NeOn-project.org

NeOn: Lifecycle Support for Networked Ontologies

Integrated Project (IST-2005-027595)

Priority: IST-2004-2.4.7 – "Semantic-based knowledge and content systems"

D4.5.3 Integration of Context Sensitive Search of Ontologies with Watson

---

Deliverable Co-ordinator: Boštjan Pajntar

Deliverable Co-ordinating Institution: J. Stefan Institute (JSI)

Other Authors: Dunja Mladenić (JSI), Marko Grobelnik (JSI)

This deliverable provides an integration of Watson ontology search engine with the contextualizing technology of SearchPoint. Furthermore a single point of access has been made to use different ontology search engines and utilize different SearchPoint methodologies.

Both Watson Search engine and SearchPoint have been extended in order to make integration possible as well as to improve the human-computer interaction of the application

Finally, a NeOn Toolkit plugin was implemented to provide contextualized ontology search as the first step in the methodology of ontology reuse.

Document Identifier:	NEON/2009/D4.5.3/v1.0	Date due:	September 30, 2009
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	September 30, 2009
Project start date:	March 1, 2006	Version:	v1.0
Project duration:	4 years	State:	Final
		Distribution:	Public

NeOn Consortium

This document is a part of the NeOn research project funded by the IST Programme of the Commission of the European Communities by the grant number IST-2005-027595. The following partners are involved in the project:

Open University (OU) – Coordinator Knowledge Media Institute – KMi Berrill Building, Walton Hall Milton Keynes, MK7 6AA United Kingdom Contact person: Enrico Motta E-mail address: e.motta@open.ac.uk	**Universität Karlsruhe – TH (UKARL)** Institut für Angewandte Informatik und Formale Beschreibungsverfahren – AIFB Englerstrasse 11 D-76128 Karlsruhe, Germany Contact person: Andreas Harth E-mail address: aha@aifb.uni-karlsruhe.de
Universidad Politécnica de Madrid (UPM) Campus de Montegancedo 28660 Boadilla del Monte Spain Contact person: Asunción Gómez Pérez E-mail address: asun@fi.upm.es	**Software AG (SAG)** Uhlandstrasse 12 64297 Darmstadt Germany Contact person: Walter Waterfeld E-mail address: walter.waterfeld@softwareag.com
Intelligent Software Components S.A. (ISOCO) Calle de Pedro de Valdivia 10 28006 Madrid Spain Contact person: Jesús Contreras E-mail address: jcontreras@isoco.com	**Institut 'Jožef Stefan' (JSI)** Jamova 39 SI-1000 Ljubljana Slovenia Contact person: Marko Grobelnik E-mail address: marko.grobelnik@ijs.si
Institut National de Recherche en Informatique et en Automatique (INRIA) ZIRST – 655 avenue de l'Europe Montbonnot Saint Martin 38334 Saint-Ismier France Contact person: Jérôme Euzenat E-mail address: jerome.euzenat@inrialpes.fr	**University of Sheffield (USFD)** Dept. of Computer Science Regent Court 211 Portobello street S14DP Sheffield United Kingdom Contact person: Hamish Cunningham E-mail address: hamish@dcs.shef.ac.uk
Universität Koblenz-Landau (UKO-LD) Universitätsstrasse 1 56070 Koblenz Germany Contact person: Steffen Staab E-mail address: staab@uni-koblenz.de	**Consiglio Nazionale delle Ricerche (CNR)** Institute of cognitive sciences and technologies Via S. Martino della Battaglia, 44 - 00185 Roma-Lazio, Italy Contact person: Aldo Gangemi E-mail address: aldo.gangemi@istc.cnr.it
Ontoprise GmbH. (ONTO) Amalienbadstr. 36 (Raumfabrik 29) 76227 Karlsruhe Germany Contact person: Jürgen Angele E-mail address: angele@ontoprise.de	**Food and Agriculture Organization of the United Nations (FAO)** Viale delle Terme di Caracalla 1 00100 Rome Italy Contact person: Caterina Caracciolo E-mail address: Caterina.Caracciolo@fao.org
Atos Origin S.A. (ATOS) Calle de Albarracín, 25 28037 Madrid Spain Contact person: Tomás Pariente Lobo E-mail address: tomas.parientelobo@atosorigin.com	**Laboratorios KIN, S.A. (KIN)** C/Ciudad de Granada, 123 08018 Barcelona Spain Contact person: Antonio López E-mail address: alopez@kin.es

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed to the writing of this document or its parts:

JSI

OU

Change Log

Version	Date	Amended by	Changes
0.1	22-08-2009	Boštjan Pajntar	Overall structure of the report
0.2	23-08-2009	Boštjan Pajntar	Executive Summary, introduction
0.3	03-09-2009	Boštjan Pajntar	Chapter 2 - Architecture
0.4	18-09-2009	Dunja Mladenić	Chapter 2, Overall revision
0.5	20-09-2009	Boštjan Pajntar	Chapter 3
0.6	12-10-2009	Marko Grobelnik	Overall revision
0.7	29-10-2009	Boštjan Pajntar	Figures, Overall revision
0.8	14-11-2009	Dunja Mladenic, Bostjan Pajntar	Including reviewer's comments
1.0	16-11-2009	Chris Buttenshaw	Final QA

Executive Summary

This report is describing a software deliverable developed as an integration of a web application - SearchPoint [Pajntar and Grobelnik, 2008], and ontology search engine - Watson [d'Aquin 2008]. The main goal of SearchPoint is to enhance search engines by allowing the users to get multiple rankings of results for each query. The main goal of Watson is to provide entity and ontology search and these two complementary functionalities are joined in a single application.

In order to make integration possible Watson ontology search was upgraded with the capacity to extract key concepts from every ontology it searches. This high quality textual information about ontologies is used by SearchPoint to extract topics and help with classification of the search results into an OntoLight classified ontology using results of D3.2.1 [Grobelnik et al., 2007, Grobelnik et al., 2008].

SearchPoint was also upgraded as regards its usability. Initial implementation of SearchPoint for ontology search [Pajntar et al, 2009] was consolidated, so there is now just one point of access for all ontology search engines. The ranking space - client side GUI of SearchPoint - was also changed so more background information is now provided for each topic.

The last technical effort of this deliverable was to implement a NeOn Toolkit plugin which offers this application of a contextualized search engine over ontologies. This fulfils one of the first and most important steps in ontology reuse as defined in D5.3.1 [Suarey-Figueroa, 2007].

Table of Contents

List of Figures

1. Introduction

This deliverable presents the system for contextualized ontology search. The concept of ontology search is the first step in the process of ontology reuse (D5.3.1) which is one of the key aspects when dealing with networked ontologies. Furthermore, by contextualizing this process the intention was to increase the efficiency and quality of ontology search and ontology reuse.

The goal of this deliverable is to provide an application where searching of ontologies can be performed with the help of an arbitrary ontology search engine, as well as providing a contextualization of search results by providing classifiers of large grounded ontologies as provided by OntoLight Classifier [Grobelnik et al., 2008].

The system is using service oriented architecture in order to bind all the components. Most components were upgraded to make integration possible and also to enhance the usability of the application.

To make integration of SearchPoint with Watson possible, Watson was extended to provide key concepts for each ontology. This is necessary as SearchPoint works only with textual results. Both SearchPoint techniques - topic extraction and classification into ontologies - benefited from the carefully selected representative concepts which were used as textual representation of the result ontologies.

SearchPoint webservice was improved by making it possible to add new stopwords at runtime. Stopwords are words which are discarded in the pre-processing phase of text mining. Apart from selecting some ontology common words, query words which usually appear inside every hit, now get added to stopwords.

SearchPoint graphical user interface was extended so it can visualize several words for each topic, making it more clear for the user.

Several applications proposed in D3.2.4 [Pajntar et al., 2009] are now unified, integrated also with Watson search engine and extended both in the server and client side of SearchPoint. The web application can be found at http://searchpoint.ijs.si/ontologysearch/. The NeOn Toolkit plugin can be found at http://kameleon.ijs.si/searchpoint/. Both binary and source versions are available.

The remainder of this deliverable first describes the approach (Section 2), giving an overview of the architecture, then describes the various extensions and improvements of the modules. In Section 3 we showcase the usage of the system together with screenshots and finally we conclude with a short discussion in Section 4.

2. Approach Description

The main goal of this application is to provide a good ontology search tool that is context aware. To this end, several ontology search engines were used to provide ontology search results for a given query and SearchPoint is used to help with ranking of the ontology search results.

All of the modules are implemented with the principles of service oriented architecture which makes for an easy integration and easy deployment of the application as a NeOn toolkit plugin and web application.

2.1 System Architecture

We restate some of the points made in D3.2.4:

SearchPoint is a tool for searching. However, this is not the usual type of search we are accustomed to using for example on the web. In fact the usual search engine is merely one module in a longer chain of processes, which we will call modules.

The implementation of SearchPoint is modular which makes it very easy to change a single module, for example to adopt a new search engine, or a new classifier. Also, different distribution channels for such an application can be easily adopted. The architecture of SearchPoint (Fig 1) consists of the following modules chained into a pipeline:

1. Agent (usually user) provides a query (usually inside a GUI)
2. Selected search engine processes the query and returns a result-set of short textual documents – snippets
3. Automatic topic generation module provides a list of topics and result-topic similarity matrix (implemented as a web service)
4. Graph drawing and sub-graph extraction for visualization of the topics provides the positions for each topic (implemented as a web service)
5. Graphical User Interface (GUI) provides visualization of the topics; focus point selection and rearranging of the results dynamically.

In the current application, modules 3 and 4 are enveloped inside a single web service. This is in order to minimize the number of calls made; however, if needed they could easily be separated. Modules 1 and 5 are components of the same GUI provided in the form of a web application and a NeOn Toolkit. It seems natural to provide the user with a single place for posing queries and receiving/manipulating results.

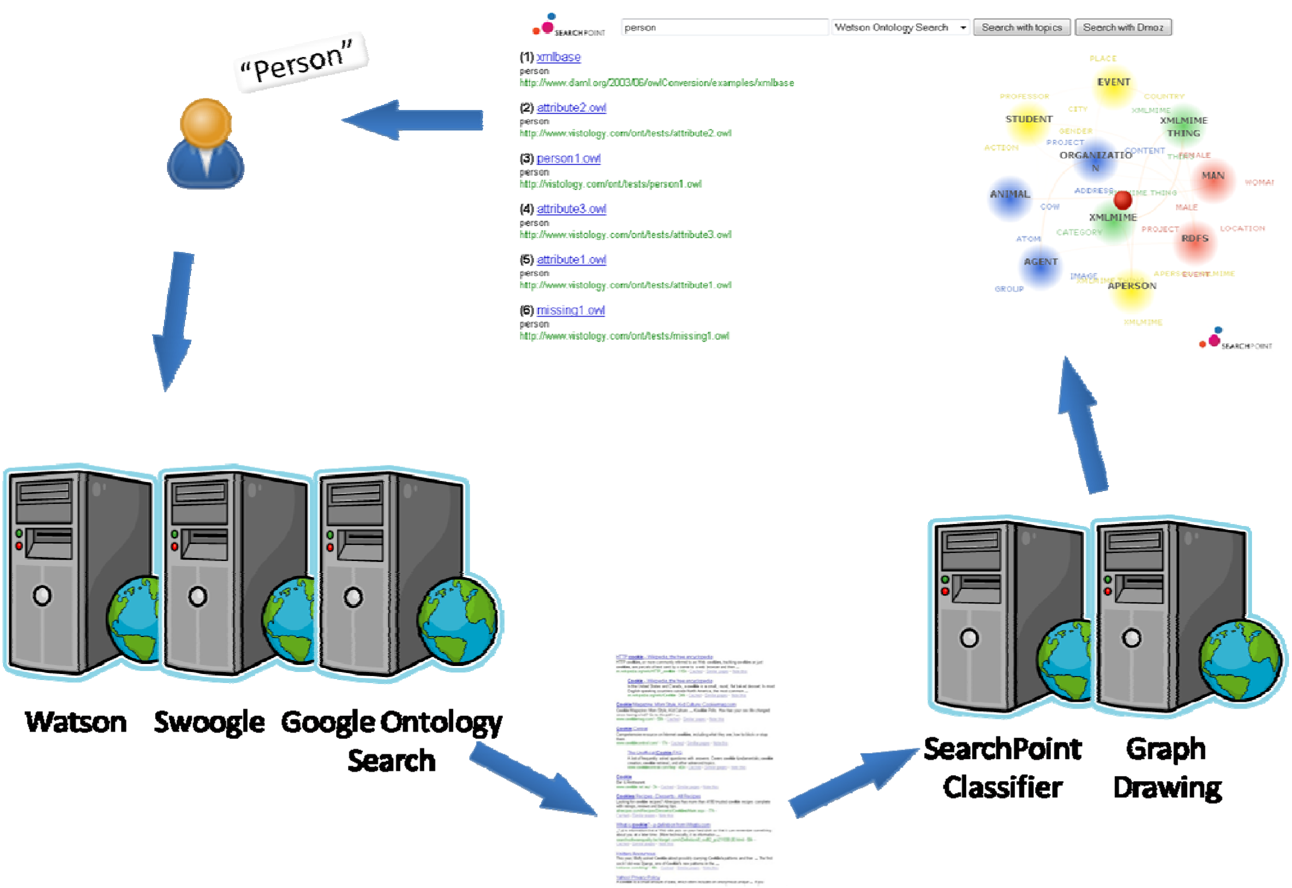

Figure 1: Architecture of the system

2.2 Descriptions of modules and improvements

There have been several improvements implemented in most of the modules. Some were necessary in order to make integration possible, some were made in order to improve the accuracy of automatic topic extraction and classification and some were made with regard to improving usability of the application.

2.2.1 Integration of Watson ontology search engine

The first goal of this deliverable was to integrate Watson search engine with the complementary contextualization of search results provided by SearchPoint. Since SearchPoint works on top of any search engine that is providing textual results, Watson first needed an adaptation in order to make integration possible.

It was therefore decided to textually annotate each ontology that Watson searches for the notion of key concepts. This textual summarization of ontologies was made possible in D4.5.4 [Dzbor, 2009] and we restate some of the definitions here:

The notion of *key concept* is highly subjective, but one easily follows the rationale of choosing *n* 'best descriptors' of the content of a particular ontology, where *n* is a number smaller than the total number of concepts defined in the ontology. The motivation of key concept identification was, to some extent, to reflect the role of so-called natural categories of the cognitive science.

The two core cognitive measures in choosing key descriptors for an ontology are: (i) concept centrality and (ii) label simplicity. The former is common e.g., in social network analysis, where it indicates how many times a given node appears in all paths between the root(s) and leaves. The latter measure reflects the number of compound terms forming the key concept label. Whereas the algorithm maximizes the former measure, the latter measure is minimized (preferring simpler names to more complicated ones).

Two additional measures are considered in the identification of key concepts: (iii) concept density and (iv) concept coverage. The former reflects how richly (frequently) a given candidate concept is described in the ontology in terms of its individuals, sub-classes, etc. The latter measure carries the usual meaning, trying to maximize the number of concepts in the entire ontology that belong to the branches headed by a particular key concept candidate.

Key concepts on their own are a good approximation of the ontology content. They essentially form a list, a vector of labels. One can then choose a vector with five, ten, twenty or more items to reflect the breadth and granularity of coverage. In integration with SearchPoint this is all that is needed as text mining methods used for classification and clustering transform text into vector models, so key concepts are perfect for this task.

Watson ontology search equipped with key concepts can be accessed as a rest web service at:
http://kmi-web06.open.ac.uk:8081/watson-sp/search?f=xml&q=publication

The second parameter (q=publication) can be used to pass a URL encoded query. By default, Watson search returns two hundred results or the maximum ontologies found. An actual example of an xml rest response can be found in Appendix A.

If the first parameter (f=xml) is omitted, we get a simple web application of the first integration of Watson and SearchPoint, as seen in Fig 2. This integration is still without other improvements and can be used as a point of reference.

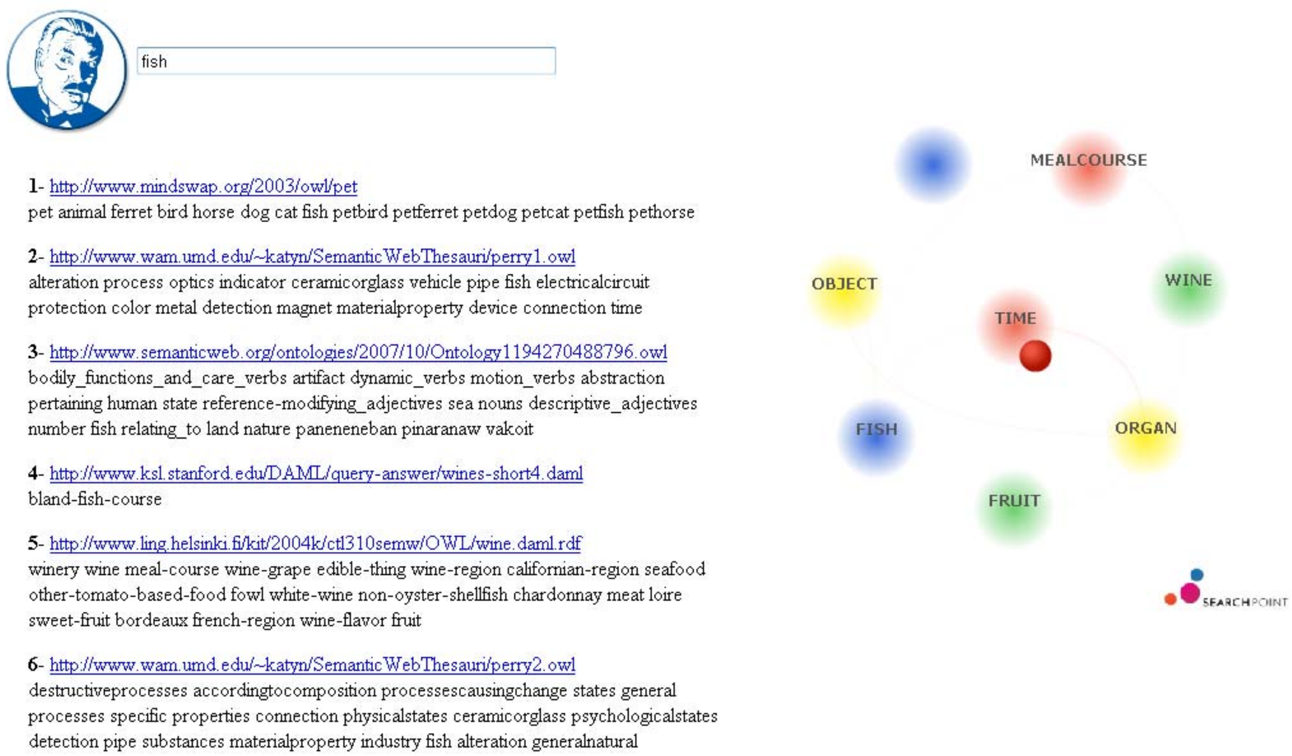

Figure 2: The first Watson - SearchPoint integration

2.2.2 Topic generation - Clustering revisited

The most obvious way to automatically generate topics out of a corpus of documents is to cluster the available results into a predefined number of clusters and use a centroid or medoid of each cluster as an individual topic. *Document clustering* [Steinbach et al., 2000] is based on a general data clustering algorithm adopted for textual data by representing each document as a word-vector, which for each word contains some weight proportional to the number of occurrences of the word.

Before this, there is a pre-processing phase, where we discard all the common words frequently found in most of the documents. Examples of such stopwords include: 'IS', 'A', 'THE', 'HE', 'SHE' etc... This pre-processing step was already implemented in the initial SearchPoint implementation in which 523 stopwords are defined.

To improve this in the ontology search scenario, we have identified seven new ontology related stopwords: 'OWL', 'RDF', 'DAML', 'WWW', 'DEF', 'ORG' and 'HTTP'. All of these new stop-words could be added into the underlying library for text-mining at compile time. However, we wanted to boost the process further by adding the words from the query, which can only be added at run-time.

Namely, search result set is a bit different from a general corpus; especially when it comes to text-mining. Search engines usually return results that contain all the query words. Therefore each hit must contain every query word, which can be quite rare, producing poor results and badly affecting the clustering of the results into topics. In practice, the query word or words amounted to at least one cluster-topic that is rather unhelpful to the user who is primarily expecting sub-topics. Therefore we have implemented the adding of additional stop-words at run-time.

We break down the query to single words and add each word as a stop-word. All these words are ignored during the clustering and the final result returns a more accurate clustering where topics are free of query words which were previously identified as distraction for the user.

2.2.3 Visualization of the Ranking Space

We have improved the graphical user interface of the *ranking space*. First, we restate the short definition and motivation for ranking space from D3.2.4, then we state the problem and finally present our solution accompanied by screenshots.

The main goal of ranking space is to visualize automatically generated topics, positioned on the screen in such a way that similar or related topics lay close together. Apart from this the user can select a *focus point* by dragging a red point that is also a part of the GUI.

Once this is achieved, it is possible to calculate new rankings of the documents for each position of the focus point. Since each point on the panel maps to a specific ranking, we call it the *ranking space.*

Ranking space defined in such a way gives the user options for exploring subtopics and defining a contextualized ranking for the search query. The problem remaining is that of presenting problematic cluster-topic annotation in such a way that it is understandable to the user.

Each topic is a centroid of a cluster and can be represented as a weighted vector of words. Most highly weighted words are most representative for the cluster. Topics/clusters are visualized as painted disks of different colours, with the most prominent word printed on top. However, the most prominent word is not necessarily most recognizable to the user, even more frequently, one word alone is usually not enough to fully understand the cluster.

However, since there are ten topics visualized on the screen, there is a problem when visualizing more than one word per cluster. We have now solved this, by using colour coding and user interaction to provide the user with a clear distinction between the additional description words and its parent clusters.

As an example we demonstrate the new visualization of the ranking space over the query "publication" given to the Watson search engine (Figure 3). There are ten subtopics found and visualized as coloured and labelled discs. Apart from this, each topic is surrounded with three additional words of a similar colour to the disc of the originating cluster.

As an additional help for the user, to correctly identify cluster and also to make the whole visualization more readable, it is possible to rotate the three additional words around the parent cluster. On mouse-over of any cluster, the three words start to rotate, which immediately clarifies which are the cluster words, as well improving the readability in case any of the words intersect with each other or perhaps fall outside the visualization screen.

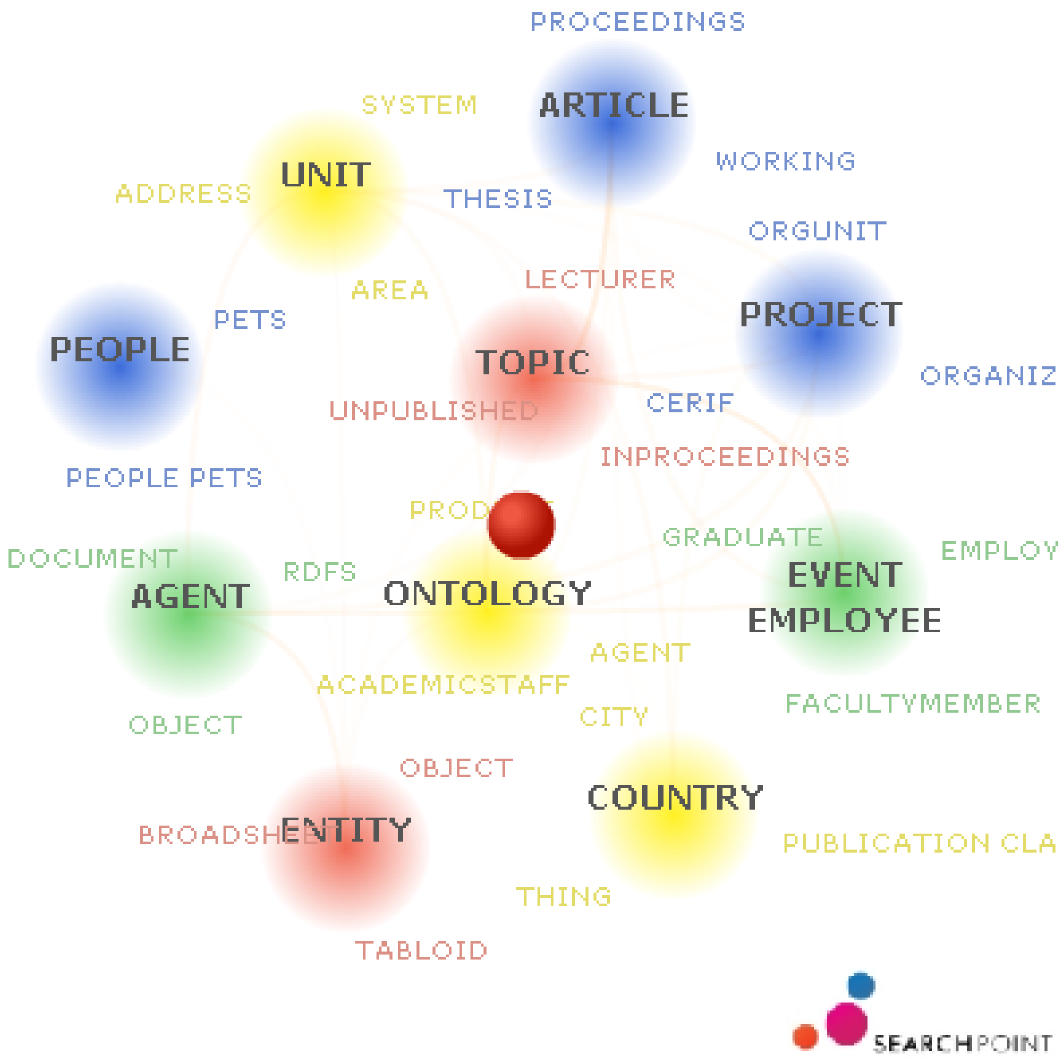

Figure 3: Visualization of the ranking space

2.2.4 Single Point of Access

All of the mentioned modules with the new improvements are now also available as a single application. The GUI provides a place for the input of the query as an input field: selection of the ontology search engine as a select box and selection of the SearchPoint method to use in the form of submit buttons. (Figure 4)

Once the query is submitted, the results are returned in a ranked resultset accompanied by the new ranking space visualization which enables the user to rerank the results and analyze subtopics of the query and resultset. (Figure 5, 6)

Figure 4: Default view of the unified SearchPoint for Ontology Search

2.3 Distribution

Due to the application consisting of several modules running on different servers, a service oriented architecture paradigm was used at the time of implementation of the application. This also makes it easy to make different distributions of the application where needed. The only requirement is working internet access in order for the application to work.

2.3.1 Web Application

The first and most natural way to distribute the application is in the form of a web application. All of the modules and improvements discussed so far can be found at:

http://searchpoint.ijs.si/ontologysearch

In Figure 5 we can observe the initial ontology results for the query "professor" submitted to the Watson ontology search engine. At the top a new query can be inputted, a different search engine can be selected from the select box, or a different topic extraction method can be used by pressing the corresponding button.

On the bottom left, there is a ranked list of results, ten per page (only five are visible in the screenshot) and on the bottom right, there is a visualized ranking space. Additional keywords for topics can be moved by hovering over the appropriate topic. The red focus point can be moved in order to dynamically rerank the left result set.

The reranking takes place completely on the client side, no additional calls are made to any server.

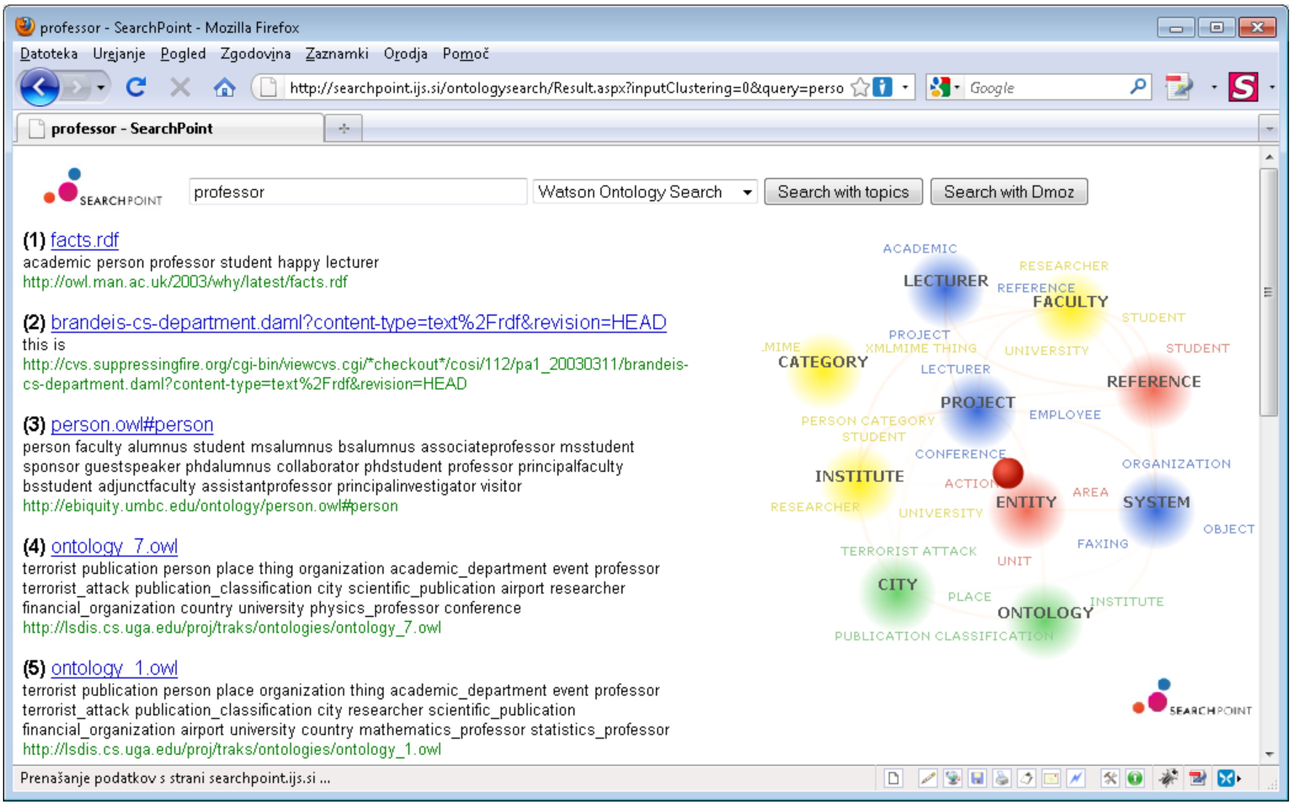

Figure 5: SearchPoint Ontology Search web application

2.3.2 NeOn Toolkit plugin

As defined in D5.3.1 ontology search is a first step in the ontology reuse cycle - an important functionality that needs to be present when dealing with networked ontologies. Currently, NeOn toolkit only provided Watson ontology entity search plugin, which searches only for the entities and not the complete ontologies. This does not satisfy all the needs, for example when whole ontologies or ontology modules are to be reused.

This is the main reason we have created a SearchPoint Ontology Search plugin for the NeOn toolkit. Both the binary version of the plugin and its sources can be found at:

http://kameleon.ijs.si/searchpoint/

To install the plugin, binary version (.jar) must be copied to the installed NeOn toolkit "plugins" directory and the NeOn toolkit needs to be restarted. To start the plugin, the user must navigate to Window -> Show View -> Other -> UI -> SearchPoint.

A new view in the toolkit is opened (Figure 6) and the user interaction is the same as before (Section 2.3.1)

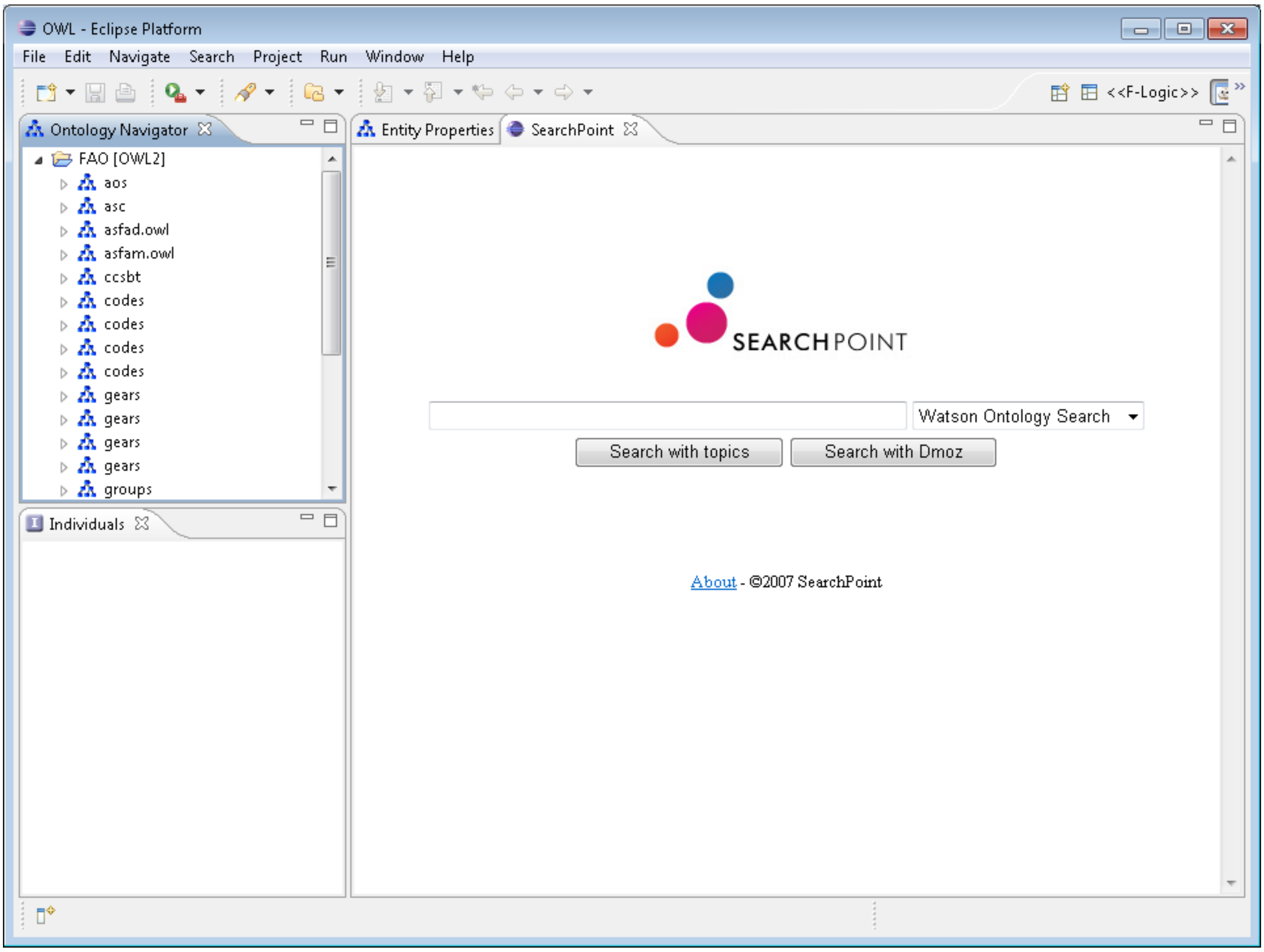

Figure 6: SearchPoint Ontology Search as a NeOn Toolkit plugin

3. Example usage of the system

Most of this section is an update from the D3.4.2 Section 3. Here we will briefly describe the basic functionalities of the prototype (Section 3.1) and in the end we will showcase one use case accompanied by screenshots (Section 3.2). Discussions about scenarios of use and usability can be reviewed in D3.4.2 (Section 3.2).

Everything demonstrated here applies both to the NeOn Toolkit plugin as well as the web application.

3.1 Basic description of the functionalities

GUI consists of an input field where the user can provide the query. There is a select box for selection of the search engine, the user can submit a query in several ways and each method for topic extraction is connected with one submit button. Apart from this, there is a result list and the ranking space, where the user will be able to choose the focus point.

Three search engines are currently supported:

- Watson: http://kmi-web06.open.ac.uk:8081/watson-sp/
- Swoogle: http://swoogle.umbc.edu/
- Google ontology search: http://www.google.com

After the user inputs the query and selects the search engine and method for topic extraction the result list is returned. At the beginning the results are ranked the same way as they are returned by the search engine. Each result item is visualized together with the original ranking. At the start this is therefore (1, 2, 3, 4…). The user can now inspect the topics for the current search and select any point on the ranking space or drag the focus point around the ranking space. The hits get reordered in real time, so the user is able to pinpoint a good selection for the focus easily, by just observing the quality of the shown results.

There is also a history of the positions of the focus point. In case the user has found a good position and then moved the focus to see other rankings it is easily possible to navigate to a previous position by clicking on one of the three buttons of the history bar.

In the history bar there are three positions:

- Back: for returning the focus to one step back
- Forward: to move it to the next position
- Origin: to clear the history and return to default ranking provided by the search engine.

The buttons are only visible when they are available. For example the Forward button is not visible until back has been clicked. No button is visible at the start, or upon clicking origin button, since there is no history yet at that time.

3.2 Showcase

In the showcase we will demonstrate the contextualization power of SearchPoint. The main goal is to contextualize search over a given background knowledge with the use of automatic topic generation techniques and re-ranking possibilities of the ranking space.

The scenario will be that of a user set with the task of finding the most suitable ontology for reuse concerning humans which will be integrated into a bigger ontology that models a person as a biological entity together with animals, plants and other life.

The user first starts the NeOn toolkit and brings up the view of SearchPoint Ontology Search (Figure 6). The query "person" is input to the Watson search engine with topic extraction method.

A result list of ontologies containing "person" concept is returned and accompanied by the ranking space (Figure 7). The top 8 results only contain the URL of the ontology and a single key concept "person". This by itself is highly uninformative to the user who would need to check out the actual ontology, view lower ranked hits, or redefine his query.

Ranking space on the other hand immediately profiles the different clusters of ontologies that contain “person”. For example, we have following sub topics with additional keywords and qualitative analysis of the actual ontologies:

- **Organization**, Address, Project, Content: Modelling organizations and everything connected with it
- **Man,** Woman, Male, Female: Modelling the human as a biological being
- **Animal**, Cow: Modelling biological and fictional animals, creatures
- **Student**, Office, Gender, Action: Modelling a specific domain of colleges
- **Event**, City, Place, Country: Modelling events
- **RDFS**, Event, Project, Location: Modelling projects
- And additional five clusters as can be seen in Figure 7

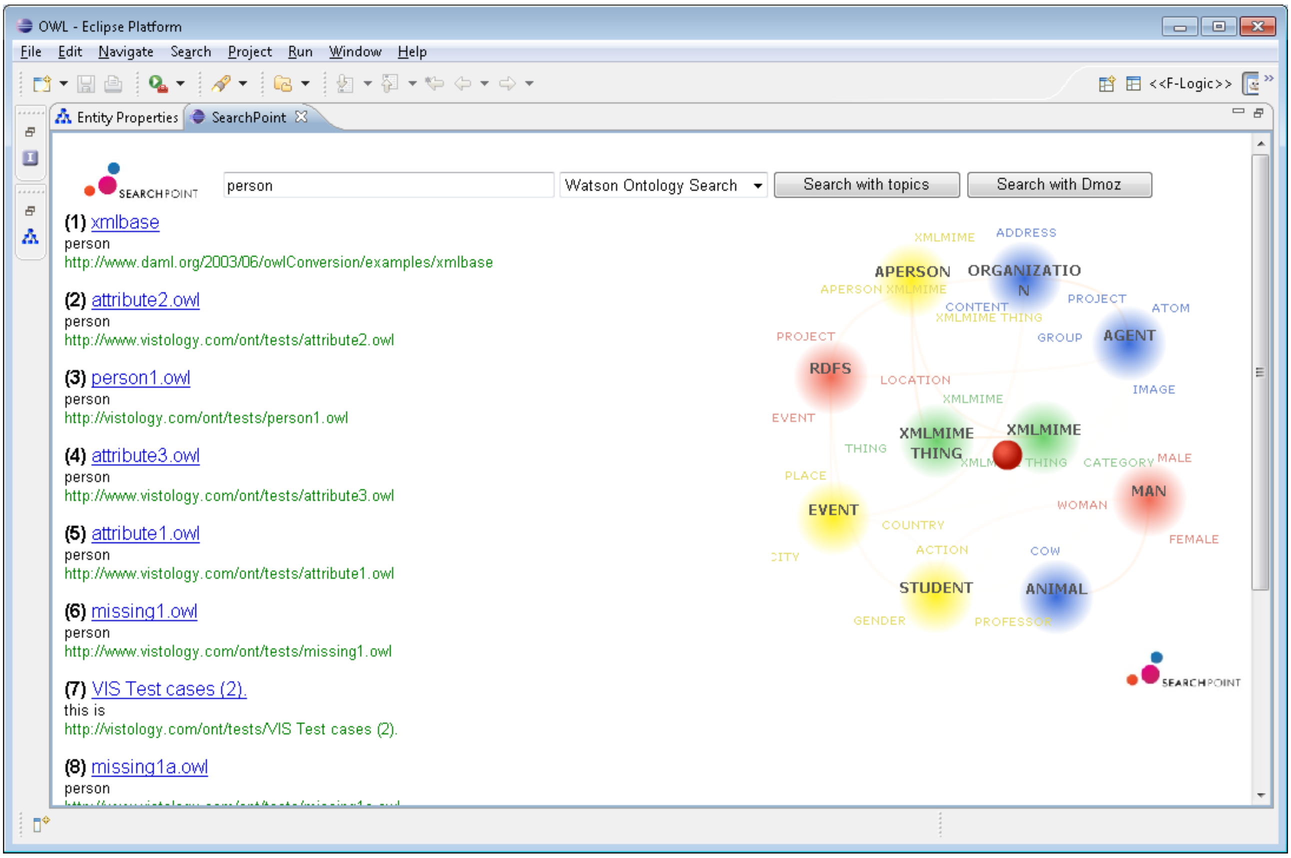

Figure 7: Ontology Search for the query "person"

The new visualization of ranking space immediately conveys the high level picture of each topic. Since our user is interested in person as a biological entity he moves the focus point in between the most relevant topics "Man" (Woman, Male, Female) and "Animal" (Cow). (Figure 8)

The top results (see below) actually give the best ontology results for the given tasks.

- (26) ont1.daml: animal male female person plant man woman
- (33) rdfg3a.rdf: person animal man woman class
- (34) rdfg3b.rdf: man animal person woman class
- (32) ont3.daml: animal person male person_with_two_sons...
- (81) no file name: person animal man ... woman tall_thing male ... human_being ...

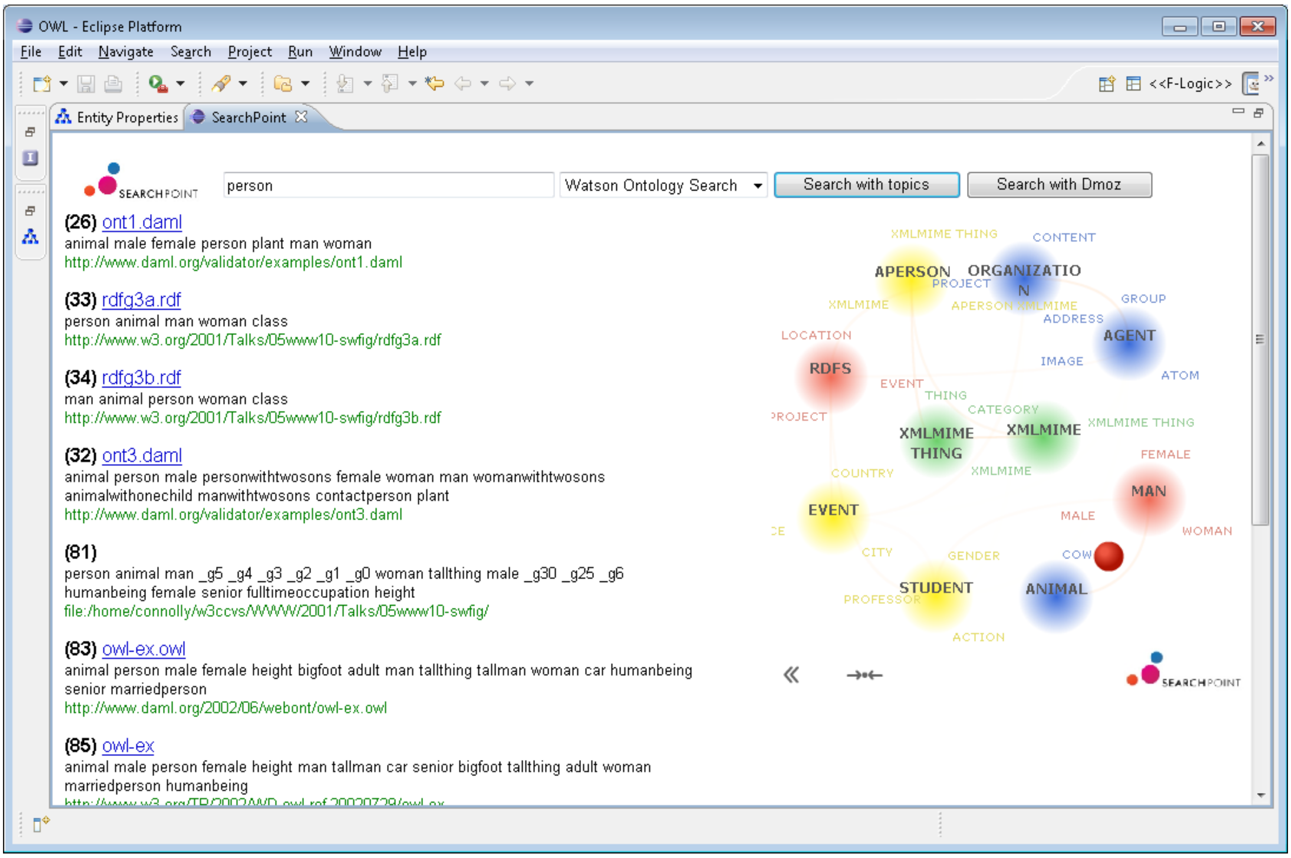

Figure 8: Focus point moved between "animal" and "man" topics

4. Conclusion and Discussion

In this work we have integrated SearchPoint with Watson search engine, extended Watson with key concepts to make integration with SearchPoint possible, improved both the server side SearchPoint webservice in order to permit run-time passing of stopwords and improved the graphical user interface of the SearchPoint client side.

We have created a single unified application that binds all the various modules and makes contextualized ontology search over several search engines and different methods possible. We have distributed this application both as a NeOn Toolkit plugin as well as a web application.

Careful thought has been given in order to enhance the human-computer interaction. A lot of information is interactively presented to the user in a non obtrusive way with techniques to make the large amount of data distinct and comprehensible.

This deliverable also fulfils one of the contextualizing aspects promised in WP3. Ontologies can now be searched in the context of a general background knowledge or other grounded ontologies. This has been the effort of several functionalities, starting with summarization of ontologies with key concepts (D4.5.4). Watson search can provide basic search functionality over these annotated ontologies and this search can be contextualized with the help of SearchPoint (D3.2.4).

Appendix A: Watson XML rest response for the query "publication"

Bellow is presented the top portion of the resultset that Watson rest web service returns. The xml consists of <Result /> tags for each ontology result returned. The tag is annotated with parameters id for identification and URL for the link to the ontology. Textual content of the tag are the precomputed key concepts.

Other search engines providing a similar output to Watson ontology search could be immediately integrated into the SearchPoint as an additional search engine.

Only the first five results are shown below:


```
<StringResults xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://searchpoint.ijs.si/StringResults.xsd"
xmlns:wicket="http://wicket.apache.org/" >
<Result id="1"
url="http://lsdis.cs.uga.edu/projects/semdis/swetodblp/january2007/opus_january2
007.rdf">publication thesis edited_publication serial_publication journal
university school article_in_proceedings edited_book book_chapter series
proceedings book masters_thesis webpage doctoral_dissertation
publishing_organization article</Result>
<Result id="2"
url="http://lsdis.cs.uga.edu/projects/semdis/swetodblp/october2006/opus_october2
006.rdf">publication edited_publication thesis school serial_publication
proceedings edited_book series book university article_in_proceedings
book_chapter journal doctoral_dissertation webpage publishing_organization
article masters_thesis</Result>
<Result id="3"
url="http://korrekt.org/w/index.php?title=Special:ExportRDF/The_Two_Cultures_%28
JWS2008%29&xmlmime=rdf"></Result>
<Result id="4"
url="http://korrekt.org/w/index.php?title=Special:ExportRDF/All_Elephants_are_Bi
gger_than_All_Mice&xmlmime=rdf"></Result>
<Result id="5"
url="http://ebiquity.umbc.edu/v2.1/ontology/publication.owl#publication">
</Result>
....
</StringResults>
```

References

[d'Aquin 2008] d'Aquin, M. Building Semantic Web Based Applications with Watson. 17th International World Wide Web Conference (WWW2008) Developers Track.

[Dzbor 2009] Dzbor, M., Peroni S., Motta E., d'Aquin M. D4.5.4 NeOn Toolkit plug-in for visualization and navigation in ontologies and ontology networks based on concept summarization and categorization

[Grobelnik et al., 2008] Marko Grobelnik, Janez Brank, Blaž Fortuna, Igor Mozetič. Contextualizing Ontologies with OntoLight: A Pragmatic Approach, Informatica. 2008.

[Pajntar and Grobelnik, 2008] Boštjan Pajntar, Marko Grobelnik. SearchPoint – a New Paradigm of Web Search. 17th International World Wide Web Conference (WWW2008) Developers Track.

[Pajntar et al., 2009] Boštjan Pajntar, Dunja Mladenić. Marko Grobelnik. D3.2.4 Context-sensitive search of ontologies

[Sabou et all, 2006] M. Sabou, M. d'Aquin, and E. Motta: Using the Semantic Web as Background Knowledge for Ontology Mapping, In *Proceedings of the International Workshop on Ontology Matching (OM-2006)*, collocated with ISWC-06.

[Steingbach et all, 2000] Steinbach, M., Karypis, G. and Kumar, V. (2000). A comparison of document clustering techniques. Proc. KDD Workshop on Text Mining. (eds. Grobelnik, M., Mladenić, D. and Milic-Frayling, N.), Boston, MA, USA, 109–110.

[Suárez-Figueroa, 2007] Suárez-Figueroa M.C., Cea G.A., Buil C., Caracciolo C., Dzbor M., Gómez-Pérez A., Herrrero G., Lewen H., Montiel-Ponsoda E., Presutti V. D5.3.1 NeOn Development Process and Ontology Life Cycle