Users are always right…even when they are wrong: Making Knowledge Representation Useful and Usable

Alan Rector

School of Computer Science, University of Manchester, Manchester M13 9PL, UK
rector@cs.man.ac.uk

Abstract

Users' problems in developing applications rarely factor easily so as match the solutions provided by the Knowledge Representation community. If KR is to be applied successfully, the gap between users' "problem space" and the KR "solution space" must be addressed. Two approaches are suggested: "Intermediate representations" at a higher level of abstraction than current KR languages and a set of application oriented questions to expose how different systems address common user questions. The results are likely to be hybrid systems including heuristics and 'non-standard' reasoning.

The *Problem Space* and the *Solution Space*

What counts as successful knowledge representation? At first glance the answer seems simple: effective use in knowledge based applications that could not, or would not, have been built or maintained otherwise. Applications are built to solve problems – and ultimately problems 'belong' to users.

At one level, we appear to be in a golden age of Knowledge Representation with the explosion of the Semantic Web, and widespread recognition of RDF, OWL and related standards. On another level, there remains the same gap between the practitioners and the users pointed out by Doyle and Patil in the early 1990s (Doyle and Patil 1991) – the formalisms that are tractable and well understood do not match users' needs. Yet most of the Knowledge Representation community tends to be focused on technical arguments about different formalisms or solutions – DLs, OWL, Rules, CLP, Belief Nets, Machine Learning, etc. – rather than users' problems.

Users are the experts in their problems – even when they present them as (possibly naïve) solutions. Users' need an improved supply chain manager, an adaptable clinical data capture tool, a repository for biological data, or whatever. Users define the "problem space". More often than not will find a way to achieve something that approximates their ends, however ill founded the technologists think it in terms of the "solution space". Not infrequently users want "an answer" rather than "the best answer" – an improvement to the status quo rather than perfection. Furthermore, the limitations users accept may not reflect technologists' values. For example, a user may value predictability or expressiveness over completeness or even decidability. There is an analogy with the well-known quote from John Tukey (Tukey 1962). For a user it may be:

> …far better to have an approximate answer to the right question … than the exact answer to the wrong question…

None of this is original. It is the stuff of user-centred design textbooks and a key part of knowledge representation paradigms such as CommonKADS (Schreiber 2000). The issues date back to the arguments between the 'neats' and 'scruffies' of early AI (Sloman 1990) now extended to philosophical ontology domain (Goble and Wroe 2004).

The Challenge of the *Problem Space*

Reformulating users' problems in terms of available solutions is at the core of many other disciplines, including the author's field of Health- and Bio-Informatics. Too often the KR community appears to put the burden on users to reformulate the problem in terms of the KR community's criteria for judging solutions.

Add to this is the fact that logic is daunting to many users, even users sophisticated in other forms of mathematics. Add further to this that KR language features are more often chosen for computational tractability and logical elegance than usability or continuity. Add still further serious discontinuities – the fact that newer formalisms do not always answer questions that were the *raison d'être* for their predecessors. Then it is wonder that there is confusion and skepticism as we try to build standards around more sophisticated KR languages such as OWL.

How can the KR community address these problems? Clearly not by abandoning its high standards for precision in formulating the solution space. However, that focus needs to be supplemented by renewed attention to the problem space both theoretically and empirically.

A user- and application-oriented vocabulary is needed for user issues. The Semantic Web Best Practice and

Deployment Working Group (SWBP[1]) has made significant inroads in the OWL community with its patterns. Likewise, the OWL community has made efforts to respond to user requirements **in** drafts for "OWL 1.1". However, both responses are still too close to the languages themselves. What is needed is a higher level of abstraction that is a better match for user problems and a programme of empirical testing of both heuristic techniques to supplement formally provable solutions as to whether they actually solve user problems.

We suggest the KR community work with users to:

- *Design higher level languages* – to capture users' intentions, even if they cannot be fully implemented. "Intermediate representations" were once commonplace and can make a dramatic impact on usability (Rector et al, 1991). Users' knowledge is too rare and expensive to be restricted by our current paradigms.
- *Formulate a series of higher level questions* – and describe KR systems in terms of how they answer them, or what 'non-standard', heuristic, or approximate mechanisms might be used to answer them.
- *Gather empirical evidence* – developers' intuitions about users' needs are notoriously fallible. For example, controversies such as whether open or closed world reasoning is more appropriate for the Semantic Web (De Bruijn, et al. 2005) (Schneider and Horrocks 2006) require much firmer grounding in empirical data.

Some higher level questions

The following is intended merely as starting point derived from our experience. Others will want to add additional question from their own.

Firstly, questions about the information represented:

- *What is known about an entity? W*hat does this representation say about this entity? All entities of this type? Some entities of this type? Since a complete answer is impossible given rich inference, what sort of answers match users' needs in various situations?
- *What can sensibly be said about an entity?* A surprisingly difficult question to pose in the dominant DL paradigm, although central to frame-based systems and crucial to any application involving data capture.
- *What can sensibly be asked about an entity?* In general, the less is known about something, the more it is sensible to ask and the less it is sensible to say. For example, asking about "Green entities" is sensible. Making a statement that something is green before we know it to be a physical object is not. This intuition is poorly catered for by most systems.
- *How do I refer to this representation of this entity?* The most basic requirement of the Semantic Web, but unclear in OWL.[2]

Secondly, meta-questions about the formalisms and how to use them:

- *Why is this entity represented here? Who authored the representation? When? Why? on what authority?* Metadata is as important to knowledge representation as knowledge representation is to meta-data.
- *What methodology/formalism to use and when?* Potential users are met with a bewildering array of choices: First order Logic? OWL? RDF(S)? Frames? Databases? Rules? Logic programming? DL programming? When should which be used? Not be used? What is the minimum needed for a specific job.
- *How does this formalism relate to UML and other object oriented formalisms with which most users are familiar?* How can typical UML models, be represented in the formalism? What are the differences? What is gained or lost. If they cannot be, why not?
- *How does the formalism deal with defaults and exceptions? Uncertainty*? If it does not, is there any experience in heuristics or work-arounds?

Such an approach is likely to lead to hybrid systems including 'non-standard' reasoning and heuristics. If so, so be it – if the resulting systems answer the right questions in ways that users find useful, usable, understandable and correct to their criteria.

References

De Bruijn, J.; Lara, R.; Axel, P. and Dieter, F. (2005). OWL DL vs. OWL flight: conceptual modeling and reasoning for the semantic Web. In: *Proceedings WWW-2005.* Chiba, Japan, ACM Press: 623-632.

Doyle, J. and Patil, R. (1991). "Two theses of knowledge representation: Language restrictions, taxonomic classification and the utility of representation services." *Artificial Intelligence* 48: 261-297.

Goble, C. and Wroe, C. (2004). "The Montagues and the Capulets." *Comparative and Functional Genomics* 5(8): 623-632.

Patel-Schneider, P. and Horrocks, I. (2006). A comparison of two modelling paradigms in the Semantic Web. In Proceedings of WWW-2006, Glasgow (in press).

Rector, A. L.; Zanstra; P. E. Solomon, W.D et al. (1999). "Reconciling users' needs and formal requirements: Issues in developing a re-usable ontology for medicine." *IEEE Trans Info Technology in BioMed* 2(4): 229-242.

Schreiber, G. (2000). Knowledge engineering and management: The CommonKADS methodology. Cambridge, Mass, MIT Press.

Sloman, A. (1990). Evolving knowledge in natural science and artificial intelligence. In: Tiles, J.E and Deen, G.C. (eds) *Evolving knowledge in natural science and artificial intelligence,* London, Pitman.

Tukey, J. (1962). "The future of data analysis." *Annals of Mathematical Statistics* 33: 1-67.

[1] http://www.w3.org/TR/swbp/

[2] See http://www.w3.org/TR/swbp-classes-as-values/