inferences made within this folksonomy the reified layer from section 4.4.1 should be introduced. Obviously such a system could be targeted by malicious users, and is therefore only really appropriate in closed educational systems.

7.1.3 Issues

The biggest obstacle to creating a folksonomy is how to get many individual users to use the tools, thus getting sufficient content that will eventually form the folksonomy. It is self-evident that a folksonomy is worth little if only a few people use the tool, and that it will gain more value every time somebody uses it. This issue is probably the largest when considering social networking integration. e-Portfolios are often incorporated into the learning environment in educational institutions, and therefore the users of such systems are either forced to use them or highly motivated by the prospect of good marks. This is not the case for social networks, unless they are part of the formal evaluation process. This could also be the case for institutionally run social networks, but it is unlikely the case for a plug-in into something like Facebook.

The system would have to be cleverly designed so that the users could immediately see the benefit of interacting, because most users of Facebook use it in their own free time and view it purely as an entertainment or spare time activity, even if they actually use it for their personal learning as well. Furthermore the users would have to be convinced that the information they provide would be owned by them and treated ethically as it could potentially involve private information. These are all issues that could make it hard to integrate competency gathering into sites like Facebook.

Ethical issues are probably always going to be an issue with automated competency creation in this way. For instance who owns the information? Can the organisation running the e-Portfolio system or the social network gather the information and reproduce it in a folksonomy? Is it appropriate if the data is anonymised or can inferences be done on all the data provided?

The value of the data within of the folksonomy might also depend on these questions. For instance, if users of the system looked at it as a chore, or as a system which conflicted with their human rights, would the information that they gave to it be wrong or skewed? Additionally what would the value of the folksonomy created by students be compared to one created by experts? There is obviously great value in knowledge created by a multitude of users, but will the value be higher than the knowledge created by experts? Some knowledge probably would be more valuable by the “folksonomy process” than other knowledge. These are all questions that future designers of this kind of system would have to consider in the design.

7.2 Application of the Reified Layer

The reified layer of a reified ontology as discussed in section 4.4.1 provides a powerful possibility for making inferences in applications, because the semantics of a reified ontology use normal semantic web tools to represent these relationships. It is potentially possible to make these inferences spanning over many different tools and services.

The meAggregator project [123] was a JISC funded project which developed a toolset for aggregating content drawn from disparate sources. At the core of this toolset is a tag based filesystem termed the folksonomical filesystem (FFS). The FFS is implemented as a reified ontology with a reified layer, and it will therefore become a system which can be used to perform folksonomical reification (as described in section 3.4) and make inferences based on the reified ontology. Speculatively it could be possible to enable serendipity between different users of the system through comparing the FFS of one user with the FFS of other users. This is all made possible through the usage of the reified layer.

A reified ontology is needed in this space, because it facilitates referencing across multiple applications, vague domains by multiple users at the same time, an ability which is increasingly important on the internet, but is not supported by traditional ontology tools. This can be illustrated just by focussing on social networking sites and services. A vast number of these sites have been developed in recent years, for instance; Facebook [121], Myspace [124], Livejournal [125], Orkut [126], Bebo [127] etc. There are even tools and services available to make the creation of social networks easy, e.g. Ning [128], Elgg [129] and Insoshi [130]. This creates difficulties for the users of these sites. If Petra has friends in many sites, then she probably needs to be on all of them. She has to give all her information and upload avatar pictures each time she signs up for a new service. If she posts something in Facebook and also wants it to be posted on Bebo and Myspace, then she has to go to each site and manually post it in each service.

Some standards, such as OpenSocial [131] and Open Data Definition [132], have been developed to rectify these problems, however the problems still persist because not all services who are willing to make use of these common standards. This can be anticipated, as seemingly several of the services survive because they tie in their user base, and because they do not want to use commercial standards like Google's OpenSocial [133]. The FFS could be used to reference and share the location of important content. By creating an appropriate plug-in, either by using the industry standards or the Application Programming Interfaces (API) of the services, we could then push and pull the information between the services, creating a meta-layer of services between all the sites. Each time a user did this, the folksonomy of shared content on social networks would grow and would facilitate the creation of a reified ontology of content in social networks. This picture is true for many different services on the internet, such as learning management systems, e-Portfolios and job boards. It would possibly benefit the ability to draw inferences over the complete system. For instance inferences similar to those developed in systems used by Amazon [134] that are able to provide extra information for its customers, when they say “Customers who bought this book also bought...” The system proposed here would potentially be able to say that “People like you would normally like this web page.”, or “Students who study a similar subject to yours, which we think is modern guitar music, would normally use Bebo for their social interactions, and find jobs through their own online portfolios.”

7.3 Chatbots as ontology creation tools

The creation of standard ontologies has traditionally involved an ontology engineer who would implement the ontology based on information provided by a domain expert or a group of domain experts. This process has already been discussed and criticised earlier for being error-prone and costly. Hence it is an area of academic and corporate interest to find tools and processes that would automate the creation of ontologies. There are also efforts to do this automatically from text [135] and webpages [136].

In the chatbot example in chapter 4 the author has created several starting points in the form of ontologies, which provided the domain knowledge for the survey-chatbot to construct the questions for the interviewees. This approach still requires an ontology engineer as part of the development process. It would be feasible to devise a system based on a chatbot using OwlLang, which could reduce the need for the ontology engineer. If a system were to be implemented such that the chat structures, (which were previously based on the design of a survey chatbot, Chapter 4), still retained a survey functionality, but with the added aim of knowledge extraction, then users could be the sources of knowledge. This approach is similar to that used by an ontology engineer inquiring of the ‘domain expert’. The starting ontology would be considered to have no domain knowledge, but merely comprised of an upper ontology of logics and very basic concepts, then by initiating questions to the chatbot user (preferably a ‘domain expert’) could be used to build knowledge of the domain.

Furthermore this system could be used to ask several domain experts, which might be in disagreement, and due to the nature of the reified ontology these disagreements would all be contained in the knowledge base and available for inferencing. Obviously the system would have to “know” how to initiate the questioning of the experts and provide measures so that previous statements would be criticised by other experts. A system like this could be an interesting supplement in the toolset of semi-automated ontology design process, yet it is merely extending the existing functionality of the chatbot found in chapter 4.

7.3.1 Chatbots and Competencies

Another possible application would be to develop competency gathering chatbots. The chatbot can be used as described in the previous section, if slightly modified, to create competency frameworks for different domains from scratch. The “expressiveness” of the competency framework would depend on the number and quality of experts available in the process, but also what kind of questions were available in the chat structures of the chatbot. For instance, by including questions like; “How would an individual obtain this knowledge?”, “What measurement is used to establish this skill?” or “What questions could you ask to find out if I had this competency?” If such questions where included then the resulting knowledge or “competency framework” would include information for validation of the competencies and possible training opportunities designed to allow the users to attain the different levels of competency within the framework.

This could lead to the creation of more advanced chatbots, which by having other specialised chat structures yet at the same time utilising the common knowledge base, could ask new application specific questions. These could for instance be used to;

- interview in employment interviews assessing the competency levels of candidates.
- establish competency levels of employees and students.
- evaluate educational needs and deliver training in organisations.

Obviously further development would be needed to reach this level and sophistication. There would be further development needed on the quality of the chat structures, as we have shown they are prone to misuse and cheating. It is probably true to observe it is not as important when interviewing experts to create a new reified ontology, as it is when using the reified ontology to evaluate employees and students or when interviewing possible candidates. Additionally the actual inferencing on the reified ontologies would need substantial improvements, as the questions asked would need to be consistent even when the vagueness of the domain could lead to contradicting views among the experts. Although if the inference mechanisms became very advanced it could be foreseen that "trick" questions could be developed, where the chatbot utilised the fact that several experts where disagreeing trying to lure the user into loop holes and contradictory answers. In addition it would be necessary to have some kind of validation of the produced ontologies to gain acceptance by the wider user communities. Even though ontologies are readable by humans (contrary to neural networks) they are not easily understood, and broad take up would probably only happen if the ontologies were accredited at some level. Ongoing commitment to the underlying ontology would be an important part of an application tool set.

This work would connect the work from the chatbot system, including the reified layer, with the tools found in the Competency Suite, and would be an interesting avenue for the future of this research.

7.4 Summary

The use of the novel methodologies developed in chapter 4 and 6 within future developments have been discussed in this chapter. It focussed on how future developments could improve the domains of vague domains specifically by investigating the case study domains discussed previously in the thesis.

It was concluded that ePortfolios, social networks and application of chatbots could improve the usage of the methods developed within the ontological competency system, although there would be ethical and privacy issues in doing so. Furthermore it was described that the reified layer from the chatbot system could be utilised in other applications, this kind of development has already started through the meAggregator project. The ontology chatbot system could also be used to enhance ontology creation and further work with competencies.

8 Conclusions

This thesis has focused on creating technologies and methods for comparing knowledge in vague domains. This has been achieved by using standard ontology tools, utilised in non-traditional ways.

Within the domain of conversational agents a novel approach was successfully developed. It allows full access to the knowledgebase contained within available ontologies. This is achieved through the scripting language OwlLang, which has been developed for this purpose. This language also enables storage of new knowledge acquired through chat within the ontology, therefore enabling a growing "world knowledge" which can be used in new chat instances. While this happens new statements might be introduced into the ontologies of the chatbot with contradictions of the existing knowledge contained within them. A novel solution utilising heavy usage of reified statements (statements about statements) has been created, which creates a "reified layer" of semantics within ontologies, which can used to make inferences even when disagreeing statements are present in the ontology therefore allow the inconsistencies that might arise from the added contradictions.

This has enabled the development of an ontology driven chatbot methodology, where chat structures (trees of chat) can be been developed, as is an often used approach when developing chatbots, integrating OwlLang statements, thereby inserting knowledge from the ontology at runtime. The benefit of this is that a generic chatbot can been developed for a certain situation and by changing only the underlying ontologies the same chatbot can respond and converse about issues within separate domains. A generic survey chatbot has been developed and two different domain ontologies have been used to test the ability of rapid domain changes of the survey chatbot.

The second domain that was used within this thesis is the domain of competency and competency frameworks. As chapter 5 showed, this is a domain where even the definition of the words is disputed. Many different competency frameworks exists which only add to the vagueness of the domain, which makes it difficult to compare and contrast different competencies and competency frameworks. A novel approach was developed which enable "relaxed" semantic comparisons with non-binary results. To achieve this, a bespoke standard Very Simple Reusable Competency Mapping (VSRCM) was developed based on the idea of competency mappings [5]. By using the VSRCM and the simple industry standard Reusable Competency Definition (RCD) competencies was described and represented. The graph contained within the VSRCM allows the traversal of the competencies, thus enabling comparisons between them using the logical relationships inherent in the connections of the graph, furthermore linguistic semantics are introduced allowing comparisons of RCDs based on a less stringent logical foundation. A comparison engine was created which uses these features to compare different competencies, exploring whether they match each other's different competency components and proficiency levels. The approach to handling the vagueness of the domain was achieved through the usage of small ontologies which can be chosen based on appropriateness of the situation they are being used. An Application Programming Interface (API) has been created which 3rd parties can use to develop new services based on the underlying competency representation and comparison system.

The utilisation of a reified layer of semantic information has been shown to increase the expressiveness of standard ontologies, providing a novel approach to ontology specification where different points of views can be present side by side, yet preserving the semantic connections of the statements allowing inferences to be made consistently. The standard inferencing engines can, however, not perform these inferences, thus more work is needed in this area. The work in the competence domain have shown that an approach where small ontologies can be brought in and out of "scope" when performing the inference can generate a system that allows inferences which suites the user in the situation they find themselves. This resembles the methods of the Semantic Web where different ontologies are used based on which websites the users are concerned with. The major difference is that the "price" of mistakes in the competency domain potentially is higher (e.g. a HR department employs the wrong employee) and therefore the person using the system needs to be aware of what each small ontology states and make conscious choices of which ontologies to use. This work would, just like the Semantic Web, benefit from the usage of digital signatures, which combined with a reified layer could become a powerful methodology for automated inferencing.

There are many applications, a few discussed in chapter 7, that could benefit from better inferencing, but which at present are not doing this due to the vague nature of the domains they work in. Being able to express diversity and make meaningful inferences within the vagueness of this diversity will become more and more important as rich data sources are being developed, shared and utilised between disparate applications. This thesis demonstrates successfully that traditional ontology tools can be utilised in new ways in order to accommodate for comparisons in domains with diverging world views.

References

[1] Richard Wallace. (2006, May) AIML Overview. [Online]. Available: http://www.pandorabots.com/pandora/pics/wallaceaimltutorial.html

[2] A.L.I.C.E. AI Foundation. (2007, December) ALICEBOT. [Online]. Available: http://www.alicebot.org/

[3] Pandorabots. (2007, December) Pandorabots. [Online]. Available: http://www.pandorabots.com/botmaster/en/home

[4] TRACE. (2009, Mar.) TRACE. [Online]. Available: http://trace.education-observatories.net/

[5] Claude Ostyn. (2007, November) Proposed Draft Standard for Learning Technology - Simple Reusable Competency Map. [Online]. Available: http://www.ostyn.com/standardswork/competency/ReusableCompMapProp.pdf

[6] Tim Berners-Lee. (2007, December) Semantic Web Roadmap. [Online]. Available: http://www.w3.org/DesignIssues/Semantic.html

[7] Nicola Guarino, "Formal Ontology and Information Systems," in *Formal Ontology in Information Systems*, Amsterdam, 1998, pp. 3-15.

[8] Thomas R. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," *International Journal of Human-Computer Studies*, vol. 43, no. 5-6, pp. 907-928, 1995.

[9] Thomas R. Gruber, "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, pp. 199-220, 1993.

[10] Nicola Guarino and Pierdaniel Giaretta, "Ontology and Knowledgebases - Towards a Terminological Clarification," in *Towards Very Large Knowledge Bases - Knowledge Building and Knowledge Sharing*. Amsterdam: IOS Press, 1995, pp. 25-32.

[11] Aristotle, "Metaphysics," in *Book IV.*, 350 B.C.

[12] Michael R. Genesereth and Nils J. Nillson, *Logical Foundations of Artificial Intellingence*. Los Altos: Morgan Kaufmann Publishers Inc., 1986.

[13] Metaphysics Research Lab, CSLI, Stanford University. (2007, December) Stanford Encyclopedia of Philosophy. [Online]. Available: http://plato.stanford.edu/entries/logic-intensional/

[14] Philippe Martin. (2007, December) Knowledge Representation/Translation in RDF+OWL, N3, KIF, UML and the WebKB-2 languages (For-Links, Frame-CG, Formalized English). [Online]. Available: http://www.webkb.org/doc/model/comparisons.html

[15] W3C. (2007, December) OWL Web Ontology Language - Guide. [Online]. Available: http://www.w3.org/TR/owl-guide/

[16] Tim Berners-Lee. (2007, December) Primer: Getting into RDF & Semantic Web using N3. [Online]. Available: http://www.w3.org/2000/10/swap/Primer.html

[17] W3C. (2007, December) W3C Semantic Web Activity. [Online]. Available: http://www.w3.org/2001/sw/

[18] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen, "From SHIQ and RDF to OWL: The Making of a Web Ontology Language," *Journal of Web Semantics*, vol. 1, no. 1, pp. 7-26, 2003.

[19] W3C. (2007, December) OWL Web Ontology Language - Overview. [Online]. Available: http://www.w3.org/TR/owl-features/

[20] W3C. (2007, December) SPARQL Query Language for RDF. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[21] W3C. (2009, Oct.) W3C Semantic Web. [Online]. Available: http://www.w3.org/blog/SW/2008/01/15/sparql_is_a_recommendation

[22] Leigh Dodds. (2007, December) Introducing SPARQL: Querying the Semantic Web. [Online]. Available: http://www.xml.com/pub/a/2005/11/16/introducing-sparql-querying-semantic-web-tutorial.html

[23] Tim Berners-Lee. (2007, December) Rules and Formulae. [Online]. Available: http://www.w3.org/2000/10/swap/doc/Rules

[24] Aduna. (2007, December) openRDF. [Online]. Available: http://www.openrdf.org/

[25] Dave Beckett. (2007, December) Redland RDF Libraries. [Online]. Available: http://librdf.org/index.html

[26] Jena. (2007, November) Jena Semantic Web Framework. [Online]. Available: http://jena.sourceforge.net/

[27] Sean Bechhofer, Ralf Moller, and Peter Crowther, "The DIG Description Logic Interface," in *International Workshop on Description Logics*, 2003.

[28] Hewlett-Packard Development Company. (2007, December) Joseki - A SPARQL Server for Jena. [Online]. Available: http://www.joseki.org/

[29] Tim Berners-Lee. (2007, December) CWM. [Online]. Available: http://www.w3.org/2000/10/swap/doc/cwm.html

[30] Stanford Center for Biomedical Informatics Research. (2007, December) Welcome to protégé. [Online]. Available: http://protege.stanford.edu/

[31] Revelytix, Inc. (2007, December) Welcome to Knoodl! [Online]. Available: http://www.knoodl.com/ui/home.html

[32] BBN Technologies. (2007, December) An OWL Mode for XEmacs and GNU Emacs. [Online]. Available: http://owl-emacs.projects.semwebcentral.org/

[33] Racer Systems GmbH. (2007, December) RacerPro. [Online]. Available: http://www.racer-systems.com/

[34] Clark & Parsia LLC. (2007, December) Pellet: The Open Source OWL-DL Reasoner. [Online]. Available: http://pellet.owldl.com/

[35] Ian Horrocks and Dmitry Tsarkov. (2007, December) FaCT++. [Online]. Available: http://owl.man.ac.uk/factplusplus/

[36] IHTSDO. (2007, December) International Health Terminology Standards Development Organisation. [Online]. Available: http://www.ihtsdo.org/

[37] National Centers for Biomedical Computing. (2007, December) Open Biomedical Ontologies. [Online]. Available: http://obofoundry.org/

[38] NASA. (2007, December) Semantic Web for Earth and Environmental Terminology. [Online]. Available: http://sweet.jpl.nasa.gov/index.html

[39] Kurt Godden. (2007, December) The General Motors Variation-Reduction Adviser. [Online]. Available: http://www.allbusiness.com/professional-scientific/computer-systems-design-related/1102686-1.html

[40] OpenCyc.org. (2007, December) OpenCyc.org. [Online]. Available: http://www.opencyc.org/

[41] W3C. (2007, December) Wordnet in RDFS and OWL. [Online]. Available: http://www.w3.org/2001/sw/BestPractices/WNET/wordnet-sw-20040713.html

[42] Barry Smith, "Beyond Concepts: Ontology as Reality Representation," in *Formal Ontology and Information Systems 2004*, Amsterdam, 2004, pp. 73-84.

[43] Fabian Neuhaus, Pierre Grenon, and Barry Smith, "A Formal Theory of Substances, Qualities, and Universals," in *Formal Ontology in InformationSystems (FOIS 04)*, Turin, 2004, pp. 49-59.

[44] David Parry, "A fuzzy ontology for medical document retrieval," in *ACM International Conference Proceeding Series; Vol. 54*, Darlinghurst, 2004, pp. 121 - 126.

[45] Cory Doctorow. (2008, July) Metacrap: Putting the torch to seven straw-men of

the meta-utopia. [Online]. Available: http://www.well.com/~doctorow/metacrap.htm#2.5

[46] Collin F. Lynch, Kevin D. Ashley, Vincent Aleven, and Niels Pinkwart, "Defining "Ill-Defined Domains"; A literature survey," in *Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at the 8th International Conference on Intelligent Tutoring Systems*, 2006, pp. 1-10.

[47] BusinessWeek. (2008, July) BusinessWeek.com. [Online]. Available: http://www.businessweek.com/technology/content/apr2007/tc20070409_961951.htm

[48] W3C. (2008, July) W3C Recommandation. [Online]. Available: http://www.w3.org/TR/xmldsig-core/

[49] Larry Loeb. (2008, July) ibm.com. [Online]. Available: http://www.ibm.com/developerworks/xml/library/s-digsig.html

[50] Dan Brickley and Libby Miller. (2008, July) xmlns.com. [Online]. Available: http://xmlns.com/foaf/spec/

[51] F. Dawson and T. Howes. (2008, July) ietf.org. [Online]. Available: http://www.ietf.org/rfc/rfc2426.txt

[52] Harry Halpin, Brian Suda, and Norman Walsh. (2008, July) w3.org. [Online]. Available: http://www.w3.org/2006/vcard/ns

[53] Tim Berners-Lee. (2008, July) w3.org. [Online]. Available: http://www.w3.org/DesignIssues/RDFnot.html

[54] Ben Adida and Mark Birbeck. (2008, December) W3C Working Group Notes. [Online]. Available: http://www.w3.org/TR/xhtml-rdfa-primer/

[55] William Mendenhall, Dennis D. Wackerly, and Richard L. Scheaffer, *Mathematical Statistics with Applications*, 4th ed. Boston: PWS-KENT, 1990.

[56] William Mendenhall, Dennis D. Wackerly, and Richard L. Scheaffer,

Mathematical Statistics with Applications. Boston: PWS-KENT Publishing Company, 1990, pp. 11-12.

[57] Thomas Vander Wal. (2008, July) vanderwal.net. [Online]. Available: http://www.vanderwal.net/folksonomy.html

[58] Gene Smith, *Tagging People-Powered Metadata for the Social Web*. Berkeley: New Riders, 2008.

[59] Tim O'Reilly. (2008, July) oreillynet.com. [Online]. Available: http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[60] James Surowiecki, *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*. London: Little Brown, 2004.

[61] Patrick Parslow, Karsten Øster Lundqvist, Edwin Porter-Daniels, Richard Hussey, and Shirley Williams, *Folksonomological Reification*. Reading: OdinLab, 2009, online: http://www.lulu.com/content/paperback-book/folksonomological-reification/6043166.

[62] John Yen and Reza Langari, *Fuzzy Logic. Intelligence, Control, and Information*, 4th ed. Delhi: Pearson Education (Singapore), 2005.

[63] Giorgos Stoilos, Giorgos Stamou, Vasillis Tzouvaras, Jeff Pan, and Ian Horrocks, "Fuzzy OWL: Uncertainty and the Semantic Web," in *International Workshopon OWL: Experience and Directions*, 2006.

[64] Ching-Man Au Yeung and Leung Ho-Fung, "A Formal Model of Ontology for Handling Fuzzy Membership and Typicality of Instances," *The Computer Journal*, vol. 53, no. 3, 2010.

[65] Fernando Bobillo, Miguel Delgardo, and Juan Gomez-Romero, "Crisp Representations and Reasoning for Fuzzy Ontologies," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 17, no. 4, pp. 501-

530, 2009.

[66] Audun Jøsang, "A Logic for Uncertain Probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, pp. 279–311, 2001.

[67] Simon Haykin, *Neural Networks. A Comprehensive Foundation*, 2nd ed. Delhi: Pentice-Hall of India, 2004.

[68] Henry A. Rowley, *Neural Network-Based Face Detection*. Pittsburgh: Carnegie Mellon University, 1999.

[69] AI Research. (2007, December) AI Research - Creating a new form of life. [Online]. Available: http://a-i.com/

[70] Jason L. Hutchens, "Conversing with Stochastic Language Models," in *Third Workshop on Human-Computer Conversation*, 2000.

[71] Simon Haykin, *Neural Networks. A Comprehensive Foundation*. New Delhi: Prentice-Hall of India, 2004.

[72] Finn V. Jensen, *Introduction to Bayesian Networks*. New York: Springer-Verlag, 1996.

[73] Dictionary.com. (2010, Jan.) Dictionary.com. [Online]. Available: http://dictionary.reference.com/browse/chatbot

[74] James Allen, *Natural Language Understanding*, 2nd ed. Delhi: Pearson Education, 2004.

[75] Python Software Foundation. (2007, December) Python. [Online]. Available: http://www.python.org/

[76] George Kingsley Zipf, *Selected Studies of the Principle of Relative Frequency in Language*. Cambridge, Massachusetts: Harvard University Press, 1932.

[77] Joseph Weizenbaum, "ELIZA--A Computer Program For the Study of Natural Language Communication Between Man and Machine," *Communications of the*

ACM, vol. 9, no. 1, pp. 36-45, January 1966.

[78] Loebner Prize. (2007, December) Loebner Prize. [Online]. Available: http://www.loebner.net/Prizef/loebner-prize.html

[79] Oracle. (2010, May) java.net. [Online]. Available: https://javacc.dev.java.net/

[80] Noel Bush. (2010, May) aitools.org. [Online]. Available: http://aitools.org/Program_D

[81] Davy Temperley, John Lafferty, and Daniel Sleator. (2010, May) AbiWord. [Online]. Available: http://www.abisource.com/projects/link-grammar/

[82] Karsten Lundqvist, Shirley Williams, and Keith Baker, "Evaluation of Quality of e-Learning Environments," in *55h European Conference on e-Learning*, Winchester, 2006, pp. 216-223.

[83] Menon Network. (2010, May) SEEQUEL. [Online]. Available: http://thor.lrf.gr/seequel/index

[84] Colin Mc Cullough and Virginie Aimard. (2008, July) trainingvillage. [Online]. Available: http://www.trainingvillage.gr/etv/Upload/Exchange_views/Surveys/Report_survey_Teachers_and_Learners_and_e-learning_final.pdf

[85] EFQUEL Foundation. (2010, May) EFQUEL. [Online]. Available: http://www.qualityfoundation.org/

[86] Commission of the European Communities, "on the establishment of the European Qualifications Framework for lifelong learning," Brussels, 2006.

[87] Commission of the European Communities, "on key competences for lifelong learning," Brussels, 2005.

[88] HR-XML Consortium. (2010, May) HR-XML Consortium. [Online]. Available: http://www.hr-xml.org

[89] Richard S. Mansfield, "Building Competency Models: Approaches for HR Professionals," *Human Resource Management*, vol. 35, no. 1, pp. 7-18, 1996.

[90] Michael L. Brown, Karsten O. Lundqvist, Shirley Williams, and Keith Baker, "ontoReadiness: A meta-Ontology for Readiness Certification and Career Portability," *The European Journal for the Informatics Professional*, vol. 8, no. 3, pp. 56-61, 2007.

[91] Jo Boon and Marcel R. van der Klink, "Competencies: the triumph of a fuzzy concept," *International Journal of Human Resources Development and Management (IJHRDM)*, vol. 3, no. 2, pp. 125-137, 2003.

[92] The Columbia Guide to Standard American English. (2007, December) The Columbia Guide to Standard American English. [Online]. Available: http://www.bartleby.com/68/76/1376.html

[93] Jonathan Winterton, Françoise Delamare - Le Deist, and Emma Stringfellow, "Typology of knowledge, skills and competences: clarification of the concept and prototype," Thessaloniki, 2005.

[94] EIfEL, "European Competency Frameworks - TRACE Work package I Report," Champlost, 2005.

[95] CEN, "European ICT Skills Meta-Framework - State-of-the-Art review,clarification of the realities, and recommendations for next steps," Brussels, 2006.

[96] O*NET. (2007, November) O*NET Online. [Online]. Available: http://online.onetcenter.org/

[97] Claude Ostyn. (2007, December) Competency data standards resources. [Online]. Available: http://www.ostyn.com/rescompetency.htm

[98] Michael Brown, Internal Trace project meetings, 2005-2007.

[99] IEEE, "Learning Technology Data Model for Reusable Competency

Definitions," New York, IEEE 1484.20.1, 2007.

[100] Ed Walker and Robby Robson, RCDEO to enter IEEE Standardization Process, October 3, 2003.

[101] IMS Global Learning Consortium, Inc. (2007, December) IMS Global Learning Consortium. [Online]. Available: http://www.imsglobal.org/competencies/index.html

[102] IMS Global Learning Consortium, Inc. (2007, December) IMS ePortfolio Best Practice and Implementation Guide. [Online]. Available: http://www.imsglobal.org/ep/epv1p0/imsep_bestv1p0.html

[103] Karsten Øster Lundqvist. (2007, December) Karstens ePortfolio. [Online]. Available: http://www.oster-lundqvist.com/karsten/?status=competenceMap

[104] NIST. (2007, December) Dictionary of Algorithms and Data Structures. [Online]. Available: http://www.nist.gov/dads/HTML/directAcycGraph.html

[105] WordNet. WordNet Glossary. [Online]. Available: http://wordnet.princeton.edu/gloss

[106] Karsten Øster Lundqvist, Shirley Williams, and Keith Baker, "Feasibility of the KSAO Approach in ICT Competency Frameworks," Reading, 2006.

[107] Scienter Espana, "Report on Feasibility of the O*NET KSAO Approach in the European Context – Career Guidance Sector," Granada, 2006.

[108] Scienter, "Report on O*NET approach in European context," Bologna, 2006.

[109] EIfEL, "Feasibility of the O*NET KSAO Approach in Functional Analysis-based Competency Frameworks in the domains of eLearning and Key Skills," Champlost, 2006.

[110] BECTA. (2009, Dec.) ICT Competecnes Framework. [Online]. Available: http://www.becta.org.uk/techcomp/docs/ict_competences_framework.pdf

[111] O*NET. (2009, Dec.) O*NET OnLine. [Online]. Available: http://online.onetcenter.org/skills/#group4

[112] Nik F. Zerdahelyi, "JOSE Jobsearch," in *Final Report*. Reading: TRACE Consortium, 2007, p. Appendix 22.

[113] Xin He et al., "A Competency Based Ontology-Driven Framework Using Agent-Oriented Semantic Web Approach," in *Proceedings of the 12th Chinese Automation & Computing Society Conference in the UK*, 2006.

[114] Etienne Wenger, *Communities of practice: Learning, meaning, and identity*. Cambridge: Cambridge University Press, 1999.

[115] Competency Working Group IEEE, "Discussions about competencies and how to utilise them," 2007-2008.

[116] Simon Grant, "Private discussions," 2008.

[117] Theo Mensen and Luc Vervenne. (2009, Mar.) ePortfolio 2006 - Conference papers. [Online]. Available: http://www.eife-l.org/publications/eportfolio/proceedings2/ep06/E1_vervenne.pdf

[118] elgg. (2008, December) elgg. [Online]. Available: http://elgg.org/

[119] Brighton University. (2008, December) community@Brighton. [Online]. Available: http://community.brighton.ac.uk/

[120] Shirley Williams, Karsten Lundqvist, and Keith Baker. (2007, December) Learning Technologies - Proceedings of iLearn 2007. [Online]. Available: http://www.eife-l.org/publications/proceedings/ilf07/LearningLandscape_001.doc.pdf/view

[121] Facebook. (2008, December) Facebook. [Online]. Available: http://www.facebook.com

[122] Pat Parslow, Karsten Øster Lundqvist, Mike Evans, Edwin Porter-Daniels, and Shirley Williams, "eLearning Communities and spaces," in *Internet 9.0:*

Rethinking Community, Rethinking Place, Copenhagen, 2008.

[123] University of Reading, Odinlab. (2008, December) meAggregator. [Online]. Available: http://www.meaggregator.org

[124] Myspace. (2008, December) Myspace. [Online]. Available: http://www.myspace.com/

[125] Livejournal. (2008, December) Livejournal. [Online]. Available: http://www.livejournal.com/

[126] Orkut. (2008, December) Orkut. [Online]. Available: http://www.orkut.com

[127] Bebo. (2008, December) Bebo. [Online]. Available: http://www.bebo.com/

[128] Ning. (2008, December) Ning. [Online]. Available: http://www.ning.com/

[129] Elgg. (2008, December) Elgg. [Online]. Available: http://www.elgg.org/

[130] Insoshi. (2008, December) Insoshi. [Online]. Available: http://portal.insoshi.com/

[131] Google. (2008, December) OpenSocial. [Online]. Available: http://code.google.com/apis/opensocial/

[132] Elgg. (2008, December) Open Data Definition. [Online]. Available: http://www.opendd.net/

[133] Dan Farber. (2008, December) ZDNet. [Online]. Available: http://blogs.zdnet.com/BTL/?p=6813

[134] Amazon. (2008, December) Amazon. [Online]. Available: http://www.amazon.com/

[135] Luiz C. C. Carvalheira and Edson Satoshi Gomi, "A Method for Semi-automatic Creation of Ontologies Based on Texts," in *Advances in Conceptual Modeling – Foundations and Applications*. Berlin: SpringerLink, 2007, pp. 150-

159.

[136] Sigfried Handschuh, Steffen Staab, and Fabio Ciravegna, "S-CREAM — Semi-automatic CREAtion of Metadata," in *Lecture Notes in Computer Science*. Berlin: SpringerLink, 2002, pp. 165-184.

[137] Thomas R. Gruber, "Ontolingua: A Mechanisme to Support Portable Ontologies," Stanford, Technical Report KSL 91-66, 1992.

Bibliography

Authors publications and presentations:

Karsten Lundqvist. (2010), "ASSET: Enhance Feedback Provision using Video", *Presentation Plymouth eLearning* 2010

Karsten Lundqvist. (2009), "ASSET: Moving Forward Through Video Feedback" *Presentation at Online Educa* 2009

Kirsten Oster Lundqvist, Karsten Oster Lundqvist (2009), "Facebook Faith - Social Networking in a Faith Based Community", 180 degree Symposium, *Andrews University Seminary*, Berrien Springs, Michigan, USA

Karsten Lundqvist (2009), "The Challenge of Aggregation", *Presentation Plymouth eLearning* 2009

Shirley Williams, Pat Parslow, Karsten Øster Lundqvist (2008), "Position paper on the Future of Social Networking*", W3C Position paper on the Future of Social Networking*

Pat Parslow, Karsten Øster Lundqvist, Mike Evans, Edwin Porter-Daniels, Shirley Williams (2008), "eLearning Communities and spaces*", Internet 9.0 Copenhagen*

Karsten Øster Lundqvist, Rob Ashton, Mike Evans, Pat Parslow, Edwin Porter-Daniels, Shirley Williams (2008), "MeAggregator", *Internet 9.0 Copenhagen*

M.P. Evans, S.A. Williams, P. Parslow, R.P. Ashton, K.O. Lundqvist and E. Porter-Daniels (2008), "The MeAggregator: Personal Aggregation and the Life-Long Tail of the Web", *International Symposiom on Human Aspects of Information Security & Assurance*

Patrick Parslow, Karsten Øster Lundqvist, Edwin Porter-Daniels, Shirley Williams (2008), "Folksonomological Reification", *Book published www.lulu.com*

Karsten O. Lundqvist, Shirley Williams, Patrick Parslow, Michael Evans, Keith Baker, Robert Ashton (2007), "MeAggregator™", *Proceedings of ePortfolio* 2007

Michael L. Brown, Karsten O. Lundqvist (2007), "ontoReadiness: una metaontología para la certificación en preparación. y portabilidad de la carrera profesional", *CEPIS UPGRADE* Novática: revista de la Asociación de Técnicos de Informática pp 40-44

L. Brown, Karsten O. Lundqvist, Shirley Williams, and Keith Baker (2007), "ontoReadiness: A meta-Ontology for Readiness Certification and Career Portability", *CEPIS UPGRADE* the European Journal for the Informatics Professional pp56-61

Karsten O. Lundqvist, Keith Baker, and Shirley Williams (2007), "Competency Relationships and Implications for Transparency between Competency Frameworks", *Presentation at Key Competency Conference*

Contributer to (2007): "HR-Semantics Roadmap - The Semantic challenges and opportunities in the Human Resources domain", *The Ontology Outreach Advisory publication*

Shirley Williams, Karsten Lundqvist and Keith Baker (2007), "Building Communities via a Learning Landscape", *Proceedings of iLearn* 2007

Karsten Lundqvist, Keith Baker and Shirley Williams (2007), "An Ontological Approach to Competency Management", *Proceedings of iLearn* 2007

Karsten Lundqvist, Shirley Williams and Keith Baker (2006), "Evaluation of Quality of e-Learning Environments", *the 5th European Conference on e-Learning*, Winchester, edited by Dan Remenyi, ISBN 978-1-905305-30-8, pp216-223

Xin He, Xiaochun Cheng, Karsten Lundqvist, Kecheng Liu, Yan Zheng, and Xianchuan Yu (2006), "A Competency Based Ontology-Driven Framework Using Agent-Oriented Semantic Web Approach", *Proceedings of the 12th Chinese Automation & Computing Society Conference in the UK*

D Rodríguez, Karsten Oster Lundqvist, S Williams, K Baker (2006), "Personal metadata for cataloguing microcontent – linking to large ontologies?", *Micromedia & e-Learning 2.0: Gaining the Big Picture Proceedings of the Microlearning Conference 2006*, ISBN 3-901249-99-0, pp288-294

Other literature

A.L.I.C.E. AI Foundation, (2007). 'Alicebot', *A.L.I.C.E AI Foundation*, USA.

Adida, B., & Birbeck, M. (2008). 'RDFa Primer Bridging the Human and Data Webs'. *W3C Working Group*, USA

Aduna. (2002), 'User Guide for Sesame', *Aduna*, Netherlands

Allen, J. (2004), 'Natural Language Understanding (2nd Indian Reprint ed.)', *Pearson Education*, India.

Aristotle. (350 B.C.), 'Metaphysics. In Book IV', *Aristotle*, Greece.

Bechhofer, S., Moller, R., & Crowther, P. (2003) 'The DIG Description Logic Interface', *International Workshop on Description Logics*, Italy

Berners-Lee, T. (2005), 'Primer: Getting into RDF & Semantic Web using N3', *W3C*, USA

Berners-Lee, T. (2006), 'Rules and Formulae', *W3C*, USA

Berners-Lee, T. (1998), 'Semantic Web Road map', *W3C*, USA

Berners-Lee, T. (1998), 'What the Semantic Web can represent', *W3C*, USA

Boon, J., & van der Klink, M. R. (2003), 'Competencies: the triumph of a fuzzy concept', *International Journal of Human Resources Development and Management* Switzerland

CEN. (2006), 'European ICT Skills Meta-Framework - State-of-the-Art review,clarification of the realities, and recommendations for next steps', *Brussels: CEN*, Belgium

Clark & Parsia LLC. (2006), 'Pellet: The Open Source OWL-DL Reasoner', *Pellet*, USA

Commission of the European Communities. (2005), 'on key competences for lifelong learning.', *Commission of the European Communities*, Belgium

Commission of the European Communities. (2006), ‘on the establishment of the European Qualifications Framework for lifelong learning.’, *Commission of the European Communities*, Belgium

Doctorow, C. (2001), ‘Metacrap: Putting the torch to seven straw-men of the meta-utopia’, *Doctorow*, Canada

Dodds, L. (2005). ‘Introducing SPARQL: Querying the Semantic Web’, *XML.COM*, USA

Genesereth, M. R., & Nils, N. J. (1986). ‘Logical Foundations of Artificial Intellingence’, *Morgan Kaufmann Publishers Inc*, USA

Gruber, T. R. (1993), ‘A Translation Approach to Portable Ontology Specifications’, *Knowledge Acquisition*, UK

Gruber, T. R. (1992), ‘Ontolingua: A Mechanisme to Support Portable Ontologies’, *Stanford University*, USA

Gruber, T. R. (1995), ‘Toward Principles for the Design of Ontologies Used for Knowledge Sharing’, *International Journal of Human-Computer Studies*, USA

Guarino, N. (1998), ‘Formal Ontology and Information Systems’ in N. Guarino (Ed.), *Formal Ontology in Information Systems*, IOS Press, Netherlands

Guarino, N., & Giaretta, P. (1995), ‘Ontology and Knowledgebases - Towards a Terminological Clarification’, in N. Mars (Ed.*) Towards Very Large Knowledge Bases - Knowledge Building and Knowledge Sharing*, IOS Press, Netherlands

Haykin, S. (2004), ‘Neural Networks. A Comprehensive Foundation’, *Prentice-Hall of India*, India

Hewlett-Packard Development Company. (2003), ‘Joseki - A SPARQL Server for Jena’, *Joseki - A SPARQL Server for Jena*, USA

Horrocks, I., Patel-Schneider, P. F., & Harmelen, F. v. (2003), ‘From SHIQ and RDF to OWL: The Making of a Web Ontology Language’, Journal of Web Semantics, UK

IEEE. (2007). Learning Technology Data Model for Reusable Competency Definitions. New York: IEEE.

IMS Global Learning Consortium, Inc. (2005), ‘IMS ePortfolio Best Practice and Implementation Guide’, *IMS Global Learning Consortium*, USA

IMS Global Learning Consortium, Inc. (2002), ‘IMS Reusable Definition of Competency or Educational Objective Specification’, *IMS Global Learning Consortium*, USA

Jena. (2007), ‘Jena Semantic Web Framework’, *Jena Foundation*, USA

Jensen, F. V. (1996), ‘Introduction to Bayesian Networks’, *Springer-Verlag*, USA

Loeb, L. (2001), ‘XML signatures: Behind the curtain’, *IBM*, USA

O*NET. (2007), ‘O*NET Online’, *O*NET*, USA.

O'Reilly, T. (2005), ‘What Is Web 2.0’, *O'Reilly Net*, USA

Ostyn, C. (2006), ‘Proposed Draft Standard for Learning Technology - Simple Reusable Competency Map’, *ostyn.com*, USA.

Smith, B. (2004), ‘Beyond Concepts: Ontology as Reality Representation. Formal Ontology and Information Systems’, *IOS Press*, Netherlands

Smith, G. (2008), ‘Tagging People-Powered Metadata for the Social Web’, *New Riders*, USA

Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J., & Horrocks, I. (2006), ‘Fuzzy OWL: Uncertainty and the Semantic Web’, *W3C*, USA

Surowiecki, J. (2004). ‘The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations’, *Little Brown*, UK

Vander Wal, T. (2007), ‘Folksonomy’, *vanderwal.net*, USA

W3C. (2004), ‘OWL Web Ontology Language – Guide’, M. K. Smith, C. Welty & D. L. McGuinness (Ed.), *W3C*, USA

W3C. (2004, February 10), ‘OWL Web Ontology Language – Overview’, D. L. McGuinness & F. v. Harmelen (Ed.), *W3C*, W3C. (2007), 'SPARQL Query Language for RDF', E. Prud'hommeaux and A. Seaborne (Ed), *W3C*, USA

W3C. (2008), 'XML Signature Syntax and Processing', *W3C Recommandation*, USA

Wallace, R. (2006), 'AIML Overview', *A.L.I.C.E. AI Foundation*, USA

Winterton, J., Delamare - Le Deist, F., & Stringfellow, E. (2005), 'Typology of knowledge, skills and competences: clarification of the concept and prototype', *Cedefop*, Greece

WordNet. (2007), 'WordNet, a lexical database for the English language', *WordNet – Princeton University*, USA

Yen, J., & Langari, R. (2005), 'Fuzzy Logic. Intelligence, Control, and Information (Fourth edition)', *Pearson Education*, Singapore

Appendixes

Can be found on the accompanying CD or at http://www.oster-lundqvist.com/karsten. They are:

- Appendix A: Chatbot Ontologies
- Appendix B: OwlLang JavaCC code
- Appendix C: E*NET ontologies
- Appendix D: Java code