Situated Cognition and Knowledge Acquisition Research

Paul Compton
School of Computer Science and Engineering, The University of New South Wales
Sydney 2052, Australia. compton@cse.unsw.edu.au

Keywords: Situated Cognition, Ripple-Down Rules, Knowledge Acquisition, Knowledge Engineering

Introduction

The difficulties in building expert systems led to a philosophical and cognitive science analysis of why it was so difficult obtaining knowledge from experts, largely from a situated cognition perspective (Winograd and Flores 1987; Clancey 1997). Situated cognition offers a broad-ranging perspective, applying particularly to education, but the key idea in relation to expert systems, can probably be summarised as simply that experts never describe how they reach a conclusion, rather the knowledge they express is a justification for their conclusion created for the particular context. In the late 80s and early 90s there was intense discussion about situated cognition at the knowledge acquisition workshops but this largely philosophical discussion soon faded. As Menzies comments:

> It could also be argued that a philosophical perspective on human reasoning has little relevance for tool builders such as pragmatic knowledge engineers. (Menzies 1998)

The consequent history of the knowledge acquisition workshops and conferences has been largely and appropriately a history of trying to develop engineering approaches. The intention of this paper is to briefly consider some of the major strands of this research and the impact or lack of impact of a situated cognition perspective on this research.

Situated Cognition

I was involved in the development of one of the first medical expert systems to be in routine clinical use and had the task of maintaining it – and in four years the knowledge base doubled in size while the accuracy went from 96% to 99.7% (Compton, Horn et al. 1989). Encouraged by Mildred Shaw whom I met at the 1988 5th Generation Conference in Tokyo and Bill Clancey's keynote on situated cognition at the Australian AI conference that year, I submitted a paper to the 1989 EKAW which was an attempt at a philosophical analysis of the difficulties of maintaining an expert system, and took a broadly situated cognition position (Compton and Jansen 1990). For me the 1989 EKAW was largely a debate about situated cognition. It was a delight and the debate continued at the 1989 Banff KAW and the next few KAW but then gradually died out. In 1998 there was a special issue of this journal on situated cognition and knowledge-based systems, but it was no longer an area of active interest for this community, although of

preprint of : Compton, P. (2013). "Situated cognition and knowledge acquisition research." *International Journal of Human-Computer Studies* **71**: 184-190. A special issue of the journal on 25 years of knowledge acquisition research

significance elsewhere. Clancey for example had moved to a study of activities and influences in the workplace as critical factors in designing human-computer systems (Clancey, Sachs et al. 1998).

The 1991 paper by Sandberg and Wielinga summarises what was probably the general view (Sandberg and Wielinga 1991): that the ideas of situated cognition are interesting, but the attack on the conventional paradigm of cognitive science and AI is overstated and there is no need to change the paradigm. In particular they stated:

> An expert system that solves problems through behaviour similar to that of a human expert, can be viewed as a theory of the problem-solving behaviour of that expert. However, good fit between model and data does not necessarily mean that all underlying machinery (representations, processes, hard - wetware) is identical.

That is, the emphasis in situated cognition on the inability of experts to articulate their knowledge is misplaced, because the aim is not to capture knowledge, but to model problem-solving. The resulting emphasis on modelling problem-solving rather than capturing the expert's problem-solving knowledge, is probably partly a result of the situated cognition debate, but can there be another impact?

Situated cognition as a positive statement

The ideas of situated cognition have largely been presented as stating what people can't do, but it is probably more useful to re-express this as what situated cognition has to say about what people can do. Situated cognition suggests that people never explain the process of how they reach a conclusion; rather they create explanations to justify their decision in the particular context of the decision. Clancey went as far as saying that there is no knowledge in the mind, knowledge only comes into existence in its expression in a context (Clancey 1997). But situated cognition does not deny that people are able to make sound decisions and do justify these decisions in context, so putting this more positively:

1. Any rule or knowledge statement a person provides will apply to the current situation, but will be over-general, as it will not identify all the features that distinguish this context from all possible contexts.

The second issue is the nature of the concepts people use when identifying features.

2. A concept used in a rule or knowledge statement, will point to some feature in the data, but this feature may not have a well-defined meaning which applies outside the context.

A radiologist might say a patient has asbestosis because the image of the lung shows a honeycomb pattern. But there is no definition of a honeycomb pattern; it is the name the radiologist is using when pointing to part of an image. If one asks for a definition it is very difficult to obtain a precise definition which applies to all contexts.

Although stating what people cannot do or have difficulty with (according to situated cognition), these two statements are also positive statements as to what people are able to do. The first statement implies that people do identify features that distinguish the case – although not in all possible contexts, while the second statement emphasizes that people do identify features in the data – although unable to give definitions of the features that are always applicable. Combining these two statements of situated cognition as a positive statement of a minimum of what people can do, we come up with something like:

> *If a person states that a different decision should be made in two different situations, then they can identify features in the data that do actually distinguish the two situations.*

Perhaps the person is wrong, and the same decision should be made for both situations, or perhaps both decisions are wrong, but if a person is rational and states that two different decisions should be made then they have identified some difference in the two situations. They may also misname or make up a name for the difference, but if they are rational they will only claim the situations should be treated differently if they believe they have identified some difference in the data. This is essentially the Principle of Rationality:

The remainder of this paper (briefly) considers some of the major themes in the knowledge acquisition workshops and conferences in relation to situated cognition, and in particular the idea that at a minimum people can identify features that differentiate cases they claim are different.

Software Engineering and Modelling

A central theme at the workshops through much of the 90s was the notion of modelling problem solving. Rather than simply asking the expert for their knowledge, research focussed on identifying the types of problems that occurred and the different problem-solving methods that could be applied to problems. The ideas was that the knowledge engineer would come to a problem armed with a library of re-useable methods (whether fully specified and coded or not). Chandrasekaran initially suggested identifying tasks independent of the domain knowledge (Chandrasekaran 1983) while Clancey suggested there were broad classes of problem solving methods such as heuristic classification (Clancey 1985). A wide range of research emerged based on the idea that such knowledge level modelling would enable an engineer to approach building a knowledge-based system in a much more systematic way.

It is impossible to review all this work, so I will refer only to CommonKADS (Schreiber, Akkermans et al. 1999). This provided a comprehensive and systematic approach to developing a knowledge-based system. There is no doubt that the systematization that came through CommonKADS and other modelling work, has made a fundamental contribution to the understanding and overall software engineering of knowledge based systems. (Schreiber, Akkermans et al. 1999) also includes a section on eliciting knowledge as the authors are well aware of the problems of eliciting knowledge from experts, but this section of their work essentially outlines previous techniques such as card sorting, laddering and repertory grids (discussed further below). However, this does not remove the problem identified by situated cognition; as the CommonKADS authors note elsewhere:

> Although methodologies such as CommonKADS support the knowledge acquisition process in a number of ways (e.g. by providing modelling constructs and template models) experience shows that conceptual modelling remains a difficult and time-consuming activity (Speel, Schreiber et al. 2001).

They note further:

> This is not to say that it is impossible in principle to make tacit knowledge explicit, but it is difficult and the standard knowledge engineering repertoire does not include techniques to support this (Speel, Schreiber et al. 2001).

This is the problem identified by situated cognition: that experts do not provide knowledge that covers all possible contexts. I think it is probably fair to say that methodologies like CommonKADS provide enormous assistance in providing resources in framing and focussing the particular elicitation task, but the problems identified by situated cognition still remain.

Although not specifically assessing the impact of methodologies like CommonKADS, it is interesting to note the 2008 survey of Zacharias (Zacharias 2008). 64 developers with an average of 6.6 years of experience answered most of the survey questions. 60% indicated their knowledge bases frequently gave the incorrect answer and 34% indicated that sometimes they gave incorrect results. The biggest need was identified as debugging/verification tools. Debugging here means adding and modifying knowledge to get the knowledge base to give the correct answers in all circumstances. Although the survey asked about process, it did not specifically seek information about whether better processes such as CommonKADS were perceived as a need.

In summary, the value of modelling and software engineering approaches is to ensure that one gets to the point of acquiring knowledge from experts in a reliable and systematic way – but the challenge of over generalised knowledge remains.

Ontologies and the Semantic Web

The idea of a library of problem-solving methods went hand in hand with a more systematic approach to knowledge representation independent of any particular domain, and this became a dominant theme at the workshops. One concern has been the theoretical bases for ontologies and metadata and more generally for logically sound reasoning and representation and Gaines predicts advance here will be of major importance in the development of the semantic web (Gaines 2012). Of more immediate concern here are practical tools such Protégé (Gennari, Musen et al. 2003) which has 200,000 registered users according to its website. Although Protégé is a general framework and toolkit the emphasis in Protégé has moved to a primary focus on developing ontologies, now particularly for the semantic web. They key question is again how such technologies relate to ideas of situated cognition. It seems there is little relation, for example although Protégé provides a very powerful environment it does not itself address knowledge elicitation questions, with standard elicitation technology being provided by plug-ins e.g (Wang, Sure et al. 2006) as well as rule technology (O'Connor, Knublauch et al. 2005).

It seems that since the central concern is to provide a principled approach to representing domain knowledge there is little direct concern with the issues raised by situated cognition – a different problem is being addressed. However, the problems of situated cognition still seem likely to arise with respect to useability, merging and alignment of ontologies. Ontology merging is a major research issue, but the challenge is not simply that different terminologies have emerged, but that people disagree even on ontology alignment (Tordai, van Ossenbruggen et al. 2011). Tordai et al. also also comment: "Humans rarely have problems with disambiguating the meaning of words in a discourse context". To develop large ontologies requires collaborative work (Konstantinou, Spanos et al. 2010) and there has been a wide range of research on this topic, but it is unclear whether a consensus amongst ontology developers is sufficient to provide useability. UMLS the well-established unified medical language system has over one million concepts, five million synonyms and hundreds of terminologies. As pointed out by Rosenbloom et al. the issue is not how complete UMLS might be, but how end-users can be supported in using it (Rosenbloom, Miller et al. 2006).

Knowledge Creation and Elicitation

There are a range of long-standing techniques that assist in eliciting knowledge but I will focus here on techniques based on Kelly's Personal Construct Psychology (Kelly 1955). These ideas were introduced to the knowledge acquisition community by Shaw and Gaines (Gaines and Shaw 1980) and John Boose (Boose 1984). The central technique in Personal Construct Psychology is to ask a person to select three objects in a domain of interest and then ask in which way two of these objects are alike and different from the third. This is based on the same principle as situated cognition: if someone says two objects are different they must be able to identify different features. Using the third object helps the user identify the more important differentiating features. The most accessible modern version is at http://repgrid.com/.

Although based on the same idea of distinguishing objects, experts construct knowledge with different terminologies and disagree with each other's terminologies (Shaw and Woodward 1988). They also note that these experts are well used to disagreeing with each other. I suggest that people are able to share ideas while disagreeing about terminology because conversations are about specific problems and specific situations and terminology differences are clarified through discussion of the differences between different situations – and the more concrete the situations the more easily this can be done. So despite the power and value of the PCP approach in assisting in the creation and articulation of knowledge – and its close relation to situated cognition, it does not of itself ground the constructs (axes of difference) in differences in actual data. This is not a criticism, but simply an observation.

Ripple-Down Rules (RDR)

There has been a series of RDR papers presented at the knowledge acquisition workshops over the years, particularly at the Pacific Rim Knowledge Acquisition Workshops, covering a wide range of different research, but it does not qualify as a major theme, as the papers have been presented by a fairly small sub-community. A review of much of this work can be found in (Richards 2009), The reason for presenting RDR here is because of the way it relates to situated cognition.

RDR was motivated by the problem that knowledge is only provided in context and proposed a refinement structure as way of limiting the context in which knowledge is then used (Compton and Jansen 1990). It turned out that this refinement structure was useful in machine learning and a number of RDR learners were published including Gaines Induct (Gaines and Compton 1992), the basis of the well-known RIDOR program from Weka; however, for knowledge acquisition from humans it gradually became clear that the refinement structure was largely irrelevant and the real strength came from asking people to differentiate between cases. The refinement structure was first supplemented with the idea of looking at case differences to assist in developing a rule but Kang's thesis showed that using a flat rule structure the same or better results were achieved simply by selecting features to differentiate cases (Kang 1996).

The essential idea was that an expert system would be put into use, with or without some initial rules, and its output monitored. Whenever it gave incorrect advice or failed to give advice for some data, the expert would be asked to specify the correct piece(s) of advice the system should have given and identify the features in the data that justified this advice being given. As the knowledge based developed cases were stored (generally the cases for which previous rules had been added) so if the features selected by the expert also occurred in these previous cases, they were presented to the expert to either decide that the new conclusion(s) should apply to the previous cases or to select further features from the present case to distinguish it from previous cases. The resulting rule is then automatically located in the knowledge base, as specified by the particular RDR method used. There are many assumptions here, for example that there is a language available that describes the features the expert wishes to identify, but at the core the method reduces to the task suggested by the situated cognition that a person should be able to select features to distinguish cases which they consider are different.

Data on how well the method works comes from Pacific Knowledge Systems Pty. Ltd[1]. (PKS). PKS logs the time its customers take from when they call up a case, which needs different conclusion until they have finished adding rule(s) that exclude previous stored cases. The log data necessarily also includes time spent on interruptions. Over many knowledge bases and many different domain experts the log data show it takes only a few minutes to add a rule (Compton, Peters et al. 2011). For example the median time add a rule to knowledge bases of between 1,000 and 2,000 rules is less than three minutes. For a single very large knowledge base in the data set, the median time to add a rule after 10,000 rules was 10 minutes; however, the longer time is largely processing time, rather than expert time. Regardless of the number of cases to be excluded, experts tend to produce a sufficiently precise rule to exclude all cases after no more than two or three

---

[1] I have a small shareholding in Pacific Knowledge Systems and provide it with some part-time consulting

have been seen. Figure 1 shows a detailed example of a single knowledge base. This is a knowledge base of about 3,000 rules, which to date has been used to interpret over 7 million patient results for general biochemistry tests and provides about 250 different pieces of advice (which may also be combined in reports). About 10% of the rules define intermediate features (heuristic classification). The data shown in Figure 1 is up until early 2011, but the knowledge base is still in routine use. The individual points show the raw data for the time taken to add a rule and exclude past cases. The line shows median time over the last 50 rules; the median is used because the data includes interruptions. The vertical lines separate the years over which the knowledge acquisition occurred and the hours show the total time spent each year on knowledge acquisition.

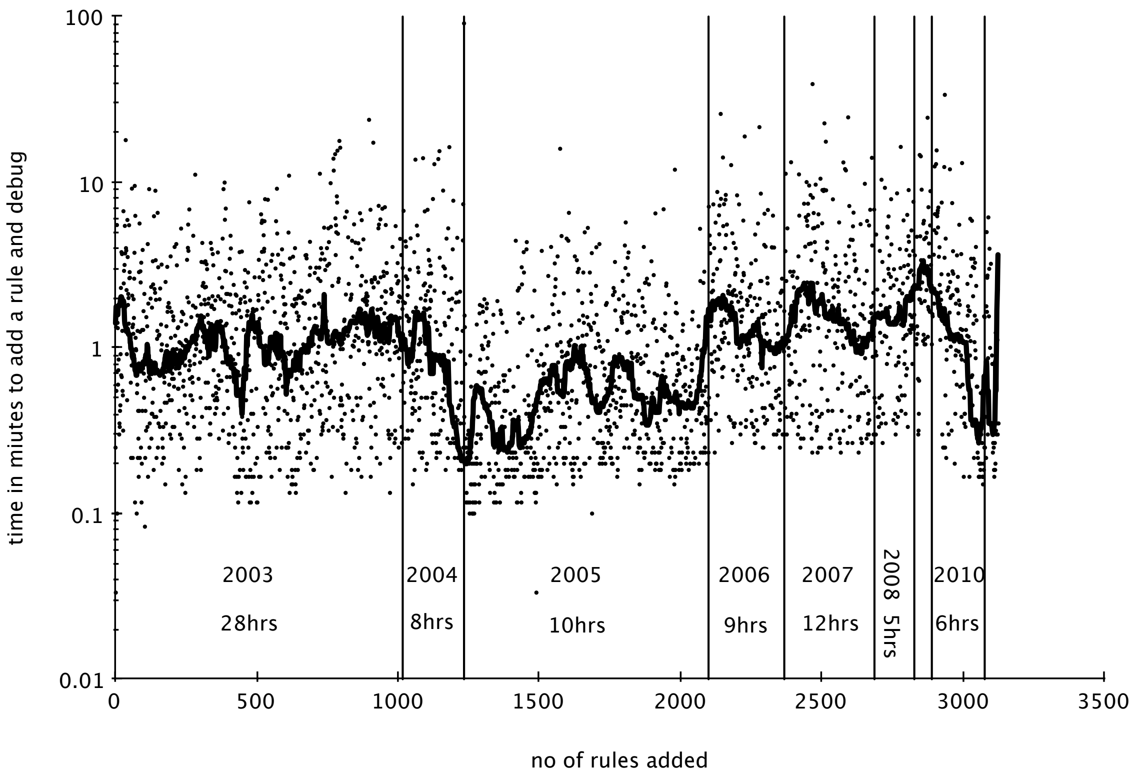

Figure 1. The time taken to add each rule and exclude conflicting cases

Clearly rule addition (and excluding conflicting cases) is an extremely rapid task and it continues for a long time; i.e. cases keep emerging for which the existing rules are over-general or missing, but the total time taken is extraordinarily small. It is beyond the scope of this paper to consider questions such as repetition in the knowledge base, but references to a range of such can be found in (Richards 2009). It should also be noted that this knowledge base is in use and has been since 2004. The point of presenting this data here is simply to highlight that the situated cognition task of asking people to identify the features that differentiate cases is extremely simple even in a rich and complex domain such as time-varying biochemistry results for a large number of analytes and including other clinical information that may be available in a laboratory information system.

PKS is not the only company to have developed RDR technology. Ivis uses RDR mainly for translating product descriptions to standard ontologies for TESCO to facilitate selling on-line (Sarraf and Ellis 2006). IBM uses RDR cleansing and augmenting data for big datasets (Dani, Faruquie et al. 2010; Nambiar, Faruquie et al. 2011) .

Clearly one needs the scaffolding provided by methodologies like CommonKADs or Protégé if approaching a new and unknown problem. In the domain of interpreting laboratory data PKS already has such scaffolding in place, so the problem reduces to the task of obtaining expert heuristics, where the situated cognition case-difference approach comes into play.

Extracting knowledge from data

I include here both machine learning and the wide range of research presented over the years on areas such as information extraction from text. Such research avoids the problems of situated cognition and the disconnect between elicited knowledge and data by learning from data so that the resulting system can be applied to more examples of the same type of data. Such systems are not a total panaceas as their development is always limited by the range and appropriateness of training data.

Knowledge acquisition from people is normally only undertaken in domains of specialist expertise; however, in common sense tasks such as dealing with text, people are well able to identify features that differentiate different pieces of text e.g. named entities start with a capital, although such rules seem hopelessly over-general. Dani et al showed that accumulating such over-general rules resulted in systems that outperformed machine learning (Dani, Faruquie et al. 2010) and it has also been shown that adding such rules can also improve the performance of general systems when applied to somewhat different domains (Kim and Compton 2012; Kim and Compton 2012).

Conclusions and future possibilities

Although situated cognition was a significant focus of discussion in the early years of the knowledge acquisition workshops, this interest largely faded as the focus moved primarily to comprehensive frameworks to deal with the problems of representation and reasoning and overall software engineering rather than actual acquisition.

When situated cognition is re-expressed as a minimum capability of what people can do, it emphasises that they can and do differentiate concrete situations and identify the differentiating features. Ripple-Down Rules suggests this can be made an extraordinarily rapid and simple task, so the question arises of where this might be of value in relation to other interests. Clearly it could integrate with the various modelling approaches that have been developed, but there is still further research to be done on how knowledge is incrementally added to the knowledge base for the full range of problem types.

Of more importance is how case differentiation might be used in the semantic web and in bridging the gap between the semantic web and Web 2.0; Konstantinou et al. see a role for end-users “filling the gaps” (Konstantinou, Spanos et al. 2010). Given that it seems relatively easy to build case-differentiation systems for text (Dani, Faruquie et al. 2010; Kim and Compton 2012; Kim and Compton 2012), perhaps this could be used in providing and correcting semantic-web annotation. The same simple approach is already used to standardise product descriptions (Sarraf and Ellis 2006).

One initial application might be in developing text-processing resources, particularly for underresourced languages. Crowd-sourcing has been shown to be useful in labelling text (Snow, O'Connor et al. 2008), but rather than applying machine learning to then develop automated text processing tools, could the crowd actually provide text-processing rules, by providing case-differentiation? For example, if people tag the word "run" as a verb or a noun in different contexts, then they must also be able to identify differences in the contexts, e.g. the verb "run" is preceeded by a noun or preposition. Such a rule seems far too simple and context-specific – but it provides a lot more information than simple context-specific tagging of "run" as a verb. Could large numbers of such rules (and mechanisms to arbitrate between them) provide crowd-sourced knowledge bases for text-processing, particularly for underresourced languages? Could the same approach be used to learn to automatically annotate web pages for the semantic web?

One obvious challenge in achieving this is to develop the mechanisms to arbitrate between rules provided by many individuals, but perhaps the more challenging task is to provide languages that support people's unerring ability to identify features that distinguish different situations, but are rich enough for the semantic web.

Acknowledgements

I suspect that I must be near the record for participation in the knowledge acquisition conferences and workshops, as I have participated in 29 of the 50 events to date. My on-going involvement has been largely because of the welcome and encouragement I received from Mildred Shaw and Brian Gaines.

Pacific Knowledge Systems provided the data shown in Figure 1. The Australian Research Council has supported research on Ripple-Down Rules over many years.

References

Boose, J.H., 1984. Personal construct theory and the transfer of human expertise. Proceedings AAAI-84, 27-33.

Chandrasekaran, B., 1983. Towards a taxonomy of problem solving types. AI Magazine Winter/Spring, 9-17.

Clancey, W.J., 1985. Heuristic classification. Artificial Intelligence 27, 289-350.

Clancey, W.J., 1997. Situated Cognition: On Human Knowledge and Computer Representations. Cambridge University Press.

Clancey, W.J., Sachs, P., Sierhuis, M., van Hoof, R., 1998. Brahms: simulating practice for work systems design. Int. J. of Human Computer Studies 49(6), 831-865.

Compton, P., Horn, R., Quinlan, J.R., Lazarus, L., 1989. Maintaining an expert system. in: Quinlan, J.R. (ed.), Applications of Expert Systems. Addison Wesley, London, pp. 366-385.

Compton, P., Jansen, R., 1990. A philosophical basis for knowledge acquisition. Knowledge acquisition 2, 241-257.

Compton, P., Peters, L., Lavers, T., Kim, Y.S. (2011). Experience with long-term knowledge acquisition. Proceedings of the sixth International Conference on Knowledge Capture, KCAP 2011. Banff, Alberta, Canada, ACM**,** pp. 49-56.

Dani, M.N., Faruquie, T.A., Garg, R., Kothari, G., Mohania, M.K., Prasad, K.H., Subramaniam, L.V., Swamy, V.N., 2010. Knowledge Acquisition Method for Improving Data Quality, in: Services Engagements, in,IEEE International Conference on Services Computer (SCC), Miami, IEEE, pp. 346-353.

Gaines, B., 2012. Knowledge Acquisition: Past, Present and Future. Int. J. of Human Computer Studies (this issue).

Gaines, B.R., Compton, P.J., 1992. Induction of Ripple Down Rules, in: Adams, A., Sterling, L, (eds.), AI '92. Proceedings of the 5th Australian Joint Conference on Artificial Intelligence, Hobart, Tasmania, World Scientific, Singapore, pp. 349-354.

Gaines, B.R., Shaw, M.L., 1980. New directions in the analysis and interactive elicitation of personal construct systems. Int. J. of man-machine studies 13(1), 81-116.

Gennari, J.H., Musen, M.A., Fergerson, R.W., Grosso, W.E., Czubezy, M., Eriksson, H., Noy, N.F., Tu, S.W., 2003. The evolution of Protégé: an environment for knowledge-based systems development. Int. J. of Human Computer Studies 58, 89-123.

Kang, 1996. Multiple Classification Ripple Down Rules (PhD thesis). University of New South Wales.

Kelly, G., 1955. The Psychology of Personal Constructs. Norton, New York.

Kim, M., Compton, P., 2012. Improving Open Information Extraction for Informal Web Documents with Ripple-Down Rules, in: Richards, D., Kang, B.H., (eds.), Knowledge Management and Acquisition for Intelligent Systems (PKAW 2012), Springer, pp. 160-174.

Kim, M., Compton, P., 2012. Improving the performance of a Named Entity Recognition system with knowledge acquisition, in: The 18th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2012). Springer, Galway City, pp. 97-113.

Konstantinou, N., Spanos, D., Stavrou, P., Mitrou, N., 2010. Technically approaching the semantic web bottleneck. Int. J. of Web Engineering and Technology 6(1), 83-111.

Menzies, T., 1998. Towards situated knowledge acquisition. Int. J. of Human Computer Studies 49(6), 867-893.

Nambiar, U., Faruquie, T.A., Prasad, K.H., Subramaniam, L.V., Mohania, M.K., 2011. Data Augmentation as a Service for Single View Creation, in,IEEE International Conference on Services Computing (SCC), Washington, IEEE, pp. 40-47.

O'Connor, M., Knublauch, H., Tu, S.W., Musen, M.A. (2005). Writing rules for the semantic web using SWRL and Jess. Protégé With Rules Workshop. Madrid.

Richards, D., 2009. Two decades of Ripple Down Rules research. The Knowledge Engineering Review 24(2), 159–184.

Rosenbloom, S.T., Miller, R.A., Johnson, K.B., Elkin, P.L., Brown, S.H., 2006. Interface Terminologies: Facilitating Direct Entry of Clinical Data into Electronic Health Records. Journal of the American Medical Informatics Association 13, 277-288.

Sandberg, J.Wielinga, B., 1991. How situated is cognition, in,12th International Joint Conference on Artificial Intelligence, Sydney, Morgan Kauffman, pp. 341-346.

Sarraf, Q., Ellis, G., 2006. Business Rules in Retail: The Tesco.com Story. Business Rules Journal 7(6), http://www.BRCommunity.com/a2006/n2014.html.

Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W., Wielinga, B., 1999. Knowledge Engineering and Management: The CommonKADS Methodology. MIT Press, Cambridge Mass.

Shaw, M., Woodward, J., 1988. Validation in a knowledge support system: construing and consistency with multiple experts. Int. J. of man-machine studies 29, 329-350.

Snow, R., O'Connor, B., Jurafsky, D., 2008. Cheap and fast---but is it good?: evaluating non-expert annotations for natural language tasks. in: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pp. 254–263.

Speel, P., Schreiber, A.T., Van Joolingen, W., van Heijst, G., Beijer, G., 2001. Conceptual models for knowledge-based systems. in: Kent, A., Williams, J.G. Encyclopedia of Computer Science and Technology. Marcel Dekker.

Tordai, A., van Ossenbruggen, J., Schreiber, G., Wielinga, B. (2011). Let's agree to disagree: on the evaluation of vocabulary alignment, in: K-CAP '11: Proceedings of the sixth international conference on Knowledge capture. ACM, Banff, pp. 65-72.

Wang, Y., Sure, Y., Stevens, R., Rector, A., 2006. Knowledge elicitation plug-in for Protégé: Card sorting and laddering, in: The Semantic Web ASWC 2006, pp. 552-565.

Winograd, T., Flores, F., 1987. Understanding computers and cognition. Addison Wesley, Reading, MA.

Zacharias, V., 2008. Development and Verification of Rule Based Systems—A Survey of Developers, in: Bassiliades, N., Governatori, G., Paschke, A., (eds.), RuleML 2008, Orlando, Springer-Verlag, pp. 6-16.