The Knowledge Reengineering Bottleneck

Editor(s): Krzysztof Janowicz, Pennsylvania State University, USA and Pascal Hitzler, Wright State University, USA
Solicited review(s): Bernardo Cuenca-Grau, University of Oxford, UK and Sören Auer, Universität Leipzig, Germany

Rinke Hoekstra [a,b]
[a] *Department of Computer Science*
VU University Amsterdam
e-mail: hoekstra@few.vu.nl
[b] *Leibniz Center for Law*
Universiteit van Amsterdam
e-mail: hoekstra@uva.nl

Abstract. Knowledge engineering upholds a longstanding tradition that emphasises methodological issues associated with the acquisition and representation of knowledge in some (formal) language. This focus on methodology implies an ex ante approach: "think before you act". The rapid increase of linked data poses new challenges for knowledge engineering, and the Semantic Web project as a whole. Although the dream of unhindered "knowledge reuse" is a technical reality, it has come at the cost of control. Semantic web content can no longer be assumed to have been produced in a controlled task-independent environment. When reused, Semantic Web content needs to be remoulded, refiltered and recurated for a new task. Traditional ex ante methodologies do not provide any guidelines for this ex post knowledge reengineering; forcing developers to resort to ad hoc measures and manual labour: the knowledge reengineering bottleneck.

Keywords: knowledge engineering, ontology reuse, design patterns, linked data, dirty data, data reuse, provenance

1. Introduction

The field of knowledge engineering upholds a longstanding tradition that emphasises methodological issues associated with the acquisition and representation of knowledge in some (formal) language. Examples are the development of task-independent ontologies and the recent interest in design patterns. However, the focus on methodology implies an ex ante approach: "think before you act". And in fact, the same attitude is prevalent in traditional web-based publication of information. Information is moulded, filtered and curated in a way that befits the purpose of the information provider. In this position paper, I argue that the field of knowledge engineering is facing a new challenge in the linked data age as information providers become increasingly dependent on external data and schemas.

1.1. Ex Ante Knowledge Engineering

The ex ante approach of knowledge engineering originates in the problems identified in the development of large scale expert systems in the eighties and early nineties. Well known examples are Clancey's identification of types of knowledge in a knowledge base [4], the KADS and CommonKADS methodologies of [2,16] that separate a conceptual domain model from problem solving methods in the specifications of a knowledge based system, and Gruber's now famous characterisation of 'ontology' [10] and their physical reuse in the Ontolingua server [7] that culminated in the now commonplace use of the term to refer to a set of axioms that can be exchanged as a file. Ontologies soon became the center of attention for the field of knowledge acquisition – leaving problem solving methods largely ignored until only recently in e.g. [18]. It is the type of knowledge represented as an ontology – terminological knowledge – that was the main inspiration for the data model and semantics of the main Semantic Web languages.

The main focus was now directed towards the specification of design criteria and corresponding methodologies that ensured the development of ontologies suited for their main purpose: reuse in multiple systems [9]. For, it was thought, if ontologies are well-designed, they can be reused as task-independent knowledge components, enabling and facilitating more rapid construction of knowledge based systems by circumventing the knowledge acquisition bottleneck [8]. In the late nineties, and early 2000s, with the expected increase in the number of ontologies, a similar bootstrap seemed attainable by developing methods for reusing (parts of) ontologies in developing new ontologies, thus spawning research on ontology types, ontology merging, ontology alignment [15], ontology mapping, and – more recently – ontology modularisation.

In [12] I criticised the underlying assumptions of the alignment and merging of ontologies as these inevitably alter the ontological commitments of an ontology, rendering the claim of more reusable and compatible knowledge system components an empty one.[1] This criticism is moderated by the fact that many (if not most) ontologies are never used as a component of an expressive knowledge based system, but rather as facilitator for knowledge *management*; i.e. as 'semantic' annotations of information resources (documents, users). Knowledge management has indeed turned out to be the key use case for ontologies (and vocabularies) on the Semantic Web [6,12,18]. This is partly given by limitations of web-scale reasoning on expressive ontologies, although these limitations are of decreasing severity [17].

2. The Bottleneck

The methodologies and technical solutions we briefly discussed in the preceding section have been motivated and developed in a world *without actual data*: ontology engineering is an activity that takes place at design time. In a knowledge management setting, ontologies are often used for the annotation of fresh data. But the world has changed; the linked data cloud is growing at an exponential pace, and more and more applications become dependent on it. This has a significant effect on the way in which knowledge is being reused on the web.

Feigenbaum's knowledge acquisition bottleneck refers to the difficulty of correctly extracting expert knowledge into a knowledge base:

> "The problem of knowledge acquisition is the critical bottleneck problem in artificial intelligence." [8, p.93][2]

In contrast, the *knowledge reengineering bottleneck* refers to the general difficulty of the correct and continuous reuse of *preexisting* knowledge for a new task. The first difference between the two bottlenecks is that knowledge acquisition concerns the extraction of *generic* knowledge from a domain expert, while knowledge reengineering involves both generic and *assertional* knowledge. Indeed, knowledge engineering has contributed a lot to *enabling* schema level reuse, but traditional ex ante methodologies do not provide any guidelines for this ex post knowledge reengineering. Semantic web developers therefore resort to ad hoc measures and manual labour. The second difference is that on the linked data web, reuse is not a copy-and-paste operation, but rather a continuous relation of trust between a knowledge provider and its 'clients'.

Simply replace 'applied artificial intelligence' with 'the semantic web' in the following quote from Feigenbaum:

> "If applied artificial intelligence is to be important in the decades to come, we must have more automatic means for replacing what is currently a very tedious, time-consuming and expensive procedure." [8, p.93]

The tedious procedure alluded to by Feigenbaum is the procedure by which we integrate (existing) knowledge into a new system. The web of data may be more accessible than expert knowledge in a human brain, it is often expressed in a very convoluted manner, making it hard to reuse [11].

3. Challenges

The rapid increase of both quantity and importance of linked data poses new challenges for knowledge engineering and the Semantic Web project as a whole:

Challenge 1: Data Dependency Knowledge engineering is not yet fully accustomed to the ubiquity of instance data. An example is current work on ontology and vocabulary alignment. The Ontology Alignment Evaluation Initiative (OAEI) annually specifies

[1] In fact, this extends to the reusability of ontologies and ontology design patterns.

[2] The knowledge acquisition bottleneck is often misunderstood as the high threshold in effort before knowledge representation starts to pays off, and practical reasoning problems can be solved.

a set of ontologies for benchmarking alignment systems. These systems are evaluated against a reference alignment, or checked for coherence, but not against a set of instance data.[3] At the moment, this does not seem to be a very pressing issue. The most prominent use case for ontology alignment is information retrieval, and formal characteristics of the aligned ontologies and datasets play only a limited role. In a retrieval setting, alignment quality can be assessed by comparing precision and recall with or without using the alignments. A limited loss of retrieval quality can be outweighed by the added advantage of search using two vocabularies. In a more knowledge intensive setting, however, loss of quality has a more significant effect: instance data can be classified under the wrong type. How current ontology alignment techniques will scale to use cases for tasks that require higher expressiveness is at the present time still an open question.

Challenge 2: Limited Control Although the dream of unhindered knowledge reuse is a technical reality, it has come at the cost of control. Similar to the Web 2.0 revolution, where information consumers transformed into information producers; semantic web content can no longer be assumed to have been produced in a controlled environment. First of all, this means that data is 'dirty'; it may not be the latest version, it may be inconsistent, it may use multiple identifiers for the same resource, it may have gaps in coverage, or be redundant. The prototypical example of the dangers of this type of issues is the excessive use of owl:sameAs assertions between resources in different data sources. Furthermore, there is no guarantee that the ontologies that define the classes and properties used in the data are used in the specified way: the relata of a property may not be of the correct type, the data may be expressed in terms of an older version of a schema, or the data may cause the schema to become inconsistent.

Recently, the SIOC Project has made a change to its schema – an increasingly popular vocabulary for expressing social networking knowledge.[4] sioc:User was changed to sioc:UserAccount to avoid conflation of the class with foaf:Person. The change was announced on the SIOC website, and the schema owner advised users to change their data accordingly. Arguably, a change to such a widely used schema can have enormous consequences, certainly as we cannot assume that all occurrences of sioc:User will be replaced, nor that tool developers will provide the necessary update. But, what *are* these consequences, and how do we prevent or amend them?

The pragmatic, ad hoc approach to dirty data is to "just fix it". An example is the recently started "Pedantic Web" group; a group of concerned experts that functions as a communications channel between data owners, schema owners, and users, allowing them to file bug reports, and suggest fixes.[5] Indeed, repairing dirty data and schemas is a noble effort, but it is doubtful whether this initiative can scale and remain effective over the coming years.

In the end, data and schema quality have to be assured in some automatic way. Description logics reasoners will tell you whether a knowledge base is consistent, but there is a tradeoff in optimisation between expressive TBox reasoning, or reasoning on a large ABox (see e.g. [5]). Approaches that allow reasoning on very large amounts of (dirty) data, such as [17], are based on forward chaining algorithms that do not detect inconsistencies or other problems. An additional issue is that the results of tableaux algorithms are very hard to explain [13] and problems can only be fixed one at a time. Techniques for reasoning with inconsistent ontologies, such as e.g. [14], show promising results but their value depends on task context. Knowledge engineering can certainly play a role in investigating reasoning strategies for tasks on the web of data.

Another question is, what should a knowledge reuser do when encountering a problem? If it is not your own data, who should fix it? The model chosen by the BBC music website is to fix the original information source (e.g. Musicbrainz).[6] Clearly this model only works when dealing with community-developed *open* data; in a more restricted setting, other models will be more suited (including not fixing it). Different users may adopt conflicting models for the same data: a knowledge provider has to make clear how its data and/or schema should be used, what its versioning regime is, and has to provide provenance information for quality assurance.[7]

[3] See for evaluation methodology the OAEI and Ontology Matching workshop pages at http://oaei.ontologymatching.org/.

[4] SIOC: Semantically-Interlinked Online Communities. See http://sioc-project.org.

[5] "We want you to fix your data", see http://pedantic-web.org/

[6] See http://www.bbc.co.uk/music and http://www.musicbrainz.org, respectively.

[7] See other contributions in this volume, and the W3C Incubator Group on Provenance, http://www.w3.org/2005/Incubator/prov/.

Challenge 3: Increased Complexity The issues raised by the two preceding challenges are not new to many of us working with Linked Data. However, in context of the decennia-old debate between *neats* and *scruffies*,[8] these challenges are currently addressed only through the pragmatics of the latter perspective. Most of the experience gained there precipitates in blog posts, or best practices documents, rather than traditional scientific discourse.[9]

With scruffy linked data on the rise, it is likely that new Semantic Web applications [6] will capitalise on this data and move beyond the simple lookup and mashup services listed by [18]. These applications may not all live on the web or produce linked open data, but they will depend on it and require more expressiveness. As a consequence, the complexity and task-dependence of content on the web of data will increase, emphasising the need for a knowledge reengineering perspective. What does task-dependence of data mean on the web? Is there a role for knowledge engineering insights from the nineties, such as the problem solving methods of CommonKADS [3]? Understanding patterns in data reuse (as opposed to ontology design pattens) is currently uncharted territory.

Challenge 4: Increased Importance As the scale of the web of data increases, the number of applications that depend on it will increase as well. One of the major successes of the linked data initiative is the take-up by non-academic parties, such as the BBC, the UK and US governments, and more recently Google and Facebook. These parties are new stakeholders on the web of data, and it is not likely that this take-up is going to stop anytime soon. At the moment it is unclear how these non-academic parties will behave in the future, but linked data has already left the toy worlds of AI researchers and is increasingly mission critical to stakeholders. Facing the challenges iterated above becomes more important as coverage grows in influential domains such as commerce and legal and government information.

4. Discussion

In this short paper I call for a new role for knowledge engineering that takes the ubiquity of instance data into account. The challenges discussed in section 3 are not new, but have to be faced in order to make the Semantic Web – and not just a web of data – a success. Indeed, that these challenges arise is a sign of a maturing domain. The dependency on data means that the web of data has become an object of study in its own right. It has grown beyond the control of the (academic) community that gave rise to it – similar to the Web itself [1].

Insights from knowledge engineering have played an important role in the initial design of Semantic Web technology, but the field seems to be sticking to its own turf rather than reaching out to help overcome the reengineering bottleneck.

Acknowledgements

I would like to extend my thanks to the reviewers for their comments on the first version of this paper.

References

[1] Tim Berners-Lee, Wendy Hall, James Hendler, Nigel Shadbolt, and Daniel J. Weitzner. Creating a science of the web. *Science*, 313, August 2006.

[2] J.A. Breuker and B.J. Wielinga. Knowledge acquisition as modelling of expertise: the KADS-methodology. In T. Addis, J. Boose, and B. Gaines, editors, *Proceedings of the European Knowledge Acquisition Workshop*, pages 102 – 110, Reading GB, 1987. Reading Press.

[3] Joost Breuker and Walter Van De Velde, editors. *CommonKADS Library for Expertise Modeling: reusable problem solving components*. IOS-Press/Ohmsha, Amsterdam/Tokyo, 1994.

[4] William J. Clancey. The epistemology of a rule-based expert system - a framework for explanation. *Artificial Intelligence*, 20(3):215–251, 1983. First published as Stanford Technical Report, November 1981.

[5] Bernardo Cuenca Grau, Boris Motik, Zhe Wu, Achille Fokoue, and Carsten Lutz. OWL 2 Web Ontology Language: Profiles. Technical report, W3C, 2009.

[6] M. d'Aquin, E. Motta, M. Sabou, S. Angeletou, L. Gridinoc, V. Lopez, and D. Guidi. Towards a new generation of semantic web applications. *IEEE Intelligent Systems*, 24:20–28, 2008.

[7] Adam Farquhar, Richard Fikes, and James Rice. The ontolingua server: a tool for collaborative ontology construction. *International Journal of Human-Computer Studies*, 46(6):707–727, 1997.

[8] See [12, Ch.2] and `http://en.wikipedia.org/w/index.php?title=Neats_vs._scruffies&oldid=323249466` for an overview.

[9] An example is Jeni Tennison's blog on her experiences with translating UK government data to RDF, `http://www.jenitennison.com/blog/`.

[8] Edward A. Feigenbaum. Knowledge engineering: the applied side of artificial intelligence. *Annals of the New York Academy of Sciences*, 426:91–107, 1984. Original publication in 1980 as report of the Stanford department of Computer Science.

[9] Mariano Fernández-López and Asunción Gómez-Pérez. Overview and analysis of methodologies for building ontologies. *The Knowledge Engineering Review*, 17(2):129–156, 2002.

[10] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, 1994. Kluwer Academic Publishers.

[11] Pascal Hitzler and Frank van Harmelen. A reasonable semantic web. *Semantic Web*, 2010. this issue.

[12] Rinke Hoekstra. *Ontology Representation – Design Patterns and Ontologies that Make Sense*, volume 197 of *Frontiers of Artificial Intelligence and Applications*. IOS Press, Amsterdam, June 2009.

[13] Matthew Horridge, Bijan Parsia, and Ulrike Sattler. Lemmas for justifications in OWL. In Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, and Ulrike Sattler, editors, *Description Logics*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[14] Zhisheng Huang, Frank van Harmelen, and Annette ten Teije. Reasoning with inconsistent ontologies. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 454–459, Edinburgh, Scotland, Aug 2005.

[15] Michel Klein. Combining and relating ontologies: An analysis of problems and solutions. In *Proceedings of the Workshop on Ontologies and Information Sharing (at IJCAI 2001)*, pages 53–62, Seattle, WA, 2001.

[16] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van den Velde, and B. Wielinga. *Knowledge Engineering and Managament: The CommonKADS Methodology*. MIT Press, 2000.

[17] Jacopo Urbani, Spyros Kotoulas, Jason Maassen, Frank van Harmelen, and Henri Bal. OWL reasoning with WebPIE: calculating the closure of 100 billion triples. In *Proceedings of the Seventh European Semantic Web Conference*, LNCS. Springer, 2010.

[18] Frank van Harmelen, Annette ten Teije, and Holger Wache. Knowledge engineering rediscovered: Towards reasoning patterns for the semantic web. In N. Noy, editor, *The Fifth International Conference on Knowledge Capture*, pages 81–88. ACM, 2009.