Reflections on 25+ years of Knowledge Acquisition

Bob J. Wielinga
Vrije Universiteit Amsterdam
b.j.wielinga@uva.nl

23 September 2012

Abstract

In this paper I give a short reflection on Knowledge Acquisition as a subfield of AI and Knowledge Engineering over the last 25 years or so. My major message is that knowledge modeling is an underrated but still important method to reduce the complexity problems that arise in constructing knowledge-based applications. Scale -as apparent in the Semantic Web- is an other important parameter in recent developments in Knowledge Acquisition: it requires other techniques than those of the 1980’s. Natural Language Processing is the most promising way forward, but also the most difficult source of the acquisition of formalized knowledge. I will argue that some of the lessons learned in building knowledge-based systems may carry over to reasoning in the Semantic Web and to knowledge acquisition from natural language web sources.

1 Lessons from the Past

Mankind has been concerned with knowledge and knowledge processes for many thousands of years. Brian Gaines discusses in this issue evidence of knowledge capture as early as 30,000 years ago [Gaines2012]. Philosophers have been concerned with the nature of knowledge over the last two and a half millennia. However it is only a relatively short time ago that questions arose how to represent knowledge in such a way that it can be used by computer programs and - more importantly in the context of this special issue- how to acquire such formalized knowledge. Knowledge representation has been a topic in AI ever since the field emerged in the 1950’s, but knowledge acquisition only became an issue in the early 1980’s with the advent of computer systems that could use considerable amounts of knowledge to solve problems: expert systems.

My personal interest in knowledge acquisition for computer systems started in 1983 with one of the first ICT (Esprit) projects of the European Community: P12 [Hayward et al.1987]. The goal of that project was to develop engineering principles for the acquisition and implementation of expert knowledge. Using techniques from psychology -interviews, repertory grids, modeling- and AI we attacked the knowledge acquisition problem ([Wielinga and Breuker1984]). Early lessons learned were that verbal data -as acquired by various interviewing techniques- do not speak for themselves, they need interpretation to become usable in computer sys-
tems. Such interpretation requires modeling [Wielinga and Breuker1986]. These early exercises in knowledge acquisition spawned a series of European projects aimed at developing a structured methodology for engineering knowledge-based systems resulting in the CommonKads methodology [Schreiber et al.1999]. The starting point of CommonKads was the notion of the knowledge level [Newell1982]. Modeling at the knowledge level enabled a decoupling of the implementation details and the actual content of the knowledge. CommonKads advocated the following principles for knowledge modeling:

- Knowledge Acquisition requires interpretation and modeling at the knowledge level ([Newell1982]);
- Knowledge comes in 3 types: domain knowledge (domain schemata and instances thereof), inference knowledge and task (control) knowledge;
- Knowledge templates are a crucial tool for knowledge acquisition and knowledge reuse;
- In particular Problem Solving Methods (PSM’s) [Benjamins and Fensel1998] are template models for generic reasoning processes coupled to domain knowledge schemata.

The metaphysical stance of CommonKads was based on pragmatic considerations. Domain knowledge is clearly the field of the domain expert and can be elicited by fairly simple techniques. Domain schemata and inference knowledge are the domain of the knowledge engineer who interprets reasoning steps of the human expert in terms of cognitive processes, the task level specifies a control strategy which closely relates to AI strategies for search and problem solving. The distinctions made in CommonKads and comparable approaches [Clancey1985, Steels1990] were useful in building knowledge-based systems and libraries of reusable components successfully supported the knowledge engineering process [Valente et al.1998, Breuker1999].

What happened to the ideas that CommonKads advocated? John Fox, in his reflection on 25 years of the “Knowledge Engineering Review” [Fox2011], wondered whether the idea of reusable components, such as PSM’s, as building blocks for knowledge-based systems that was so prominent in the 1990’s has failed or that it has just gone out of fashion, and if so why? There is no simple answer to this question. The world has changed with the Semantic Web [Gil2011, d’Aquin et al.2008]. With the Semantic Web the domain of KA has become much larger, but the tasks have become simpler. Knowledge engineering projects that are just concerned with single user, stand alone knowledge based systems in a closed world are not a major concern of the KA community anymore, albeit that such systems are still being developed in commercial contexts (see for example [Zhou et al.2007, Yang et al.2006]). The tasks in the expert systems era, such a diagnosis, assessment, planning, configuration and design, were complex but well defined, while the tasks in the Semantic Web context are much more difficult to classify. We are now concerned with tasks like semantic search, information extraction, the representation of large knowledge bases and knowledge integration. However, the powerful notions of
reusable knowledge structures as defined in CommonKads have not disappeared, but come back in a somewhat different form, as we will discuss in subsequent sections.

The special issue of AI EDAM [Brown2009] shows other evidence that the notion of PSM is still relevant in Knowledge Engineering. In particular [O’Connor et al.2009] and [Domingue and Fensel2009] argue in that issue that the PSM approach is still a promising line of research also in the context of the web, but that there is a lack of tools to support the configuration of PSMs into practical applications. This lack of tools to support knowledge level modeling and PSM configuration is remarkable. In the 1990’s much interest and effort was dedicated to the development of tools for KA and many new tools were developed and presented for example during the Banff KAW conferences. Only one of all these tools appears to have survived: Protege [Musen1989].

In the early days of KADS the knowledge model contained a fourth layer: the strategic layer. The knowledge involved in this layer was subject of an extensive research project (REFLECT, [Reinders et al.1991]), but was removed from the CommonKads knowledge model since it concerned knowledge about reasoning processes and not about the domain or task itself. Meta-level reasoning in the context of knowledge-based systems has not been a topic of major interest in the KA community. However, issues like how to select or configure a problem solving strategy or how to deal with inconsistent results of a reasoning process seem still to be valid in current research [O’Connor et al.2009]. The IBROW project [Benjamins et al.1998] aimed at the automatic configuration of problem solving methods from PSM components distributed on the web. A specification language for defining problem solving competence, UPML [Fensel et al.1999], was developed and several implementations of a brokering service for problem solving were created [Benjamins et al.1999, Crubézy et al.2003]. Work on semantic web services could potentially benefit from these techniques for integrating and combining distributed software components. Unfortunately the results of the REFLECT and IBROW projects have not been taken up, possibly because the issues tackled in these projects were somewhat ahead of their time.

In the 1990’s the focus on small scale, complex knowledge bases shifted somewhat to larger knowledge repositories which had a less complex structure, but were better understood and carried the promise of reusability: ontologies. In philosophy an ontology is a theory of being, but in Computer Science an ontology is a sharable conceptualization of some domain [Gruber1993]. Van Heijst et al [van Heijst et al.1997] add to that definition that ontologies should be at the knowledge level and that the viewpoint taken in the conceptualization may be affected by the particular domain and the task that the ontology should support. As a consequence a distinction between different types of ontologies was introduced: generic ontologies, domain ontologies and application ontologies.

In the 1990’s it was gradually realized that linguistic databases [Fellbaum1998] and thesauri such as the Art and Architecture Thesaurus [The Getty Foundation2000], could become valuable resources for the Knowledge Acquisition community [Wielinga et al.2001]. At the same time it became clear that such sources were not perfect: the notion of concept was somewhat overloaded, relations were not used consistently and coverage was imbalanced. Dealing with these problems became a major research issue.
[Gangemi et al.2003]. In spite of these problems such resources are now part of the LOD cloud and are being used in many applications.

In conclusion, 25 years of Knowledge Acquisition research has highlighted some useful concepts: the importance of the knowledge level, the typology of knowledge types, reusability of knowledge at different levels of abstraction and the advantages of libraries of knowledge components.

2 Problems of the Present

Although CommonKads has been widely adopted in the Knowledge Engineering community (see for example [Prat et al.2012, Yang et al.2006, Zhou et al.2007]), in Semantic Web research the use of modeling for reasoning has been almost non existent, with a few exceptions [Arguello et al.2008, van Aart et al.2010]. The Semantic Web community has largely ignored the lessons learned in engineering knowledge based systems. Van Harmelen et. al. [van Harmelen et al.2009] have noted this problem and present an extensive analysis of a number of Semantic Web applications. Although they use a set of primitive inferences which is more logic based than the usual set of inferences used in CommonKads, their work clearly shows an added value of modeling Semantic Web applications at the knowledge level using a structured approach. They conclude that many Semantic Web applications can be generalized to a small set of generic tasks that involve only limited forms of reasoning. The question "where is the reasoning in the Semantic Web and what forms does it take?" is still a very valid one.

The importance of modeling becomes even more evident when one considers the increasing complexity of standard representation schemata arising in the Semantic Web context. The relations in RDF(S) are limited and the inferences that can be made over these relations, such as inheritance, are fairly well-understood. The introduction of more complex schemata such as SKOS, introduces relations with a much more fuzzy semantics and hence much less clear inferences that can be made with them. What can we derive from the statement that A skos:hasBroader B? One obvious inference is that B is a more general concept than A, hence the set of instances of A may overlap with those of B. More subtle inferences could be that certain properties of B carry over to A, but maybe not all of them. Alternatively, narrower concepts of B can have some similarity to A. Making inferences over relations that have a more intuitive than a pure logic interpretation explicit was one of the goals of CommonKads. Complex schemata, such as the Europeana Data Model [Hennicke et al.2011] for integrating large amounts of heterogeneous data sets, introduce an even larger set of relations that require complex interpretation processes to be used in intelligent processes. Modeling large and complex datasets at the schema level is well underway in the Semantic Web community, but modeling the reasoning processes over these schemata is still very much an open problem.

Knowledge Acquisition in the last two decades of the previous century was largely a small scale enterprise. Knowledge bases consisted of a few thousands concepts at most. Current efforts in knowledge research aim at capturing much larger sets of knowledge
elements and attempt to build applications that employ very large knowledge bases. Elicitation techniques developed in the 1980’s are just not efficient enough to cater for the creation of such large scale knowledge repositories. Where will the knowledge of even larger knowledge bases come from? The Linked Open Data initiative [Bizer et al.2009] shows that large amounts of knowledge can be made accessible. The problem however is how to link the millions of knowledge elements of individual sources to each other and how to integrate all these pieces of knowledge into a coherent whole. A first step in this integration process is that of mapping ontologies. The work of Tordai [Tordai2012, Tordai et al.2011, Tordai et al.2010] shows that mapping large scale ontologies such as Wordnet [Fellbaum1998] and the Art and Architecture Thesaurus [The Getty Foundation2000] can be done automatically to a certain extent, but many problems remain:

- coverage even between fairly general ontologies differs considerably;
- ambiguity in concept labels causes major problems in ontology mapping;
- hierarchical structures modeling similar domains are often very different, because of different viewpoints on the world that ontology builders adhere to;
- concepts are often fuzzy rather than Aristotelian, well defined entities.

The latter point has been well documented in psycholinguistic studies (for example [Rosch et al.1976, Lakoff1987]). Lakoff argues that concepts as cognitive constructs in human thinking and language should be seen as conceptual clouds rather than as single points of meaning. For example the concept “mouth” has a central (cf. Rosch prototypes) meaning of an entrance through which some substance can be transported, but it has also many derived -instrumental, metaphorical, metonymic- meanings such as the mouth of a river or a mouth as a communication device. This complexity of concepts makes mapping and integration of ontologies a difficult task. Complex relational networks and even more complex reasoning processes will be needed to produce behavior that really mimics human capabilities.

Whereas the sources of knowledge in the expert systems era were human experts, for large scale Knowledge Acquisition we now have to resort to other source materials, notably texts representing knowledge in natural language. Extracting knowledge from text remains a very difficult task. Parsers that produce parse trees for sentences are widely available, but the interpretation of such tree structures into knowledge is still a difficult area. Certain types of structures such as Named Entities, dates and simple events can be extracted from parsed texts, but more complex concepts and relations are still very difficult to detect.

3 Future Directions

Assigning meaning to paths in RDF graphs can be achieved by using semantic patterns that group sequences of relations into abstract relations, much in the same way as
inferences in CommonKads generate new knowledge by combining simple domain facts. Semantic relation search will greatly improve the usefulness of the large amount of simple relations currently present in the Semantic Web.

In recent work on ontology engineering a similar notion, that of knowledge patterns is emerging [Clark et al.2000, Schreiber and Aroyo2008, Gangemi2005, Gangemi and Presutti2010]. Knowledge patterns are units of meaning that can be used as building blocks of ontologies and domain knowledge bases. The idea of knowledge patterns is similar to domain schemata in CommonKads, although the latter patterns are closely connected to PSM’s and the importance of patterns as stand alone meta-level structures was not prominent in CommonKads. Building libraries of reusable knowledge and inference patterns appears to be a promising way forward to make sense of the large amount of implicit knowledge in the Semantic Web.

The potential of the huge amounts of knowledge in the form of natural language documents is even greater, but more difficult to explicate. Language technology -lexical databases, parsers, information extraction techniques- has reached a point where it is usable on a large scale. However, interpreting the structures that such technology produces is still very difficult. Early attempts to guide natural language interpretation using template models such as scripts [Schank and Abelson1977] were successful in small domains, but did not scale up. Some research projects already explore similar lines. For example, the DART database [Clark and Harrison2009] contains a large amount of simple propositions of the type “an airplane can fly”. It would be very interesting to see whether generalizations of these propositions can lead to interesting relational patterns. Framenet [Baker and Sato2003] and its translation into knowledge patterns [Nuzzolese et al.2011] may be another source of interesting, reusable knowledge structures.

A main challenge for today’s research in KA is to develop semantic patterns which can fulfill a similar role as generic models in problem solving.

4 Conclusions

Allen Newell’s notion of the Knowledge Level has been widely acknowledged in the AI community, but at the same time its implications appear to have been largely forgotten. CommonKads has attempted to provide a framework to describe knowledge structures and corresponding reasoning processes in semi-formal models at the knowledge level. Such models have been successful in small scale knowledge-based system applications, but have scarcely been applied in large scale Semantic Web applications. Developing and testing such models and abstracting them into templates (cf. inference patterns and PSM’s) will -in my view- be a crucial advance to the development of Semantic Web technologies.

Scaling up Knowledge Acquisition to sizes that are promised by the Semantic Web, will be a major challenge both to researchers in the Knowledge Sciences and to those in the Natural Language Processing communities. Attacking these challenges could benefit from taking some of the lessons of Knowledge Acquisition research from the previous century more seriously than has been done so far.
References


