An evidence-based approach to collaborative ontology development

Emma Tonkin[1], **Heather D. Pfeiffer**[2] **and Andrew Hewson**[1]

Abstract. The development of ontologies for various purposes is now a relatively commonplace process. A number of different approaches towards this aim are evident; empirical methodologies, giving rise to data-driven procedures; self-reflective (innate) methodologies, resulting in artifacts that are based on intellectual understanding; collaborative approaches, which result in the development of an artifact representing a consensus viewpoint.

We compare and contrast these approaches through two parallel use cases, in work that is currently ongoing. The first explores a case study in creation of a knowledge base from raw, semi-structured information available on the Web. This makes use of text and data mining approaches from various sources of information, including semi-formally structured metadata, interpreted using methods drawn from statistical analysis, and data drawn from crowd-sourced resources such as Wikipedia.

The second explores ontology development in the area of physical computing, specifically, context-awareness in ubiquitous computing, and focuses on exploring the significant impact of an evidence-led approach. Both examples are chosen from domains in which automated extraction of information is a significant use case for the resulting ontology. In the first case, automated extraction takes the form of indexing for search and browse of the archived data. In the second, the predominant use cases relate to context-awareness.

Via these examples, we identify a core set of design principles for software platforms that bring together evidence from each of these processes, exploring participatory development of ontologies intended for use in domains in which empirical evidence and user judgment are allied.

1 INTRODUCTION

Ontologies, defined in the computer science area as "agreement about a shared, formal, explicit and partial account of a conceptualisation" (Spyns, 2002) are increasingly visible in various disciplines, particularly in the area of knowledge management, where the encoding of static domain knowledge is a key process (Aldea et al, 2003). Ontologies are generally applied for a number of purposes, including the following (Noy and McGuinness, 2001): to share common understanding of the structure of information among people or software agents; to enable reuse of domain knowledge; to make domain assumptions explicit; to separate domain knowledge from the operational knowledge; and to analyze domain knowledge.

Various methods for ontology generation have been identified: these could be described in general as introspection, or self-reflection (externalisation of expert knowledge), collaborative development through introspection and discussion (joint use of expert knowledge; see for example Della Valle et al, 2008), and data-driven or corpus-driven means (ie. unsupervised methods of ontology generation). Linking two or more of these methods together is also possible; for example, Carvalheira and Gomi (2007) describe a method that makes use of automated ontology generation to fuel a semi-automatic, or 'hybrid', overall process.

Ontologies are also becoming so large that it is hard for a single group to effectively develop and build a single ontology (T. Tudorache, et. Al, 2008); therefore, some collaborations have gone to using systems such as Protégé to keep track of the development process.

The debate surrounding the process of ontology building contains parallels with other forms of knowledge organisation and historical discussions on the topic of eliciting information about knowledge, language and the structures that underlie our everyday activities. In certain domains, the choice between introspective and data-driven approaches is one that defines the shape of the discipline. One example is the study of grammar in human languages; another is the perceived gulf between structured taxonomy/vocabulary for classification and the use of unstructured or very loosely-structured approaches, such as social tagging.

Here, we discuss an approach that links together a data-first approach with a collaborative discovery. Reports synthesised from expert knowledge have the advantage of very closely approaching the individual's own viewpoint; if then bolstered by discussion with others, the result may approach a consensus viewpoint. Such an approach does not take into account the visibility or availability of these features within the data that is available. Under some circumstances, this characteristic is not a defect for an ontology; however, if it is to be used for a data-driven or highly data-dependent application, such as for example a system that classifies documents within an ontology, or a context-aware wearable device using of a set of sensor signals to characterise and perhaps identify the current context in which the user stands, it is advantageous for the ontology to approach the dataset (to the extent that concepts and datasets may be expected to map).

[1] UKOLN, University of Bath. Email: {e.tonkin, a.hewson} @ukoln.ac.uk.

[2] Klipsch School of Electrical and Computer Engineering, NMSU. Email: hdp@cs.nmsu.edu.

2 A SOFTWARE-ENGINEERING APPROACH TO ONTOLOGY DEVELOPMENT

The practical importance in a collaborative context of an agile, open process was repeatedly emphasised during each case study, particularly in more interpretive developments. Agile development is supported by a number of new techniques, practices and tools have been developed; it tends to favour working solutions over future capabilities and encourages near-continuous engagement with users, non-specialist participants taking part in the development process, responding to changes in functional requirements as both the developer and the user increase their understanding of the problem space.

This approach emphasises test driven development (TDD), frequent testing of software during the development approach, rather than employing testing at the end of the main period of development - a late stage in the development process. Clear, measurable functional requirements are of importance, although the tests chosen may be revised flexibly at any point in the process. Outputs are tested frequently, during very short development iterations, which reduces the risk of 'drifting' away from core functional requirements.

Our work explores application of these principles to the development of structures designed for knowledge management (KM). These are generally aimed at a clearly defined problem-space. Functional requirements are derived from close association with users, domain information and evidence, in an ongoing iterative development process of agile development and review. We look towards a process of agile, test-driven ontology development.

3 IDENTIFYING SOFTWARE PLATFORMS FOR ONTOLOGY DEVELOPMENT

Various software have been evaluated, including CharGer (Delugach, 2001), Protégé and custom software developed for non-expert use. A number of informal procedures for collaborative ontology development were sketched out, with particular focus on actively working with existing patterns of collaboration within the domains under investigation. The use of a variety of tools implying very different levels of detail and technical accuracy or direct applicability implies a greater load on those responsible for completion of this progress.

However, certain tools offered greater accessibility for the non-expert. The corresponding benefit of this process was that, given an accessible, simplified surrogate, participants were able to quickly reach a level at which they were comfortable to discuss and contribute actively on issues such as the perceived quality, relevance or completeness of an ontology. The strawman surrogates under discussion, and the results of the discussion, are usually ambiguous, incomplete or contain invalid statements or assertions, so the raw results are indicative rather than directly applicable.

4 TESTS, METRICS AND EVALUATION

Various methods were identified and used for evaluation of ontologies. These can be compared to the general typology offered by Brank et al (2005):

- Comparison to a gold standard: an advantage of simplified representation is participants' ability to compare candidate structures to models originating within other knowledge management/ information retrieval domains.

- The use of the ontology within an application: test-driven approaches here were taken from the tradition of paper prototyping, including modifications of the card sorting methodology and use of a 'storyboarding' approach to facilitate discussion of the developed knowledge structure within a practical context of use of relevance to the participant.

- Comparison with a source of data: in each use case, data and text mining offered the likelihood of retrieving useful information. In the former case (textual resources), various automated methods were applied in order to retrieve information of a mostly statistical nature; this, it was felt, is primarily useful for refinements such as ranking options or identifying areas that are likely to contain weaknesses. In the ubiquitous computing domain, however, much the data available is more closely tied to a relatively static set of physical characteristics, notably geographic/positional information, and hence may be applied much more directly in the ontology development process.

- Evaluation by humans: how well an ontology meets predefined criteria. The definition of those criteria is a complex process in itself – indeed, the development process mentioned in this extended abstract is as concerned with exploring potential criteria for success as it is with development of structures designed to correspond with that vision. Brank et al. note that this process is particularly suitable to structural and syntactic levels, which corresponds well to our observations.

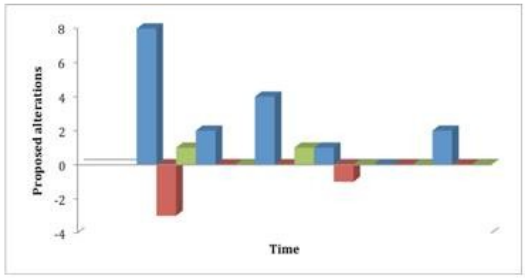

Figure 1. Data suggesting a trend towards stability with additions, pruning and structural changes. (Case Study 1)

During the case studies, a trend towards increasing structural stability has additionally been observed, particularly in the latter case. That is, the quantity, severity and nature of requested changes has been observed in general to diminish over time, suggesting that, as Viegas et al (2004) observe in their exploration of the stability of collaboratively-created user content, the first users to work on a given structure generally sets the tone of the result and, therefore, their work usually has the highest survival rate (see Figure 1).

5 CONCLUSION

The work described here essentially attempts to explore how a specialised and complex process may be rendered more accessible to a wider audience of potential users and contributors. Exploring the boundaries between ontology development and related areas of research may, as suggested by Brank et al., permit participants in a development process may integrate insights from related domains. We found that prototyping approaches using simple surrogates are useful in encouraging discussion and input, and that the relevance of data sources is dependent on the area, aspirations and contexts of use. The usability engineer is seldom considered a key participant in ontology development, but in cases in which ontology development becomes a collaborative (usually computer-supported) process, the participants become a key element in the success or failure of that development process.

REFERENCES

[1] Aldea, A., Bañares-alcántara, R., Bocio, J., Gramajo, J. and Isern, D. (2003) An Ontology-Based Knowledge Management Platform. In: Proceedings of the Workshop on Information Integration on the Web (IIWeb-03) at the 18 th International Joint Conference on Artificial Intelligence. Retrieved Jan 20th, 2009 from http://www.isi.edu/info-agents/workshops/ijcai03/papers/DIsern-article-ijcai.pdf

[2] Brank, J., Grobelnik, M., Mladenić, D. (2005). A Survey of Ontology Evaluation Techniques. Conference on Data Mining and Data Warehouses (SiKDD 2005), Ljubljana, Slovenia, 2005.

[3] Carvalheira, Luiz C. C. and Gomi, Edson Satoshi (2007) A Method for Semi-automatic Creation of Ontologies Based on Texts. Advances in Conceptual Modeling – Foundations and Applications. Springer.

[4] .Della Valle, Emanuele, Celino, Irene and Cerizza, Dario (2008). Agreeing While Disagreeing, a Best Practice for Business Ontology Development, In: Business Information Systems, Lecture Notes in Business Information Processing, Volume 7. ISSN 1865-1348 Springer Berlin Heidelberg.

[5] Delugach, Harry (2001). CharGer: A graphical Conceptual Graph editor. In *CGTools Workshop Proceedings in connection with ICCS 2001*, Stanford, CA, 2001. URL:http://www.cs.nmsu.edu/hdp/CGTOOLS/proceedings/index.html, Online Access: July 2001.

[6] Noy, Natalya F. and McGuinness, Deborah L (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[7] Spyns, Peter, Meersman, Robert and Jarrar, Mustafa (2002). Data modelling versus Ontology engineering Retrieved Jan 20th, 2009 from http://lsdis.cs.uga.edu/SemNSF/SIGMOD-Record-Dec02/Meersman.pdf.

[8] Tudorache, Tania, Noy, Natalya F., Tu, Samson, and Musen, Mark A. (2008) Supporting Collaborative Ontology Development in Protege. In *Proceedings of ISWC*.

[9] Viégas , Fernanda B., Wattenberg, M. and Dave, Kushal (2004). Studying cooperation and conflict between authors with history flow visualizations, Proceedings of the SIGCHI conference on Human factors in computing systems, p.575-582, April 24-29, 2004, Vienna, Austria.