Harvesting Information from the Internet to Construct Ontologies

Thomas W. Jackson[*] and Steve Smith

Department of Information Science, Loughborough University, Loughborough,

Leicestershire LE11 3TU, UK. [*]t.w.jackson@lboro.ac.uk**,** (+44) 1509 635666 ([*]corresponding author)

Harvesting Information from the Internet to Construct Ontologies

The paper evaluates the effectiveness of harvesting information from the internet to aid in the low-cost construction of an ontology. The paper describes how a proof-of-concept called OntoRanch was built, to harvest information and its relationships to construct an ontology. A systems development methodology was adopted which recognises three main stages: concept development, system building, and system evaluation. The evaluation took an interpretive hybrid approach of using both a focus group and a questionnaire to evaluate the proof-of-concept OntoRanch. The findings show that the approach of reusing information by harvesting it from the internet can provide an effective self-sustaining process that enables ontologies to be constructed in a reduced amount of time and cost.

Keywords: Ontology Development; Information Retrieval; Information Harvesting; Information Reuse; Structured Knowledge Creation

1.0 Introduction

Ontologies are often used on the World Wide Web to help in the retrieval of information. Ontologies can range from simple taxonomies used for categorisation to more complex ontologies containing numerous types of relationships and links. Ontologies offer a wide range of possibilities and offer a machine readable format for storing information. They can also be used in conjunction with traditional searching in order to aid the search process (Fensel, D., Hendler, J., Lieberman, H., Wahlster, W., 2005). The use of ontologies to overcome the limitations of keyword-based search has been put forward as one of the motivations of the Semantic Web since its emergence in the late 1990s (Vallet, Castells, Fernandez, Mylonas, & Avrithis, 2007).

The Semantic Web introduces the next generation of the Web by providing a layer of machine-understandable data e.g. for software agents for intelligent information systems, sophisticated search engines and web services. Ontologies play an important role for these knowledge intensive applications as a source of formally defined terms for communication (Sure et al., 2002). The aim is to capture domain knowledge in a generic way and provide a commonly agreed understanding of a domain, which may be reused, shared, and utilised across applications and groups. However, due to the size of ontologies, their complexity, their formal underpinnings and the necessity to come towards a shared understanding within a group of people, ontologies are still far from being a commodity (Sure et al., 2002). Ontologies have existed for a number of years and in the early 1990s ontologies moved beyond the academic domain into mainstream business (M. Hepp, 2008). Ontologies range from large taxonomies categorising Web sites, such as Yahoo, to categorisations of products for sale and their relationships as used by Amazon. They are usually created for an environment in which a common understanding of the structure of information amongst users or software agents is required (Ding, Kolari, Ding, & Avancha, 2007). A growing number of companies now offer support for ontologies and triple stores such as Oracle within in their Oracle 11g database system, incorporating ontologies into their commercial offerings. This offering has made ontology development more accessible, but there are still barriers that have to be overcome. Common understanding of information is often referred to as one of the major drivers for ontology development (Fensel, D., Hendler, J., Lieberman, H., Wahlster, W., 2005). Achieving a common understanding is quite time consuming and is seen as a barrier for many organisations back in the late nineties (Farquhar, Fikes, & Rice, 1997). However, over a decade later this is still one of the barriers to creating ontologies. In addition to achieving a common understanding there are a large number of other factors that must be considered when starting to create an ontology. Some authors even state that due to the vast amount of information that is necessary, ontology development has proven extremely expensive and possibly impracticably expensive (Farquhar et al., 1997), 7]. The

cost of ontology development and maintenance is often quoted as a key concern of the semantic web (Shadbolt, Hall, & Berners-Lee, 2006). Although the algorithms and technology associated with ontologies continue to improve, the human side of ontology development is actually the expensive part.

Ontology creation is usually a manual task that is quite time consuming. With such a complex development process and so much potential for error, ontology development comes with great risk and great expense, even when executed correctly. An example is the Gene Ontology, a well known ontology in the biomedical field, is known to have cost at least an estimated $16 million at the end of 2006 (Good et al., 2006). However, researchers such as Shadbolt, Berners-Lee & Hall have shown that in some cases, the costs are recoverable from the overall benefits of developing the ontology (Shadbolt et al., 2006). In recent years there has been some research undertaken into semi and fully automating the process (M. Hepp, 2008). This research has been focused on harvesting information from the Internet to create the ontology. The research concept is to utilise existing content and structures used on the Internet to provide some or all of an ontology. The majority of the research to date, however, has only focused on automatically harvesting concepts from the Internet to create an ontology (M. Hepp, 2008). This approach only provides the concepts and in most cases, does not involve complex relationships between the concepts.

In 2006, Hepp, Bachlechner and Siorpaes researched and discussed the possibilities of taking concepts from Wikipedia and using them as ontological structure (M. Hepp, Bachlechner, & Siorpaes, 2006; Ponzetto & Strube, 2007). Given that Wikipedia is a consensus driven system and has a great human involvement, the research questioned the validity of Wikipedia forming the basis of an ontology. The research gave some positive results, with the English version of Wikipedia containing over 850,000 entries. The study showed that not only could wikis form the basis of ontological creation but also that the URLs of Wikipedia itself could form an ontology.
A paper called “From Wikipedia to Semantic Relationships” (M. Hepp et al., 2006) showed that following a significant amount of work or training, it is possible to take relationships that have been gained from a corpus of information and apply these rules to the information stored within Wikipedia. This allows the extraction of relationships from Wikipedia. Following the training, the system could then be asked to complete a task, for example, to find all people that were born in 1900. Although there were a significant number of anomalies, relationships were discovered and it was possible to extract relationships from Wikipedia to derive useful information (Ruiz-Casado, Alfonseca, & Castells, 2006).

Research by Ponzetto and Strube looked at deriving large scale taxonomies from Wikipedia (Ruiz-Casado et al., 2006). The study took the categories of Wikipedia and then attempted to use methods to search for “is a” and “not is a” relationships within the text. Given these relationships, the system then attempted to create a taxonomy of concepts with “is a” relationships. The research provided the first logical step in automatically creating an ontology from Wikipedia and was arguably competitive to ResearchCyc, an established ontology.

The results of these research studies have shown the potential that exists for extracting information from the Internet and in particular from Wikipedia, one of the largest knowledge bases on the Internet. The key issue lies within the quality of the results extracted. Only seemingly simple relationships could be extracted from Wikipedia and the fully automated step also did not produce ontologies of a high enough quality for production use. The aim of the research presented in this paper was to develop and assess the potential of a semi-automated approach to building ontologies that could discover information from sources such as Wikipedia and Google and provide relationships between the harvested terms to construct a useable ontology. This paper builds on Ponzetto and Strube’s research into deriving large scale taxonomies from Wikipedia (Ponzetto & Strube, 2007). The paper starts by looking at the research design that was used to build and evaluate the proof-of-concept OntoRanch. It then looks at the concept of harvesting information from the internet and using a visual method to report the results to the end user for selection. Developing and implementing OntoRanch form the next section, followed by the evaluation of OntoRanch by employees who work for a multinational organisation. The paper finishes with a conclusion and limitations of the research.

2.0 Research Design

The research adopted a systems development methodology which recognises three main stages: concept development, system building, and system evaluation (Ponzetto & Strube, 2007). In terms of system evaluation a system-centred approach was taken. Wang's and Forgionne's comprehensive approach was considered but rejected due to the time constraints of the research project (Burstein, 2002). Much of the time was spent building the system, but future research will look to implement the decision-theoretic approach with more focus on gauging the reaction of the users to the system.

A number of different methods were used to test the OntoRanch system at SoftwareCo, which are detailed later in this section. SoftwareCo, is one of the largest software organisations in the world and is in the top 10 of all of the major software rankings including the Forbes2000 and Research Foundation's top 100. The organisation employs over 50,000 people in over 50 countries. The company develops a range of software solutions in house and its products are used globally. The SoftwareCo department that participated in this work was one of the rapid development and value prototyping divisions. The department's employees are highly skilled within their respective field and the department has a very unique structure. The department has attempted to create an environment specifically suited to rapid application development, testing and deployment. The department aims to have only a limited hierarchical structure, with all members of the department seen as equals and interacting with each other to take advantage of their respective skills.

2.1 Concept Development

To gain a deeper understanding of ontology development a focus group was formed. A focus group was used because the focus group method enables an understanding of how all of the members regard ontology development and the requirements of a system.

Ten members were selected for this focus group comprising of employees and contractors of SoftwareCo. The members of the focus group were all members of the rapid prototyping and development department of SoftwareCo. The focus group contained ten members who all had an understanding and work-related interest in semantic technology. The focus group participants had a mixed understanding of ontologies although all were familiar with them and all had knowledge of OWL. Although some of the members of the focus group were contractors rather than employees they shall be regarded as employees for the purposes of this work. The focus group discussed issues surrounding ontology development and their benefits and drawbacks. They were also asked to complete a questionnaire regarding their knowledge of ontologies and the issues they have with their construction. The questionnaire used a four point Likert scale of definitely, maybe, rarely and never. The results of the requirements analysis are provided in section three. 'Concept Development Part 2: Requirements Engineering'.

2.2 Evaluation of OntoRanch

Ten participants for the evaluation study were selected by one of the champions within the organisation, who selected on the criteria of someone who would give valuable input, and their knowledge of tagging (experienced and inexperienced). The ten participants were also chosen to include the most varied views including those who would be expected to be advocates of such a system and those expected to be against it.

Two evaluations were performed to evaluate the OntoRanch system. The evaluations involved a focus group, which was held at SoftwareCo and the second was a telephone interview with one of the ontology masters at SoftwareCo. The telephone interview was conducted because the employee was unable to be present during the focus group, but had used the system for a number of months to create an ontology. The first evaluation, shown by Figure 1, took a hybrid approach of using a focus group and a questionnaire. The approach taken was chosen because of the difficulties and tradeoffs that had to be made between:

- the number of champions and people available to take part in the research;
- the time that they could allow;
- and because transcription or recording of the participants was not permitted by SoftwareCo.

The participants were allowed to use the questionnaire to record their thoughts anonymously whilst the focus groups allowed elaboration and discussion to help understand the true feelings and thoughts of the group collaboratively. Although transcription or recording was not allowed by the organisation a number of notes and 'sound bites' were taken that were approved by the participants.

- **Step One** – A demonstration of the OntoRanch system was given to the employees at SoftwareCo within the focus group environment.

- **Step Two** – The focus group consisted of 10 members from the SoftwareCo. A focus group was chosen to enable an open discussion to enable participants to cover a large range of aspects that are relevant to them as a group. All participants had some form of experience with ontology development tools and were familiar with OWL. Before the focus group participants were asked how they rated their experience with ontologies. Four of the employees stated that they only had a limited experience with ontologies. The four participants with limited experience with ontologies were also given a prior introduction to some of the alternative tools available, such as protégé and TopBraid composer. The participants that had limited experience were given time to create a number of ontologies and experience those tools before the focus group to enable fair comparisons to be made.

- **Step Three** – The questionnaire contained 29 questions and questions comprised of a number of multiple-choice questions and some free text questions. The multiple-choice questions gave participants five options on an ordinal scale and included a neutral answer. The questions focused on five main themes, Discovering Information, Duplication, Usability, Restrictions and Collaboration, and Overall Evaluation.

Figure 1 – Evaluation of OntoRanch System

In the second evaluation, a telephone interview was conducted with an employee who had used the OntoRanch system extensively since its deployment within the organisation. The participant was unable to attend the focus group. The points raised by the participant are included at the end of the focus group results in the overall evaluation section.

3.0 Concept Development Part 2: Requirements Engineering

Both the focus group and the questionnaire provided useful feedback that was used in the concept development phase. The results showed that some participants rated their knowledge of ontologies highly, having worked with them quite intensively. Other members had a very limited awareness of using and developing ontologies. Three users (30%) stated they had a strong awareness of ontologies and ontology development, three (30%) had a fair awareness and four (40%) had a low awareness. None of the users stated that they didn't have an awareness of ontologies. When asked if they felt ontologies could help to retrieve information more effectively, 70% said definitely and 30% stated maybe. There was clear feeling that ontologies could be of use within the organisation, especially within the field of information retrieval. The main barriers identified by the employees to creating ontologies centred on a number of factors. The first and foremost was a lack of time or money, and in many cases these were treated as the same thing. The next key factor was a lack of knowledge or expertise, and finally a lack of tools was mentioned. Another extremely interesting observation was ontologies are "often purpose specific, but that purpose can change". This comment highlights that ontologies need to be adaptable and easily maintained.

Of the ten users, eight (80%) felt that having one user to create and maintain an ontology was a problem and that a system to allow different users to concurrently add to an ontology was more desirable. Amongst other reasons the problem of one person maintaining the ontology was related to the time that it takes for a user to create an ontology and that more than one area of the organisation may need to update the ontology. It was inferred from these discussions that collaboration would be necessary. Only one user felt that having one user create or maintain the ontology was rarely a problem and one user said it was never a problem.

There was a general feeling (80%) that having a lack of time to create an ontology was also a problem, but it was worth the effort. The time that it takes to discover the concepts that should be added and modelled in the ontology was also seen as a problem with 66% of the users agreeing with this statement. There was also a feeling that users needed training before they could create an ontology. The participants were then asked to explain any issues they may have with ontology development. Along with suggestions that time and money cause problems, there is also a lack of adequate tools to create an ontology. One participant raised the issue that “Ontologies impose a view with little flexibility” and that view often changes. It is likely that employees will view the ontology differently and coming up with one definitive view is difficult. Other users noted that it is extremely difficult to make ontological commitments from the beginning of the development process, especially as the structure will inevitably change.

Given these remarks a number of conclusions can be made. Firstly, employees feel that ontologies can be of use to the organisation. Secondly, the lack of time or money devoted to ontology development causes a problem and thus any tool developed should aim to save time and make ontology development easier for employees. The tool should also help users without them having to have a great understanding of a specific tool. The ontology development tool should also allow an ontology to adapt and change as required. One of the key findings was that the tool should be collaborative and allow more than one user to use the tool at the same time.

Given the difficulties in ontology development SoftwareCo employed a philosopher who was charged with the creation of the ontology. It was decided by the organisation that some additional restrictions on the system would be necessary in order to abstract the difficulties that might exist when creating ontologies. A number of things were done to perform this abstraction. Firstly, there should be no scope within the system to create instances. This system should be designed in order to allow the quick creation of a framework that classes could be placed into. Classes and instances in ontologies are extremely similar to those within object-oriented programming. A class represents a type of object that can be used to describe many different actual occurrences of that class that are called instances. Since ontologies model the world around them, instances within an ontology are often representations of physical objects that the ontology is modelling. As an example, a class called ‘car’ could be created, the physical car with registration plate “ab01 cde” would be an instance of the class car. The ‘car’ class could be further refined to be a class called Aston_Martin, describing all of the cars created by the company Aston Martin.

The classes form the basis of the ontology, and other systems would be capable of both using these classes and also creating instances if necessary. However, the manual task of bootstrapping and creating the ontology would only initially involve the creation of classes and the relationships between these classes. The relationships that could be created should also be restricted. An overall ontology master could define properties but the individuals working with the ontology on a daily basis should be restricted to the properties that were already defined. There was much debate over the properties that would be included and in this instance it was decided that only two relationships would exist. The first would be an ‘is a’ relationship. This ‘is a’ relationship would be expressed using the RDFS ‘subClassOf’ attribute. The other relationship that would exist would be a ‘related to’ property that would simply allow the ontology creator to state that two things were related. There were also discussions as to whether the ‘part_of’ relationship should be included during this initial phase.

The system also had to be modular. In some cases only certain parts of the ontology would be necessary and at other times the entire ontology would be needed. Fortunately the concept of namespaces is a frequent one in ontology development with RDF and OWL both allowing

namespaces to be incorporated and different files to be imported by an ontology. This allows the different namespaces be added to a different file and imported as necessary. Another important observation was that a class or concept might have many different lexical representations and that each of these lexical representations could describe the same concept. In order to simplify this, it was decided that lexical representations would be handled separately by the system and added to the concept. This would allow the person creating the ontology to simply add the lexical representation and not have to be concerned with the way that the concept was represented within OWL. The key requirement of the system was that the system should help the user to create an ontology easier than before. Namespaces are another feature of OWL. Namespaces simply provide an area in which concepts may be placed and allow the user to separate the ontology. When different namespaces are created they are often placed into different files although this is not technically a requirement. Namespaces can then be referenced within the OWL document to show that a concept that is being referred to actually belongs to a different namespace than the current one and that namespace can be imported if required.

3.1 Requirements

Based upon the review of previous tools (seen in section two), the literature review, and the focus group, a number of requirements were constructed. The requirements included:

- Users being able to quickly bootstrap and create an initial ontology containing a number of concepts whilst being given as much help as possible.

- The system must allow concurrent access. As there may be many occasions when different users would be adding to the ontology at the same time and would need to see the updates that the other person had created. Further to this, the same user may also forget that they had already dealt with a concept.

- The system should aid users in determining if a concept already existed in the ontology and thus help reduce the number of duplicated concepts entered into the system.

- The system should be designed to minimise the complexities of creating an ontology wherever possible. In many cases, it is desirable to ensure that an employee that understands ontologies in great detail creates the ontology. In this scenario SoftwareCo determined that the ontology master did not need an in depth knowledge of ontologies and the technologies surrounding them. Rather the ontology master should be of the mindset to create the ontology from a philosophical point of view.

- The system should be able to create something that enables a very fast creation of a basic ontology. Detail could be added later but the organisation needed a starting point. Many pieces of literature detail how ontologies can take a significant amount of time and investment before they can even be used (e.g. (Wang & Forgionne, 2008)). The aim with this tool is to start simple and then add greater detail later.

- The system should allow different lexical representations of a concept to be entered into the system. This would allow the same concept to be reached from different synonyms within text.

- Restrictions should be allowed to be imposed on what may be entered into the system:
    - Control over properties available to those who work with the ontology should be added so that users are not free to enter any properties they choose and are confined to those already entered into the system.
    - Instances should not be allowed from the tool

- Modularity should feature in the system by allowing users to create a number of namespaces and place concepts into those namespaces.

- The system should allow users to search for a concept rather than have to find the concept in the existing hierarchy. This again could help to reduce the likelihood of duplications.

4.0 Concept Development Part 2: Harvesting Information from the Internet

Research into harvesting information was undertaken to determine if information could be extracted from sites such as Google and Wikipedia to help users develop an ontology. The following section outlines the research behind the harvesting process and how it was developed.

The Concept Cloud method detailed in Anon's research integrates into existing search result systems, by presenting a small visualisation to the user along with the existing results, as shown by Figure 2 (Farquhar et al., 1997; Good et al., 2006; Shadbolt et al., 2006). This provides a short visual summary, enabling users to gather an overview of the content within the search result, but it does not detract from the existing method of presenting results. This is important because it does not present something entirely different to the user but supplements the results that already exist. The research question that was posed during the development of the OntoRanch system was 'would the Concept Cloud system be able to show the concepts surrounding a known page to give information on a certain subject?'

In order to explore the research question a prototype experiment was conducted with a number of concepts searched for and a Concept Cloud created for the results. The Concept Cloud created and the concepts that surround the given subject of the page were examined in order to determine whether Concept Clouds help suggest other topics that frequently relate to a given subject. The first approach was to search Google for a specific term that was known to the authors and see how the Concept Cloud related to that term. As Microsoft .Net Framework has a large presence on the web it was decided that one of the languages within that framework should be entered as a search term. As there could potentially be issues relating to 'C#' due to the sharp character, Visual Basic was entered as a search term. The phrase knowledge management was also used, as shown by Figure 3.

Figure 2 - Google search for 'Visual Basic'

Figure 3 - Google search for 'Knowledge Management'

Figures 2 and 3 show a Concept Cloud of Google searches for visual basic and knowledge management respectively. Interestingly along with the expected search terms, their abbreviations VB and KM both appear. A number of other terms related such as Microsoft Net MSDN appear. This highlights that it is possible to see a number of related terms. Previous literature has included systems that can automatically detect relationships from content on the Internet (e.g. [10]). In order

to see whether relationships might be extracted by simply entering the search terms into the Google search engine, a search term followed by ‘is a’ was entered into Google.

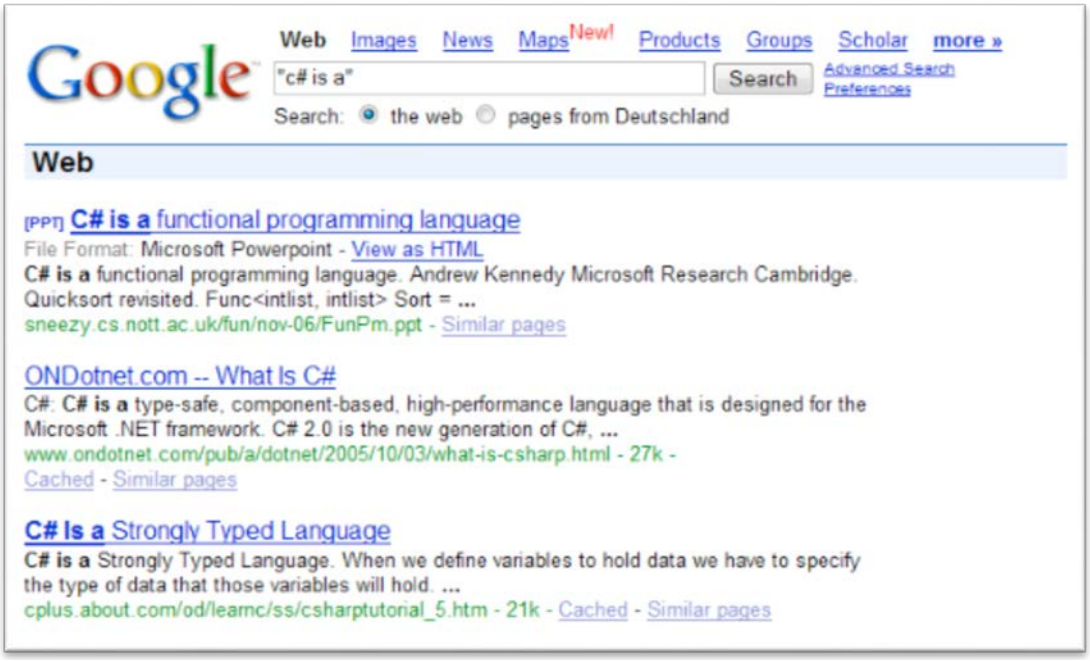

Figure 4 – A screen shot of the Google search results for “C# is a”

Figure 5 – A Concept Cloud showing the Google search for "VB is a"

Figure 4 shows the results of a Google search that searches specifically for the string ‘C# is a’ and then lists the results. The aim of this search was to use the search relevant query text that is returned with the links in order to find the answer to that specific question. Figure 4 shows that C sharp is a functional, type-safe, component-based, high-performance and strongly typed language. Most importantly it can be inferred that C# is a language. With a good enough ontology as a background it is also possible to infer things like ‘a strongly typed language is a programming language’. Again this highlights the use of ontologies. Figure 5 shows the Concept Cloud for the search ‘VB is a’. In this output, language is the top related term and programming is also mentioned. This again shows further possibility of harvesting information from Google. The use of the “is_a” term shows potential for harvesting a hierarchical relationship from Google.

Using Google as a resource provides a useful option for extracting information and concepts related to a certain term. However, if a Wikipedia article already exists for the given term then it would also be possible to extract concepts from the Wikipedia page. Wikipedia provides a vast corpus of knowledge that is always updated and maintained by a community. The relationships between concepts entered into Wikipedia are created by users and show links between two concepts that an author has deemed important enough to include. Although the credibility of Wikipedia is sometimes debated (e.g.(Anon,); (Denning, Horning, Parnas, & Weinstein, 2005)) it can help in suggesting relationships that might exist to the ontology developer. Figure 7 shows the Concept Cloud generated from the Wikipedia entry for visual basic.

Figure 7 – A Concept Cloud generate from the Wikipedia "Visual Basic" entry

Whilst generating a Concept Cloud from the entire Wikipedia article content provides a useful overview of its content, Wikipedia already contains a method for linking concepts that already exist in its system. When someone edits a Wikipedia page they have the ability to create hyperlinks to other concepts that already exist as a page in Wikipedia. Figure 8 shows a Concept Cloud generated from the links within the visual basic entry of Wikipedia. Figure 9 shows a similar cloud for the programming language, Ruby, that highlights the differences found even when searching for quite similar or related terms.

Figure 8 – Wikipedia Page Concept Cloud from the OntoRanch System for "Visual Basic"

Figure 9 - Wikipedia Page Concept Cloud from the OntoRanch System for "Ruby" highlighting the differences discovered between similar terms

The Concept Clouds generated from the Wikipedia links show a very high relation to the original concepts and highlight a large number of related concepts and technologies. The key issue with using Wikipedia is it may not contain the concepts that are necessary when building the ontology, although it can be used alongside Google to aid wherever possible. The prototype experiment has been able to answer the research question and show that it is possible to use Concept Clouds to show the concepts surrounding a certain topic. The next section describes how the ontology harvesting and creation tool was constructed to form the proof-of-concept, OntoRanch.

5.0 System Building: Developing and Implementing OntoRanch

Given that the proposed tool will harvest information and created ontologies it was decided that the system would be called OntoRanch (Ontology Ranch). There were a number of approaches that could have been taken to developing OntoRanch, but it was decided that it would be developed as a web based tool. One reason was it allowed multiple users to work on the ontology at the same time, preventing difficult synchronisation issues that might otherwise occur. Using a web-based system also makes it simple to embed another web page into the tool so that the user may make use of that page whilst creating the ontology.

OntoRanch was developed using Ruby on Rails. The choice of Ruby on Rails was due to familiarity with the environment and that rails allows very fast creation of prototypes and web content. Due to many existing systems within SoftwareCo already running Java and because many other OWL and ontology related libraries are available in Java, it was decided that JRuby would be used rather than Ruby. JRuby is an implementation of a Ruby compiler that compiles to Java byte code rather than the standard C-based implementation called MRI or Matz's Ruby Interpreter. There are several benefits to the JRuby. JRuby runs on top of the java virtual machine providing numerous benefits such as efficient garbage collection. The key advantage, however, is the full integration and interoperability between Java and Ruby when using JRuby. It is possible to call any Java library from Ruby and Ruby from Java. Another factor in the choice of programming language was the integration of AJAX. AJAX allows updates and calls to the server from a web page without the need to refresh the entire page. This would prove extremely useful in the interface of the system. Rails makes use of a model, view, controller based architecture and thus the system was designed with this in mind.

There were two key components to the OntoRanch system and some other supplementary pages. The first part of the tool is the search view. In order to try and reduce the opportunity for duplications or entries into the system that might already exist in some form or cause confusion, all entries into the system begin with the search view. The search view allows a user to search for a concept before adding a new one into the system. As the user types into the search box, the list of concepts, shown in Figure 10, is refined to show any matching concepts.

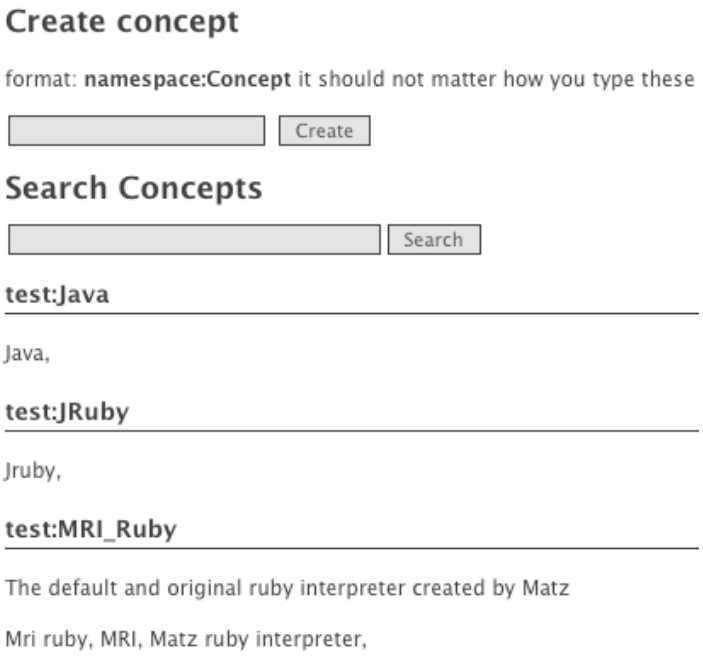

Figure 10 - OntoRanch search view

A search can be performed with or without a namespace but can also be restricted. For example to force searching within the test namespace for a concept named ruby the user may type “test:ruby” and the concept shall be refined. In order to search any namespace the user may simply enter “ruby”. Partial word matches shall also occur so “test:ru” would find the concept “ruby” within the test namespace. The system will also search the description of any concept in order to ensure that all related concepts are found. If no concept is found, the user may enter the namespace and concept name in order to create a new concept. Figure 11 shows the browse view searching for “ruby”.

Figure 11 - OntoRanch search for Ruby

Once a concept has been found or a new concept is created, the user is directed to the concept view. The concept view is reachable via its own unique URL. This makes linking to the concept quite simply. New concepts can also be created by simply entering the URL containing the namespace and concept name if desired. For example, to create the concept “ruby on rails” in the “test” namespace the following URL may be entered http://localhost:3000/concept/test:Ruby_On_Rails. Allowing a URL to be used is a simple method of creating concepts used by many online systems such as Wikipedia. This makes it easier for the user and increases the familiarity of the system, as it is similar to systems already used before.

Creating the search based system was an important decision, differing from many ontology development tools because it makes the user search before any action can be taken on the ontology. This search-first approach was assessed during the focus group in order to determine if users preferred this to the traditional method of browsing concepts in their hierarchies. The focus group was also used to assess whether users felt that this would help reduce the likelihood of duplication within the system.

The second key part of the system was the concept view. The concept view has four parts, which are highlighted in Figure 12 (Concept Label and Description, Properties, Lexical Representations, and Harvest View).

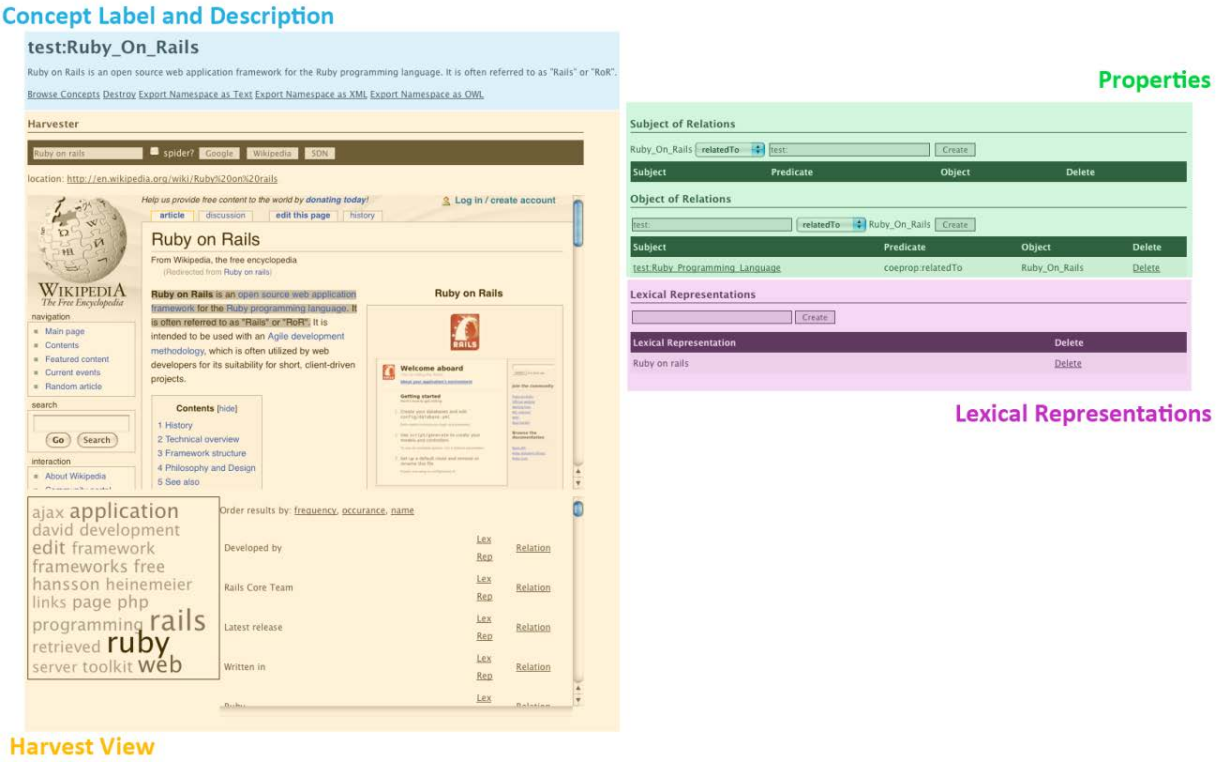

Figure 12 - The concept view

The first element of the concept view is the **concept label and description** section. This element allows the user to state the namespace and class names. The information is automatically formatted and inserts underscores and alters case according to preset rules. These rules were defined during the development of the system to ensure that all concepts entered into the system follow the same naming convention. The naming convention used by the system derives from the RDF naming convention, the only difference being that words that make up a class name are separated by underscores, whereas in RDF there is no separation of words. It was important to separate the words so that the system would be able to include spaces in any harvesting searches and so that the boundaries of different words could be interpreted by the system. During the export process the underscores are removed.

The formatting is done by an addition to the string class in ruby so that “string.conceptify” may be called at any point in time. A description may also be added, although this description is not necessarily exported it allows users to see the intended usage of this class name. If for example two similar classes exist such as the “Oracle_DBMS” or the “Oracle_Corp”, which symbolise the Oracle database management system or the Oracle Corporation exist, it prevents any misunderstanding and aids users when working collaboratively. Although it may be bad practice, both of these examples may be entered into the system in different namespaces simply as “Oracle” and the description could be used to differentiate between the two.

The next element of the system is the **properties** section or relations as they are termed within the system. Relations show all of the properties that the concept is either the ‘subject of’ or the ‘object of’ and show the entire triple. The predicate can be chosen from a list of pre-defined predicates created by the key ontology master. In order to aid the ontology creator, as the user begins to type the name of a concept into the subject or object box, all existing concepts are suggested along with their descriptions. This allows the user to insert any existing concepts. If the user wishes to insert a new concept, they simply enter a concept that has never been entered previously and it is added to the system. Suggestions are filtered based upon both the namespace and concept parts, so having the namespace present means that the search engine will search within that namespace. Partial namespace titles and concept titles are also supported. Figure 13 shows an example of the auto-complete search for concepts within the OntoRanch system.

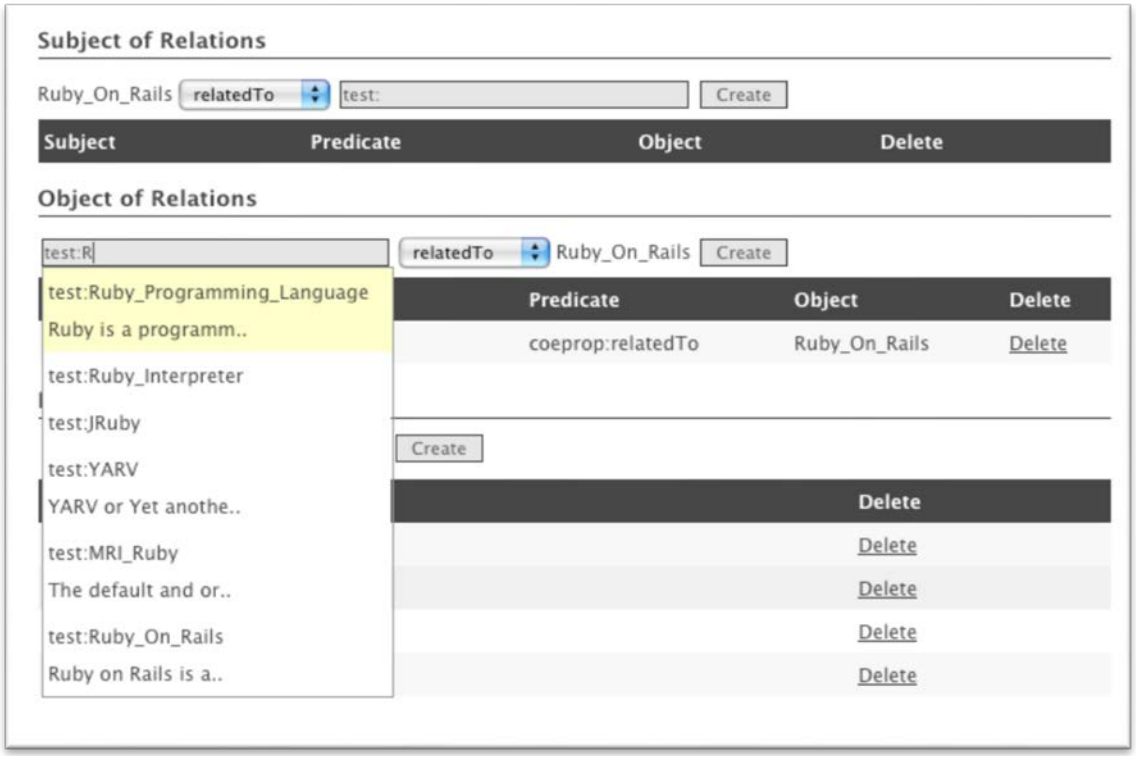

Figure 13 – Auto-complete for subjects or objects

The third section of the OntoRanch system is the **lexical representation** entry system. Lexical representations are automatically entered based upon a concept title and whenever the concept title is changed. Figure 14 shows the lexical representations entered for the example ruby on rails class.

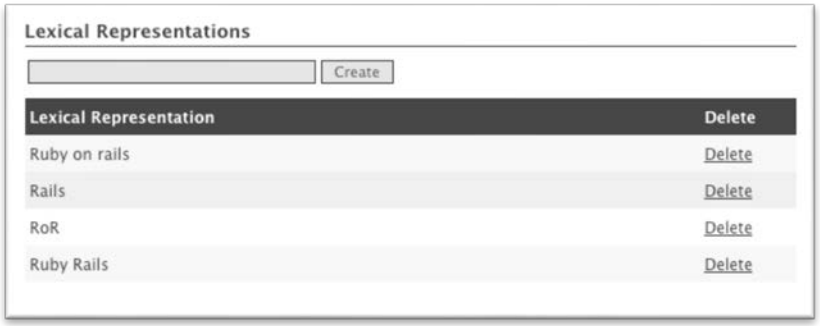

Figure 14 - Lexical representations of Rails

The most challenging part of the system to create was the **harvesting view** section. The harvest system presents a web page within the OntoRanch system. This web page is then processed and a Concept Cloud is displayed for the web page along with a list of concepts extracted from that page. The concepts extracted are either the links that exist within a Wikipedia page or terms that most frequently occur on that page for Google and other sites. The harvest system starts with a search bar allowing the user to enter any search term. The default search term is the name of the concept with spaces rather than underscores. The user may then press one of the search buttons in order to search that site and harvest the resultant page. Figure 15 shows the harvester on the Wikipedia entry for ruby on rails.

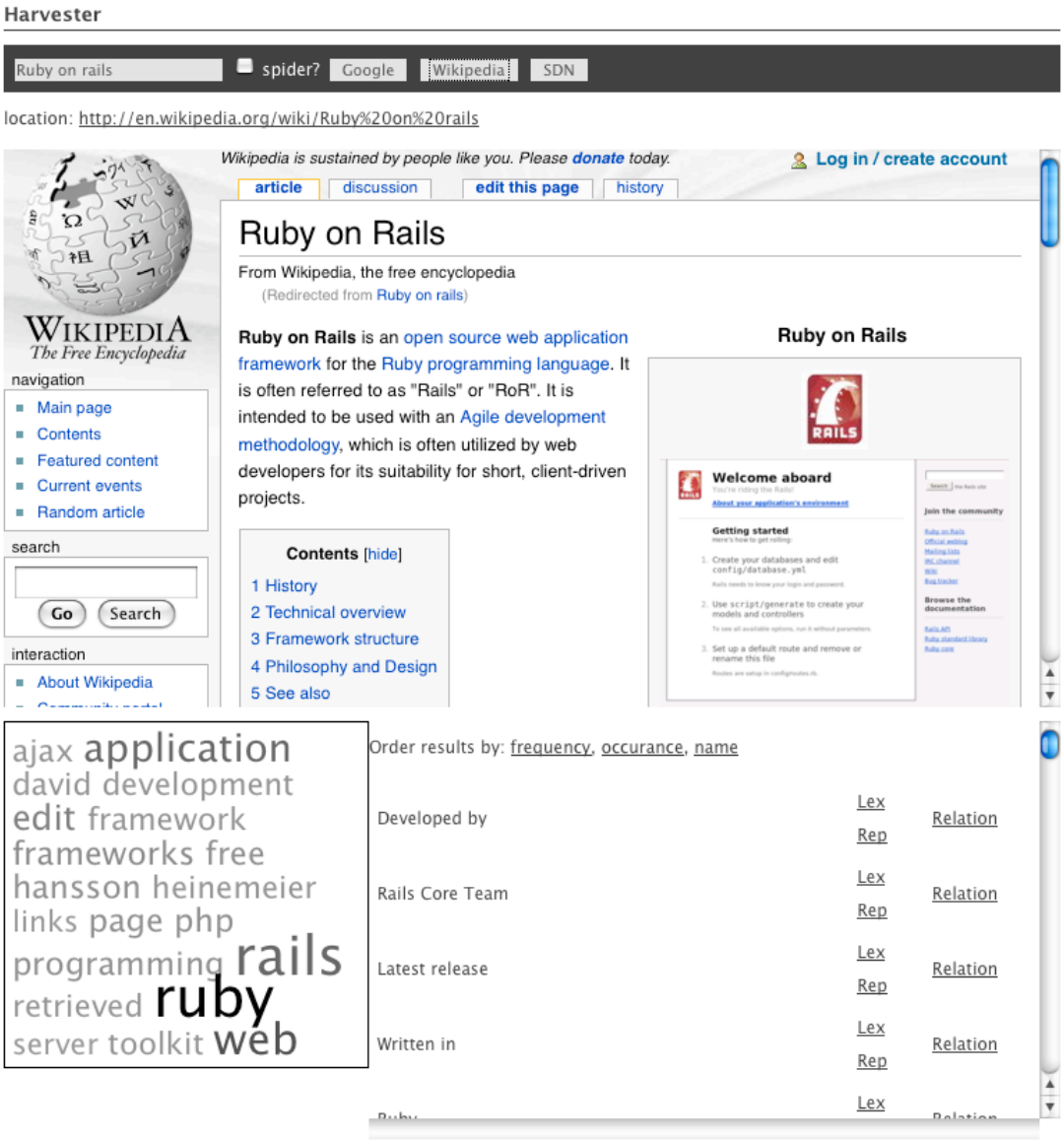

Figure 15 - The harvester searching Wikipedia

The harvester also allows the user to follow all of the links on the resulting page in order to collect the terms and create a Concept Cloud from all of the content found. This can be activated using the spider feature although it is extremely resource intensive, especially on secure sites. One of the key features of the embedded browser is that it allows the user to navigate to any page and a Concept Cloud and list of terms are extracted for that page.

The cross-domain JavaScript restriction enforced by most modern browsers prevents a site from calling JavaScript on a page or IFrame, such as the one used to create the embedded browser, from a different domain name. This created a significant challenge as in order to determine the page that the user had navigated to in the browser, a JavaScript call to that Iframe would be required. In order to solve this issue a proxy server was created and implemented using ruby on rails and the hpricot html parser for ruby. The proxy navigates to and downloads the html for a page the hpricot parser. The hpricot parser then parses the page and alters all the links so that they referred to the same location, but navigate through the proxy. CSS or cascading style sheet files were also modified so that any images or imported styles would be available. Any place that a link could exist is parsed by the engine and modified so that the proxy is used. This allows the system to always know the current page and update the Concept Clouds and related terms accordingly. It also allows the system to display a link to the current page so that the user may open it in a larger, separate browser window if required. The secondary benefit of this proxy server is that in future

work, it will allow injection of content alongside the existing content that a user is browsing. This may allow the system to do things such as scroll to the part of the article where a concept was found if the user clicks the concept in the Concept Cloud.

The harvest view also presents a number of links for each extracted concept which when clicked will either add the concept as a lexical representation of a word or will fill in the concept as the subject or object of a relation. It will also allow the user to choose the predicate before saving the property. The system allows namespaces to be modified and created by an overall administrator. These namespaces can either be local and have a local URL, or can be a remote URL. The remote URL feature can be used, for example, to import the RDF namespace. The namespaces can also be exported to OWL for importing into reasoning and related ontology systems.
There are a number of disadvantages of the system being created in its current format. The first, is the proxy server is quite resource intensive. Whenever a page is loaded into the harvester, it must be processed by the server. This processing can take considerable time when multiple accesses occur simultaneously. This is especially true when an SSL encrypted page is viewed. One of the contributing factors to this is that ruby on rails is currently not thread safe and although multiple instances of the server can be started and load balanced, it is still not an ideal solution. The current development version of ruby does however contain native threading rather than green threading and it is expected that rails will eventually become thread safe. Secondly, one of the disadvantages of the system comes from one of its advantages. The simplification of the system does impose a number of restrictions. One of these issues is that predicates for the relationships or properties that are created by the end user may not be modified by anyone other than an administrator. The advantages provided by these systems and methods were deemed by the authors and SoftwareCo to outweigh the disadvantages.

6.0 System Evaluation: Results and Analysis

Some questions were left blank by some participants, and in these cases only the completed questions were used in the results. This explains the odd percentages for some of the results as the total number might not have been out of ten participants.

6.1 Discovering Information

Participants of the focus group were first shown the OntoRanch search view, which was widely accepted. Participants of the focus group felt that it would allow them to find concepts far more easily, especially if other people had added the concepts. All of the participants felt that the search system made things easier and that it was better than having to browse a hierarchical structure, and the questionnaire results confirmed this. Employees were asked if the search based approach makes it easy to discover concepts in the system to which participants unanimously agreed (100%) that it would.

After looking at the search view, participants were shown the concept page that included the harvesting system as shown in Figure 12. Participants first examined the harvesting system and its approach. During the focus group, the key area that participants appeared impressed with was the Concept Cloud view of the page. Many of the participants felt that this alone would help to prompt them when creating an ontology. Participants felt that they should already have an understanding of an area when they were creating an ontology. They felt that the ontology creation should not be left to those users that did not understand the subject area they were describing. The Concept Cloud could therefore help to remind these users of the concepts that should be added to the ontology. The focus group participants also appeared to appreciate the web browser being built into the system. However, the fact that they could open the link in a new window was of more interest. Participants stated that they preferred to see the page in its entirety instead of within the small window of the OntoRanch system. The browser window was only really of use to find the correct page for the harvesting system to harvest.

The questions relating to the harvesting system also highlighted its potential with 89% of the participants thinking that the harvesting system worked well, providing a good list of potential concepts to add to the ontology. The same participants (89%) also agreed that the harvesting view allowed them to find adequate information regarding the concept they were adding to the system. The questionnaire results also confirmed the participants' view of the Concept Cloud system with 89%

of the participants agreeing that the Concept Cloud gave a good visual overview of the page that the system was currently displaying.

Once the harvesting demonstration was completed, all of the participants' feedback suggested that the system made it easy to add concepts from the harvested list to the ontology. The questionnaire results showed that 88% of the participants stated that adding concepts to the system was quick and easy (remaining 11% recorded neutral). Feedback from the focus group was very positive when they looked through the list of retrieved concepts surrounding a topic. Many of the focus group participants chose a subject that they were interested in and then looked for words that they would have suggested in the list of results returned. In almost all cases they were pleased to find the words that they expected to find. Those that did not find the terms they expected understood that the system would not find any results as the term did not occur commonly on the Internet but was of specialist interest to them. The questionnaire was used to verify these findings, with 89% of participants saying the system provided a good list of potential concepts.

6.2 Duplication

One of the original intentions of the system was to use Ajax in order to highlight concepts that might already exist in the system when adding new concepts to relationships. The focus group did not touch too much upon the Ajax functionality or the ability to prevent ontology masters from adding concepts that might already exist to the system. The questionnaire however asked a number of questions around this subject. A number of participants felt that the system highlighted concepts that may already exist in the ontology, 25% of the participants strongly agreed with this and 49% agreed. With one user (13%) disagreeing and one user (13%) had a neutral opinion. Two participants did not answer this question. Highlighting these concepts would enable users to see concepts that already existed in the system and allow users to link to these concepts from the one that they were creating. The main aim of this feature was to help to prevent the participants from creating concepts that already existed in the system. If the participants created duplicate concepts then the ontology would be disjointed. The reasoning performed on such an ontology may therefore be incomplete or incorrect.

Questionnaire participants were asked whether simply highlighting existing concepts would help the system to prevent participants from entering duplicate concepts. Although 38% of participants who answered agreed with this statement, 38% went for the neutral opinion and 25% actually disagreed with this statement. Two users did not answer this question. In the focus group, some of the participants stated that they felt the system only highlighted concepts that had a similar name, set of lexical representations or whose descriptions contained similar words. Participants stated that the system makes no attempt to highlight concepts that are perhaps synonyms of each other or that may describe the same concept but in a different way.

6.3 Usability, Restrictions and Collaboration

Many participants felt that the system allowed adequate division of namespaces, with 78% either agreeing or strongly agreeing. Two participants did not answer this question. The two (22%) participants that gave a neutral answer further explained that the system allowed separation of namespaces however, modularity is a design issue rather than something that the tool can provide. Participants were extremely happy with the system when it came to lexical representations of concepts. Within the focus group they stated that they felt that one of the key features of the tool was the lexical representation system and that this would prove extremely useful by itself even if the other features did not exist. Seven participants (78%) strongly agreed with the lexical representation system, two (22%) agreed and one user did not answer this question.

The questionnaire asked if it was important that the system enabled more than one person to work with the ontology at the same time. Opinions were split with the majority (67%) of participants strongly agreeing or agreeing, but there were two neutral opinions (22%) and a disagreement from one user (11%). One user did not answer this question. Again most participants felt that restricting the predicates that may be entered into the system was a good idea and made it easier to use, but one user disagreed. Most participants also agreed that the system allowed participants who were not necessarily experts when working with ontologies to enter items into the ontology. Two

participants (22% of those who answered) strongly agreed, 6 participants (67%) agreed, one user (11%) disagreed and one user did not answer this question.

The idea of adding a description to concepts was also favoured in the most part with two participants (22% of those who answered) strongly agreeing that it helps to prevent duplication of concepts and misunderstanding. Six (67%) participants agreed with this system and only one (11%) disagreed. One person did not answer this question.

Overall most participants felt that the system was easy to understand. When the questionnaire asked if the system took a long time to understand, one participant did not answer, three (33%) had a neutral opinion, four disagreed (44%) and two strongly disagreed (22%). This was also apparent within the focus group with most participants quickly understanding the system and using it without problems.

The questionnaire showed that participants felt that the OntoRanch system would take less time to create an initial ontology than traditional systems such as protégé or than simply using a text editor. All but one (86%) participant of the seven that answered this question also felt that the OntoRanch system would take less time to maintain an ontology once it had been created. The one participant (14%) felt that when maintaining an ontology, protégé would be quicker. The same user felt that protégé would provide a better-structured ontology, but did not say how.

6.4 Overall Evaluation

To conclude the focus group, employees were asked to consider the overall strengths, opportunities, weaknesses and threats of OntoRanch. In terms of **strengths** employees said "... the disambiguation helps fast ontology creation, and provides a good overview of the ontology". They said that OntoRanch also helped with thinking and was easy to use with the help of the search function, and the ability to search external sources. Employees commented on the strength of the structural side of OntoRanch, "...harvesting of lexical representation, with quick access to different lexical representations and potentially related entities". Employees highlight a number of **opportunities** and said '...a step towards a tool that can be used by end users to build business ontology" and that it is an "opportunity to work with all areas of the organisation to create a holistic view that can be used by the whole organisation, to aid with communication". Employees also raised a number of **weaknesses** with OntoRanch, in that in "can be too easy to use, careless proliferation and possible duplication". One employee said it "doesn't help to embed new entities into existing ontologies, especially if naming is different and it doesn't have a graph, able to handle of stop words, no visualisation, no help to reveal a total new ontology and potential some user interface issues'. The **threats** were thought to be around who can access it, "...you can store a lot of rubbish into these tools if a lot of people enter data not correctly, it would not help in improving work as it can be viewed by different areas of the organisation as an attempt by another area to impose their view of the world". The openness was also seen as an opportunity, but the SWOT analysis has shown that there was not an overall consensus on the value or limitations of some of the features of OntoRanch, with many conflicting views.

Following the focus group, an interview with an employee who heavily used the OntoRanch tool for ontology generation took place to gain his overall evaluation of the proof-of-concept. This employee was not present during the focus group. He stated that when entering concepts, he had already decided which concepts should be added, which made the harvesting tool less significant to him. Where the harvesting tool was of benefit was in determining the relationships that should exist to other concepts and within the lexical representation field. Once concepts are added he would search around that concept finding different lexical representations and any relationships that might exist. In this area, the auto-complete system also proved extremely useful. One slight concern from this power user was that jumping from concept to concept may lead to a slightly disjointed ontology if considerable thought was not placed into its creation. Overall he said that the system was "Very easy to use, but very powerful".

6.0 Conclusion

Discovering a method to rapidly build correctly formulated ontologies is essential if the semantic web or intelligent information systems are to be successful. The paper has described how OntoRanch has been conceived and developed to enable the reuse of existing information and highlighting identifying complex relationships to provide a method for creating an ontology, but it does rely on information that is at least semi-structured. The semi-automated method of being able to harvest and reuse existing structured and semi-structured information to create new structured information is of significant value to knowledge intensive organisations, as it provides an organisation with a quick method of structurally mapping a domain without investing too much time and money, compared to traditional methods of constructing an ontology. The evaluation of OntoRanch by ten employees has shown that the proof-of-concept can be used to create a meaningful ontology, but the openness of the system was viewed as both a benefit and a drawback.

There is likely to be an increase in systems like OntoRanch as more semi-structured information becomes available for potential re-use. Future research will look at harvesting concepts from a corporate intranet. The corporate intranet could provide an invaluable resource with information that is specifically relevant to the organisation. An organisation's intranet may contain domain specific information but also potentially more sensitive information that is not available in the public domain and that might make a welcome addition to an ontology to improve search performance.

7.0 References

Anon.

Burstein, F. (2002). System development in information systems research. *Research Methods for Students, Academics and Professionals: Information Management and Systems,* , 147-158.

Chesney, T. (2006). An empirical examination of Wikipedia's credibility. *First Monday, 11*(11)

Denning, P., Horning, J., Parnas, D., & Weinstein, L. (2005). Wikipedia risks. *Commun.ACM, 48*(12), 152-152. doi:http://doi.acm.org/10.1145/1101779.1101804

Ding, L., Kolari, P., Ding, Z., & Avancha, S. (2007). In Sharman R., Kishore R. and Ramesh R.(Eds.), *Using ontologies in the semantic web: A survey* Springer US. doi:10.1007/978-0-387-37022-4_4

Farquhar, A., Fikes, R., & Rice, J. (1997). The ontolingua server: A tool for collaborative ontology construction. *INTERNATIONAL JOURNAL OF HUMAN COMPUTER STUDIES, 6*(46), 707-728.

Fensel, D., Hendler, J., Lieberman, H., Wahlster, W. (2005). *Spinning the semantic web: Bringing the world wide web to its full potential* MIT Press, Boston.

Good, B. M., Tranfield, E. M., Tan, P. C., Shehata, M., Singhera, G. K., Gosselink, J., . . . Wilkinson, M. (2006). Fast, cheap and out of control: A zero curation model for ontology development. *Pacific Symposium on Biocomputing,* , 128-139.

Hepp, M., Bachlechner, D., & Siorpaes, K. (2006). Harvesting wiki consensus-using wikipedia entries as ontology elements. *First Workshop on Semantic Wikis,*

Hepp, M. (2008). In Hepp M., Leenheer P., Moor A. and Sure Y.(Eds.), *Ontologies: State of the art, business potential, and grand challenges* Springer US. doi:10.1007/978-0-387-69900-4_1

Ponzetto, S. P., & Strube, M. (2007). Deriving a large scale taxonomy from wikipedia. *Proceedings of the 22nd National Conference on Artificial Intelligence,* , 22–26.

Ruiz-Casado, M., Alfonseca, E., & Castells, P. (2006). From wikipedia to semantic relationships: A semi-automated annotation approach". *First Workshop on Semantic Wikis,*

Shadbolt, N., Hall, W., & Berners-Lee, T. (2006). The semantic web revisited. *Intelligent Systems, IEEE, 21*(3), 96-101.

Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., & Wenke, D. (2002). OntoEdit: Collaborative ontology development for the semantic web. In I. Horrocks, & J. Hendler (Eds.), *The semantic web — ISWC 2002* (pp. 221-235) Springer Berlin / Heidelberg. doi:10.1007/3-540-48005-6_18

Vallet, D., Castells, P., Fernandez, M., Mylonas, P., & Avrithis, Y. (2007). Personalized content retrieval in context using ontological knowledge. *Circuits and Systems for Video Technology, IEEE Transactions on, 17*(3), 336-346.

Wang, Y., & Forgionne, G. (2008). Testing a decision-theoretic approach to the evaluation of information retrieval systems. *Journal of Information Science, 34*(6), 861-876.