

Visualising Domain Ontologies Using Mindmaps to Enhance Requirements Engineering

Rami Zayed

University of the West of
England Bristol and Airbus /
United Kingdom
ramizayed@gmail.com

Mario Kossmann

Airbus / United Kingdom
mario.kossmann@airbus.com

Mohammed Odeh

University of the West of
England/ United Kingdom
Mohammed.Odeh@uwe.ac.uk

Abstract: This paper sheds light on the relevance of domain ontologies in the contexts of Systems Engineering (SE) and Software Engineering (SwE), and in particular on their importance for the Requirements Engineering (RE) process in both contexts; and it explains both the importance of and obstacles to visualising ontology content in a meaningful way so that effective communications with stakeholders based on relevant domain ontologies can be greatly enhanced.

One powerful yet flexible way to visualise ontologies is by means of mind maps. However, the automatic data transfer between the owl format and a standardised mind map format that can be viewed and edited by commercial mind mapping tools had not been enabled until recently. The OntoREM Mind-Mapper (OMM) tool was developed to bridge this gap between domain ontologies that are specified in OWL and mind mapping formats, in order to enhance the Ontology-driven Requirements Engineering Methodology (OntoREM). In addition to this primary goal of the OMM tool, it also enables the visualisation of any other ontology that is specified in line with the OWL notation, subject to compatibility with the used baseline of the OWL standard.

Keywords -Ontology; Mind mapping; Ontology Visualisation; Requirements Engineering;

1. Introduction

In many areas such as in the aerospace, pharmaceutical and medical, electronic consumer, e-commerce and logistics, as well as legal contexts, increasing numbers of experts have started to formally specify their domain knowledge to enhance knowledge transfer, avoid misunderstandings and

increase re-usability of knowledge for multiple programs or projects. One very popular way to do this is by means of ontologies in the web ontology language (OWL) notation, which bears the advantage of being still readable by humans, albeit with the needed level of understanding of ontologies, while being processable by machines. In other words, the knowledge can be specified in a way that is at the same time understandable to humans and interpretable by machines. The latter is important for all different types of analyses and automatic exploitation of the knowledge stored, for example using inference engines.

In the field of philosophy, Gregory et al. defined ontology as “*The study and description of ‘being’, or that which can be said to exist in the world*” [10]. In more recent analytical philosophies, ontology is considered as the study of *what is* – in relation to existence. However, according to Swartout and Tate (1999), the term has been borrowed by the artificial intelligence community to define “*A set of concepts or terms that can be used to describe some area of knowledge or build a representation of it*” [18]. Another well-formed definition of ontology is the one given by Gruber (1993): “*An ontology is an explicit specification of a conceptualization*” [3, p. 1], where *conceptualization* refers to the abstract understanding of the part of the world (i.e. domain) that is to be described for some reason. According to Gruber (1993), the set of entities (i.e. universe of discourse) in a given domain and their relationships and attributes should be explicitly described in the ontology representing that domain, using a declarative formalism [4]. Gasevic et al. (2006) state that ontology can be seen as a representational vocabulary of a given domain, in which the domain terminology, vocabularies, axioms and concepts including their relations, constraints and taxonomy

and hierarchies are declared and explicitly defined [1].

However, not least due to the enormous richness of the information that can be stored in a domain ontology, currently domain ontologies are still difficult to visualise in a meaningful way. There is so much information available that any visualisation showing all information available would be unreadable. Only limited aspects of such ontologies can be meaningfully displayed, which requests users to make the right selections in terms of what should be displayed, and how. This means they have to be familiar with the content and structure of the domain ontology and with the suitable ontology editor and visualisation tool to make a reasonable pre-selection of what should be displayed.

One powerful yet flexible way to visualise ontologies is by means of mind maps, using one of the available mind mapping tools. However, the automatic data transfer between the owl format and a standardised mind map format that can be viewed and edited by such mind mapping tools has not been enabled until recently. The OntoREM Mind-Mapper (OMM) tool was developed to bridge this gap between domain ontologies that are specified in OWL and mind mapping formats, in order to enhance the Ontology-driven Requirements Engineering Methodology (OntoREM). Although the OMM tool was developed with a focus on building and maintaining domain ontologies including domain requirements following the OntoREM approach, it also enables the visualisation of any other ontologies that are specified in line with the OWL notation, albeit with certain limitations depending on the baseline of the OWL standard used to specify the ontology in question.

In the following, the paper will shed light on the relevance of ontology in the contexts of Systems Engineering (SE) and Software Engineering (SwE), and in particular the importance of ontology for the Requirements Engineering (RE) process in both contexts. Then, the concept of mind maps will be introduced and the advantages of ontology visualisation, specifically in mind map formats, will be explained. Finally, example visualisations of domain ontology in the context of knowledge-driven

RE will be illustrated that were created applying the OMM tool.

2.Ontology in the Contexts of Systems Engineering and Software Engineering

The interest in ontology amongst computer scientists first began when developers and engineers became more interested in sharing and reusing knowledge across systems. One of the significant obstacles, which has been constantly facing systems and software engineers when sharing or reusing knowledge, is the lack of shared terms with explicit semantics [3]. More often than not, different systems use different terms and concepts to describe domains; they might even use different terms to describe the same piece of knowledge. For example, one system might use the term “United Kingdom” to refer to the UK, whereas another might use the term “Great Britain”. This has caused difficulties in holding a common shared understanding of all the concepts of a certain domain among software engineers or agents, let alone the disability to take some general knowledge out of one system and reuse it in another [18]. As a result the need for ontologies and their applications in computer science has increased and the idea of building universal ontologies that can act as underlying knowledge bases for various application systems has been adopted and encouraged by both software and systems engineers.

The primary purpose of such adoption is to allow developers of different systems to share and reuse a common reliable understanding of the domain at hand while explicitly defining the terms and concepts used in that domain [18]. In other words, ontology offers a formal systematic way to generate a general standard terminology for any given domain to be utilised amongst systems and software engineers as a validated knowledge database, which greatly enhances communications between developers of different systems and enables the sharing of knowledge between different software applications. In Systems Engineering (SE) and Software Engineering (SwE) developing ontologies is not a goal in itself. In fact, ontologies serve as knowledge repositories that are used to feed multiple systems and software agents when solving problems in associated domains [1].

Broadly speaking, there are obvious overlaps between what an ontology engineer does when defining domain knowledge and what a requirement engineer does when eliciting requirements from stakeholders [5]. Because of this, Breitman and Prado Leite (2003) tend to believe that the responsibility for developing and implementing ontologies belongs to requirements engineers, and they therefore classify the development process of ontologies as a sub-process of the whole RE process [2]. Figure 1 shows a diagram illustrating how ontologies can be reused to improve RE.

applying ontologies in RE significantly reduces the negative effects of [4]:

- Ambiguous requirements, which require repetitive work and hence cause time loss.
- Inadequate specifications, which lead to missing key requirements.
- Not fully explained requirements, which cause poor understanding and lead to optimistic expectations.
- Dynamic and changing requirements, which need regular revision in order to be understood and satisfied.

4. What is a Mindmap?

A mindmap can be seen as an organisational structure that radiates from a central concept using key words, numbers, branches, symbols, colours and images resulting in a graphical representation of one's

thoughts and ideas [3]. Put simply, mindmaps are graphical illustrations of thoughts and ideas based on the concept of Radiant Thinking: a term coined by Tony Buzan to describe the way the human brain processes information [3]. A typical mindmap would contain images, drawings, pictures, symbols, shapes and even words and letters formed as pictures.

In order to create a mindmap, one should start with a central topic or concept and then radiate the Basic Ordering Ideas (BOIs) (i.e. the first-level key topics) out from the topic of attention using different colours. After that one can start hierarchically expanding the BOIs using branches, sub-branches, keywords, symbols and drawings [3]. Every mindmap is infinite and can constantly branch off generating multiple associated mini mindmaps just like the human brain [3]. Figure 2 represents an example of a mindmap.

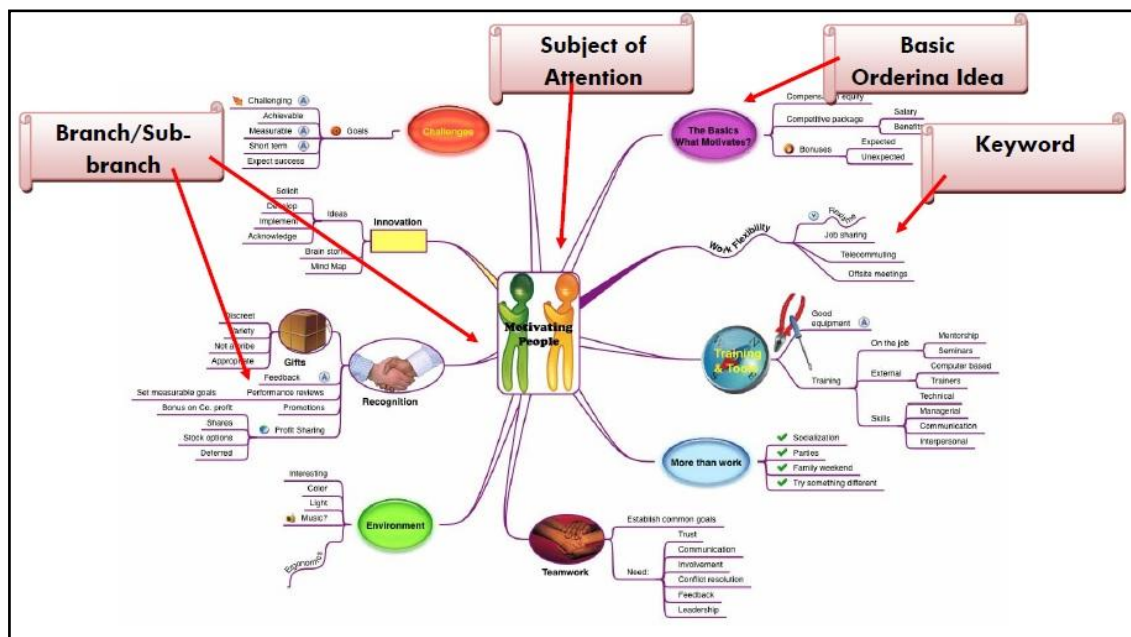


Figure 2: Example view of a mindmap [9]

5. The Advantages of Ontology Visualisation

As mentioned above, the visualisation of ontologies, or rather aspects of ontologies that are of immediate concern, is at the same time difficult and essential in order to communicate with the many stakeholders in the RE process. Mindmaps are powerful graphical representationsto visualise ontologies and such

advanced visualisation is still not yet implemented in commercial software nor provided by ontology management GUIs such as Protégé. The lack of such advanced means to visualise ontologies is one of the reasons that have made ontology databases relatively less acceptable by business stakeholders and non-technical managers. On the other hand, mind-mapping techniques have been used in a wide range of disciplines, by all kind of users: expert and non-

expert users, and supported by a number of highly advanced and powerful software tools.

OWL ontologies usually hold all the required knowledge of the domains they represent. They specify the essential domain knowledge, including the domain concepts, concerns, elements, relationships etc. Domain ontologies tend to keep evolving during their development lifecycle and they are constantly changed by domain experts to correctly reflect the domain knowledge and hence hold accurate, complete and shared understanding of the domain at hand. Using an Ontology Editor, such as Protégé, makes the gathering, updating, validating and tracing of the knowledge of the domain at hand very difficult and challenging. This is where mindmaps become of high importance and provide useful solutions for visualising the gathered domain ontology, in particular those aspects that have to be communicated to and discussed with relevant stakeholders.

This allows domain experts to easily capture and share the big picture of what has been defined in the ontologies while enabling them to instantly dig deeper for more specific details as needed. In addition, mindmaps provide much more convenient means to update and validate the previously collected domain knowledge with the relevant stakeholders as they simplify the complexity found in OWL ontologies and the existing editors, such as Protégé, by graphically visualising all the links and connections between all the ontology elements, including the classes, sub-classes, and associated data type and object properties.

Moreover, mindmaps are formidable communication tools that are easily understood and exchanged by non-technical managers and non-expert teams, which encourage domain experts and stakeholders to participate in a very effective way during the ontology development process of the domain at hand. In addition, mindmaps stimulate the human brain and hence increase the ability to brainstorm, memorise, organise and share domain knowledge and ideas, which is not the case when using ontology editors. This especially facilitates the validation and verification of the gathered domain knowledge with the relevant stakeholders and domain experts.

Furthermore, mindmaps reduce the potential of misinterpreting the concepts of the domain at hand by fostering validated understanding of the domain terminology amongst all the stakeholders involved before importing the developed ontologies into the OWL ontology files. They also significantly decrease the time needed to discover and repair any misunderstanding in the gathered knowledge as they efficiently assist in visualising the domain concepts and their relationships, picturing together all the connections between ontology elements, linking all the defined instances to their associated classes and properties, showing the dependency between the defined classes, properties and instances, representing the big picture of the class hierarchies and allowing to trace the gathered knowledge back and forth to its related high-level concepts. Figure 3 contrasts two representations of an example domain ontology, one in the ontology editor Protégé [17] and one using a mind map format provided by the Mind Manager tool (right) [12].

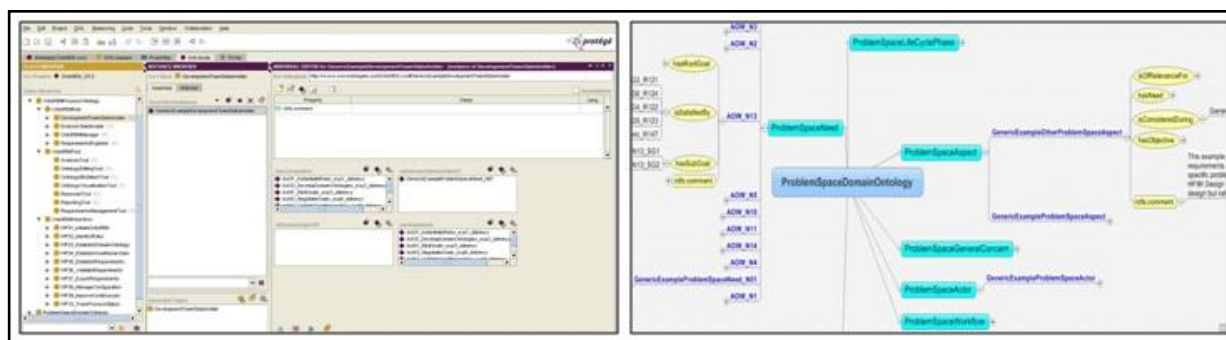


Figure 3: This diagram contrasts the OntoREM domain ontology meta model shown in both Protégé [17] and Mind Manager [12]

6. An Example of an Ontology Mindmap from the Context of ontology-driven RE

OntoREM is a novel Ontology-driven Requirements Engineering Methodology that essentially depends on a knowledge-driven, as opposed to merely process-driven, approach to RE [10]. OntoREM is a comprehensive ontology that includes knowledge from the RE domain as well as validated knowledge about the problem and solution domains at hand. A

reverse engineering process was applied to the OntoREM methodology and its associated processes and ontologies in order to map the OWL elements used in OntoREM to corresponding mind-mapping elements. This was achieved by defining a set of information transformation rules for deriving mind maps that correspond with aspects of the underlying OntoREM ontologies. Table 1 describes the mapping of the elements from the OntoREM Meta model ontology and how they will be represented in the mind map format.






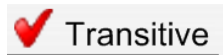


Generated Elements (Mindmaps)	Mapping Concepts (Ontology)
	A rounded-rectangle with turquoise fill colour indicates a class
	A hexogen with pale yellow fill colour indicates a datatype property
	An oval with pale yellow fill colour indicates an object property
	A bold blue font with no shape indicates an instance
	A normal black font with no shape indicates a property value
	The red-check icon indicates that whatever follows it is selected
	A yellow circle named Domain indicates domains of properties
	A yellow circle named Range indicates ranges of properties

Table 1: The defined mapping for visualising OntoREM Ontologies using mind maps

The defined set of transformation rules was implemented in the OMM tool (OntoREMMind- Mapper), to automatically visualise the OntoREM ontology by generating corresponding ontology mindmaps. Figure 4 shows the structure and relationships used to visualise ontology by means of

mind maps and explains the four levels that correspond to the classes (first level), instances of classes (second level), object and data type properties (third level), as well as the selected property values (fourth level) taken from the domain ontology.

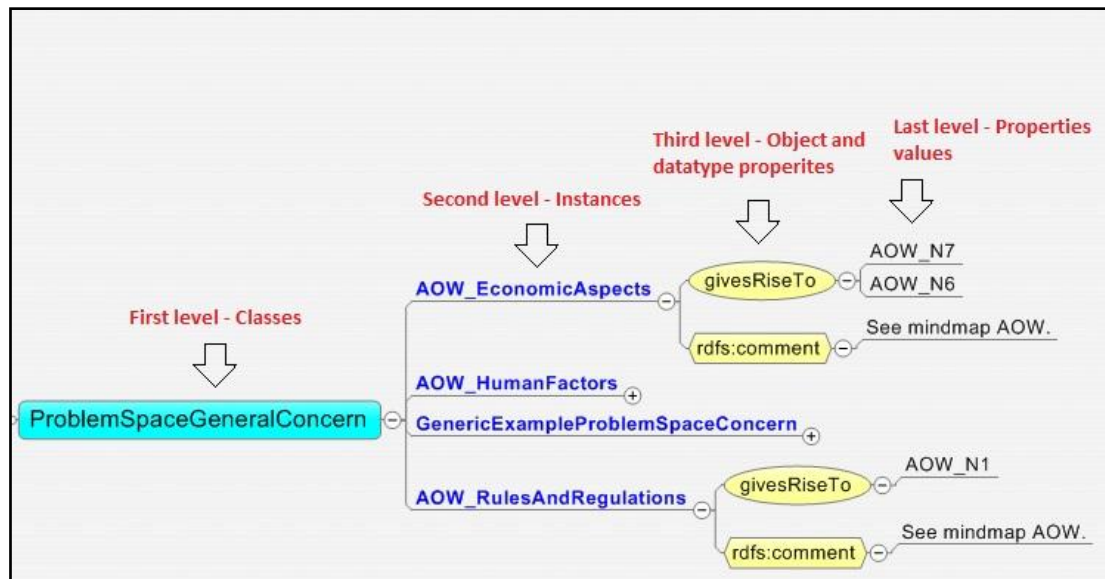


Figure 4: The structure and relationships used to visualise ontology using mindmaps

Figures 5 and 6 show high-level partial views of the OntoREM Meta model ontology – using the examples of the hierarchy of classes and the object

properties that are defined in the Meta model ontology.

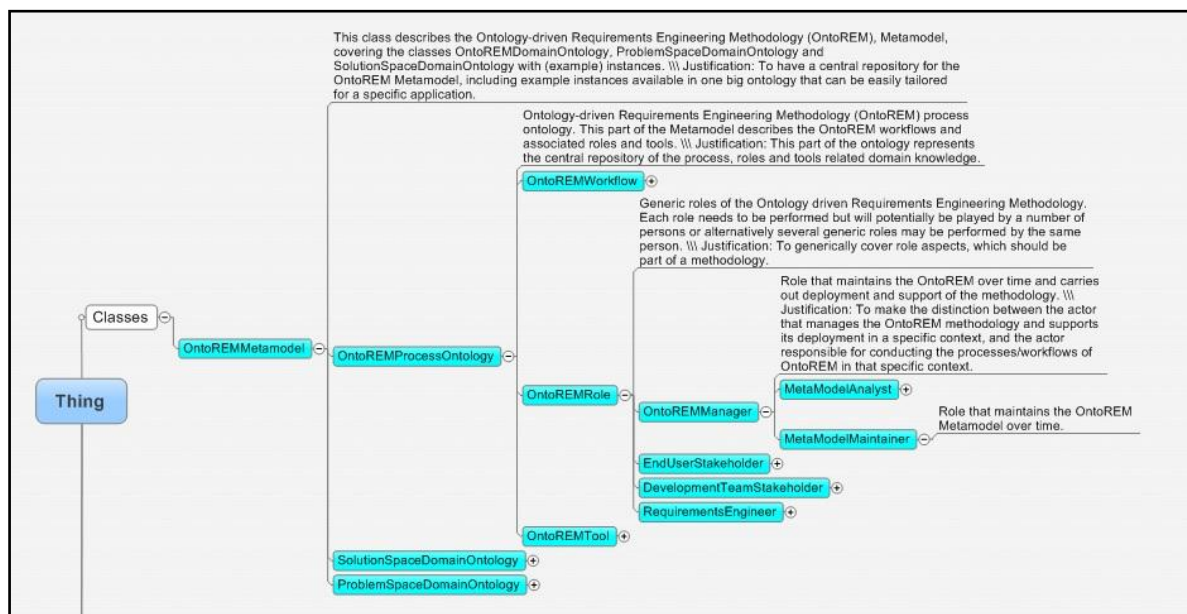


Figure 5:

Partial view of the generated mindmap of the OntoREM Metamodel (Class Hierarchy)

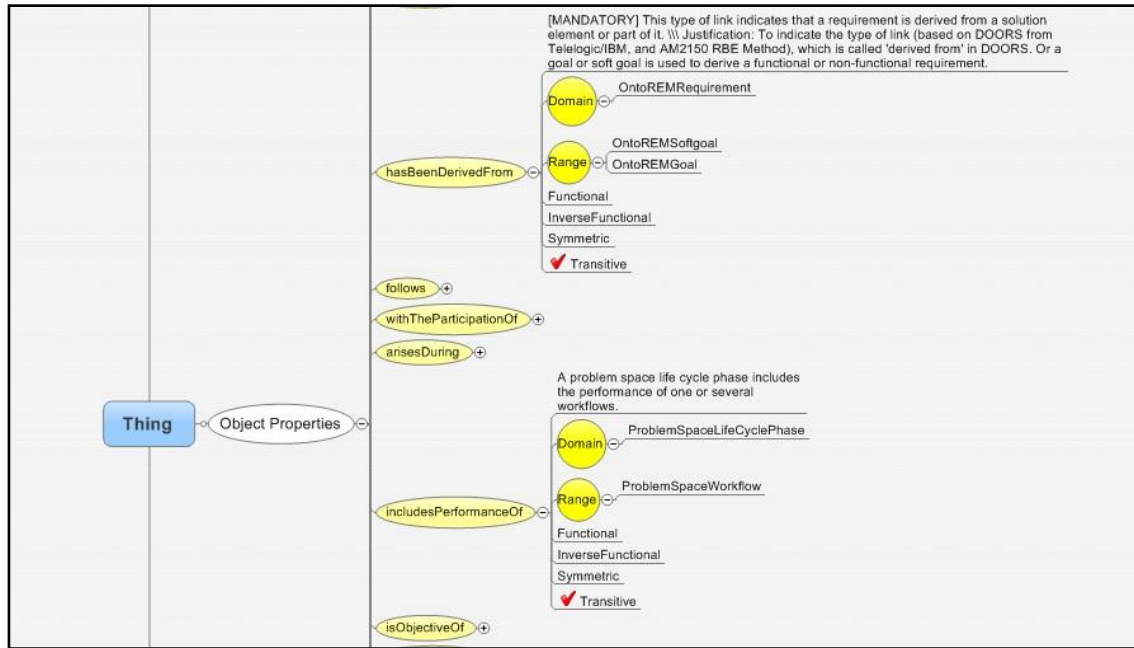


Figure 6: Partial view of the generated mindmap of the OntoREM Metamodel (Object Properties)

7. Conclusion

The paper shed light on the relevance of domain ontologies in the contexts of Systems Engineering (SE) and Software Engineering (SwE), and in particular on their importance for the Requirements Engineering (RE) process in both contexts. Then, the concept of mind maps was introduced and the advantages of ontology visualisation explained, in particular in mind map formats. Finally, example visualisations of domain ontology in the context of knowledge-driven RE were shown that were created by means of the OMM tool (OntoREM Mind- Mapper).

Due to the enormous richness of the information that can be stored in domain ontologies, they are still difficult to visualise in a meaningful way. There is so much information available that any visualisation showing all the information would be unreadable. Only limited aspects of such ontologies can be meaningfully displayed, which requests users to make the right selections in terms of what should be displayed, and how. This means they have to be familiar with the content and structure of the domain ontology and with the suitable ontology editor and visualisation tool to make a reasonable pre-selection of what should be displayed.

One powerful yet flexible way to visualise ontologies is by means of mind maps, using one of the available mind mapping tools. However, the automatic data transfer between the owl format and a standardised mind map format that can be viewed and edited by such mind mapping tools had not been enabled until recently. The OntoREM Mind-Mapper (OMM) tool was developed to bridge this gap between domain ontologies that are specified in OWL and mind mapping formats, in order to enhance the Ontology-driven Requirements Engineering Methodology (OntoREM).

Although the OMM tool was developed with a focus on building and maintaining domain ontologies including domain requirements following the OntoREM approach, it also enables the visualisation of any other ontologies that are specified in line with the OWL notation, albeit with certain limitations depending on the baseline of the OWL standard used to specify the ontology in question.

This research significantly contributed to bridging the gap between human thinking and machine processing in developing and maintaining domain knowledge including domain requirements. The important role of ontology visualisation in order to address the complexity behind ontology repositories was explained. By providing graphical mind map based representations that provide accurate views on

specific relevant aspects of complex ontologies in a more organised and readable manner, it is possible to significantly facilitate communications between knowledge analysts/engineers and domain

experts/stakeholders during the development and maintenance of domain knowledge.

References

- [1] Al-Hroub, Y. (2010) Ontology-driven Requirements Analysis Tool (OntoRAT) for Ontology-driven Requirements Methodology (OntoREM). University of the West of England: MSc Dissertation.
- [2] Breitman, K., K. and do Prado Leite, J., C., S. (2003) Ontology as a Requirements Engineering Product. International Requirements Engineering Conference [Online]. IEEE. pp. 309 – 319. Available from <http://eolo.cps.unizar.es/docencia/MasterUPV/Articulos/Ontology%20as%20a%20RE%20product.pdf> [last access 30/05/2013]
- [3] Buzan, T. and Buzan, B. (2006) The Mind Map Book. BBC Active: London.
- [4] Castaneda, V., Ballejos, L., Caliusco, M., L. and Galli, M., R. (2010) The Use of Ontologies in Requirements Engineering [Online]. Global Journal of Researches in Engineering. Vol. 10 (6). pp. 2-8. Available from <http://engineeringresearch.org/index.php/GJRE/article/view/76/71> [last access 26/05/2013]
- [5] Dobson, G. and Sawyer, P. (2006) Revisiting Ontology-based Requirements Engineering in the age of the Semantic Web [Online]. UNSPECIFIED. Available from http://www.comp.lancs.ac.uk/~dobsong/papers/dre06_ontology_based_re.pdf [Last access 22/05/2013]
- [6] Gasevic, D., Djuric, D., Devedzic, V. (2006) Model Driven Architecture and Ontology Development. Springer: Berlin.
- [7] Gregory, D., Johnston, R., Pratt, G., Watts, M., J. and Whatmore, S. (2009) The Dictionary of Human Geography. 5th ed. Wiley-Blackwell: Chichester.
- [8] Gruber, T., R. (1993) A Translation Approach to Portable Ontology Specifications [Online]. Knowledge Acquisition. Vol. 5 (2). pp. 199-220. Available from <http://www.dbis.informatik.hu-berlin.de/dbisold/lehre/WS0203/SemWeb/lit/KSL-92-17.pdf> [Last access 25/05/2013]
- [9] Jaafar J. (2009) Collaborative Mind Map Tool to Facilitate Requirement Engineering [Online]. MSc Thesis, University of Manchester. Available from http://www.cs.manchester.ac.uk/resources/library/thesis_abstracts/BkgdReportsMSc09/JaafarJ.pdf [last access 30/05/2013]
- [10] Kossmann, M., Odeh, M., Gillies, A. and Wong, R. (2008) From Process-driven to Knowledge-driven Requirements Engineering Using Domain Ontology. IS'08 INCOSE.
- [11] Kotonya, G. and Sommerville, I. (1998) Requirements Engineering: Processes and Techniques. Wiley: Chichester.
- [12] MindManager (2014). MindManager software. URL: <http://www.mindjet.com/mindmanager/> [last accessed 26/08/2014]
- [13] Noy, N., F., McGuinness, D., L. (2001) Ontology Development 101: A Guide to Creating Your First Ontology [Online]. Development Journal. Vol. 32 (1). pp. 1 – 25. Available form <http://ebookbrowse.com/ontology-development-101-a-guide-to-creating-your-first-ontology-pdf-d296922486> [Last access 26/05/2013]

- [14] Petrie, C. (2007) No Science without Semantics [Online]. Internet Computing, IEEE. Vol. 11 (4). pp. 88-87. Available from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4270556&isnumber=4270537> [last access 27-05-2013]
- [15] Riechert, T., Lauenroth, K., Lehmann, J. and Auer, S. (2007) Towards Semantic based Requirements Engineering [Online]. University of Leipzig. Available from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.150.3395> [Last access 30/05/2013]
- [16] Siegemund, K., Thomas, E., J., Zhao, Y., Pan, J. and Assmann, U. (2011) Towards Ontology-driven Requirements Engineering [Online]. The University of Aberdeen: UK.
- Available from
https://files.ifi.uzh.ch/ddis/iswc_archive/iswc/ab/2011pre/iswc2011.semanticweb.org/fileadmin/iswc/Papers/Workshops/SWESE/4.pdf [Last access 30/05/2013]
- [17] Stanford Center for Biomedical Informatics Research (2014). Protégé software. URL: <http://protege.stanford.edu/> [last accessed 26/08/2014]
- [18] Swartout, W and Tate, A. (1999) Ontologies [Online]. Intelligent Systems and their Applications, IEEE. Vol. 14 (1). pp. 18-19. Available from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=747901&isnumber=16144> [last accessed 25/05/2013]