Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2009), *23*, 327–329. Printed in the USA.

doi:10.1017/S0890060409990023

EDITORIAL

Problem Solving Methods: Past, Present, and Future

DAVID C. BROWN

Department of Computer Science, Worcester Polytechnic Institute, Worcester, Massachusetts, USA

This special issue of *AI EDAM* is devoted to invited papers concerned with problem solving methods (PSMs). Its goal is to examine whether PSMs have fulfilled their early promise, to determine whether difficulties still remain, and to make predictions about their use in the future development of knowledge-based systems, including those built and delivered over the Web.

PSMs are highly relevant for *AI EDAM* and its readers. Knowledge-based computational support for engineering design, analysis, and manufacturing has provided motivation for the study of PSMs, and the area can clearly still benefit from them. For example, "heuristic classification" is at the heart of many selection problems, such as selecting an appropriate feature to include in a design, picking the right material, using the right analysis tool, or deciding on the processing of a material. In addition, intelligent parametric design, and configuration, are both within the scope of the Journal.

A PSM describes how to reason using knowledge to achieve a goal. A task, usually specified by its goal and its inputs and outputs, is something to be done, such as determining a type of malfunction from symptoms. There are usually a variety of ways that a task can be done; that is, one or more PSMs might be appropriate. The kinds of tasks being addressed require the use of knowledge to be tackled properly and may need the application of heuristics. Such tasks are often associated with expert reasoners, such as doctors or designers.

Examples of tasks are specifying the values of a given set of parameters in response to some requirements (i.e., parametric design) or deciding to which of a given set of known situations a given situation belongs (i.e., classification). Appropriate PSMs, as identified in the literature, might be "propose and revise" and "cover and differentiate."

A PSM may be primitive or may decompose the task into subtasks, for which other PSMs may be appropriate. A PSM usually has a simple control structure that determines its pattern of inferences. It also specifies the knowledge needed and what role that plays in the PSM. Each PSM comes with certain stated assumptions about its potential connections with tasks and with knowledge. PSMs are linked to domains using ontologies.

One general goal of the work in PSMs is to be able to aid in the building of a knowledge-based system for a selected task by configuring a set of PSMs selected from a library and linking that with the appropriate knowledge. This might be human controlled or done (semi)automatically. Some recent work has focused on doing these processes over the Web.

A useful, partial list of PSM references generated by me, including many key papers, can be found on the web at www.cs.wpi.edu/~dcb/courses/CS538/References07.html.

All of the contributions for this Special Issue were invited. The invited contributors are B. Chandrasekaran, Bill Clancey, Deiter Fensel, and Mark Musen, along with their coauthors. Together they represent some of the "major players" in the development of PSMs during the period from about 1984 onward. Despite being invited, all submissions were anonymously reviewed and then revised in order to improve their quality.

The authors were asked to respond in their papers to some or all of a set of questions and challenges about PSMs provided by the Editor. The main questions were the following:

1. Have the original goals of the PSM movement been achieved, for example, those from the *role limiting* methods research (McDermott, 1988) and the *generic tasks* (GT) research (Chandrasekaran & Johnson, 1993)?
2. Is there still good work concerned with studying and developing PSMs, or have PSMs been branded a failure and largely forgotten?
3. What is the future of PSMs?

Additional more detailed questions and challenges that invite discussion include the following:

- Are there any more PSMs? In the PSM work from about 1983 to 2000 there are many lists: Schreiber et al. (1999) list 13 types of tasks, for example. Are there more to be found? Do we need them?
- What is an appropriate grain size for PSMs? In the GT line of research (Chandrasekaran & Johnson,

Reprint requests to: David C. Brown, Department of Computer Science, Fuller Labs 131, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609, USA. E-mail: aiedam@cs.wpi.edu

1993) the conclusion was reached that what were earlier thought to be primitive GTs were actually made up of other finer grained tasks, which were themselves useful tasks. While analyzing my prior work on the routine design GT, presented as the Design Specialists and Plans Language (DSPL), 13 ingredient activities were found, all potentially of general use (Brown, 1992).

- Does it make sense to open the use of PSMs to a wider range of less controlled reasoning activity, not just the traditional expert tasks? If so, is the question of how many PSMs there are, or their grain size, still relevant? That is, can we recognize what *is not* a PSM?
- Is it appropriate to consider PSMs as assembled dynamically in response to needs or retrieved as complete units? Much of the literature on PSMs tends to consider retrieving complete units. Although a number of more recent authors acknowledge the need for task–subtask decomposition with the resulting reasoner being an assembly of PSMs, there has not been a lot of attention paid to that configuration process.
- Can complete systems built using PSMs be "automagically" assembled? Can complete systems built using PSMs be assembled by humans? Is there room for both?
- Should usability be focused on more? Some recent work on PSMs (e.g., Motta, 1999; Fensel, 2000) relies on expressing knowledge (e.g., assumptions and goals) in some form of formal, logical language. Logic is notoriously user unfriendly for anyone without the right training. Although there is certainly room for this approach—what I will refer to as "PSM heavy"—surely it makes sense to have "PSM light" versions available as well, that is, versions using less intimidating languages, perhaps with less flexibility, or provided as "toolkits" in the original GT style?
- When faced with a choice of library and a choice of PSMs for the same task within each library, which PSM should be selected from which library? That is, other issues in addition to whether it can do the task, such as speed, space, accuracy, and reliability, need to be involved.
- What are the recent successes of the use of PSMs from PSM libraries? Has all the hype been justified? Why should they be considered as successes, and where have they been documented?
- What are the recent failures of PSM research? Why should they be considered as failures, and where have they been documented?
- Are there other directions within AI in which the PSM research can lead, for example, incorporating PSMs with analogical reasoning, with spatial reasoning, or with sensing?
- Currently the PSM research effort has become strongly tied to the use of ontologies. Is that a good thing? For example, if ontologies in fact are "use dependent" (as Noy & McGuinness, 2001, and others point out) and vary according to use or task, then how many ontologies do you need for PSM utility?
- Has the latest work on integrating PSM concepts with the Web been distracted by Semantic Web technology, including ontology engineering? Combining Web services and integrating data from different sources raises its own interesting challenges and has the seductive power of potential worldwide use. However, does the original need for knowledge-based problem solving systems still remain?
- There is some doubt about whether the problem of describing and using existing PSMs has been completely solved. If this is the case, why is it appropriate to open PSM research to applications in the even more varied and potentially powerful area of Web services? Because this is a harder, more open case, can they be described, discovered, and combined properly?

The Special Issue starts with Chandrasekaran's contribution, "Problem Solving Methods and Knowledge Systems: A Personal Journey to Perceptual Images as Knowledge." He was one of a small group of researchers to propose PSMs. This paper summarizes how his ideas have changed over the last 20 years. Much of this has already been documented in previous publications. However, his more recent work described here concerns perceptual and kinesthetic images, their role in reasoning, as well as their role in constructing concepts and interacting with external representations.

The second paper is "Software-Engineering Challenges of Building and Deploying Reusable Problem Solvers" by O'Connor, Nyulas, Tu, Buckeridge, Okhmatovskaia, and Musen. They address the issue of the lack of adoption of PSM techniques, discuss the need for tools to implement PSMs, and focus on their experiences and challenges constructing the BioSTORM system. Their conclusion about PSMs is mostly positive, but they warn that it is not a panacea and there are still major software-engineering challenges involved in their use for complex systems. However, they feel that building their system would have been much more difficult without PSMs and ontologies.

The third paper, by Clancey, Sierhuis, and Seah, is "Workflow Agents Versus Expert Systems: Problem Solving Methods in Work Systems Design." Their work is very much informed by the PSM vision of abstracting system components. However, despite seeing the PSM research as a "reasonable, well-grounded engineering phase of research," they feel that it did not focus enough on the actual, complex, people-based systems in which PSM-based systems need to be deployed and there was not enough focus in the PSM research on interactions with the "real world." They seem to suggest a tension between the "ill-structured" nature of such situations, and the somewhat "tidy" nature of PSMs.

The final paper in the issue is "Problem Solving Methods in a Global Networked Age" by Domingue and Fensel. Their paper is concerned with using PSM-derived ideas to support a "world-scale network" with billions of users and services. They believe that PSM-based mechanisms "provide the only viable approach" to systematically deal with such dynamic activity on such a grand scale.

In conclusion, I agreed with Prof. H. Akkermans (personal communication, 2008) when he suggested that it was as if the people in the knowledge systems and engineering community totally lost their memory when they turned to the Web. Many valuable lessons had been learned, and, even though PSM research was not finished, it was stifled by the change of direction.

PSMs are by their very nature practical. They support application building based on the results from much serious research. In a climate where funding fashion pushes the latest powerful technology as if one size fits all—neural, fuzzy, evolutionary, Bayesian—applications will be stunted and poorly understood unless proper analysis is done of the necessary ingredients of the reasoning and knowledge. PSMs reject the one size fits all approach and accurately reflect the complexity of intelligent applications.

As the papers in this Special Issue reveal, there are many interesting research issues that remain in and around the area of PSMs. The detailed questions and challenges presented here also make it clear that much remains to be done. However, even without all of the answers, based on 20 years of research there are still practical advantages to be had by adopting a PSM point of view. It is time for a revival of PSM research and practice.

ACKNOWLEDGMENTS

The authors and I thank the eight anonymous reviewers who volunteered to provide feedback on the "readability" of the draft papers. In addition, Dr. Clancy provided very valuable feedback about Chandrasekaran's draft paper. I also thank the three other invited authors, Profs. Motta, Puppe, and van Harmelen, who considered taking part, but who were unable to contribute for various reasons. In addition, I acknowledge Profs. Akkermans and Sleeman who were both very encouraging about putting this Special Issue together.

REFERENCES

Brown, D.C. (1992). The reusability of DSPL systems. *Workshop on Reusable Design Systems, 2nd Int. Conf. AI in Design*, Carnegie Mellon University, Pittsburgh, PA.

Chandrasekaran, B., & Johnson, T.R. (1993). Generic tasks and task structures: history, critique and new directions. In *Second Generation Expert Systems* (David, J.M., Krivine, J.P., & Simmons, R., Eds.), pp. 239–280. New York: Springer–Verlag.

Fensel, D. (2000). *Problem-Solving Methods: Understanding, Description, Development, and Reuse*. LNAI, Vol. 1791. New York: Springer.

McDermott, J. (1988). Preliminary steps towards a taxonomy of problem-solving methods. In *Automating Knowledge Acquisition for Expert Systems* (Marcus, S., Ed.), pp. 225–256. Boston: Kluwer Academic.

Motta, E. (1999). *Reusable Components for Knowledge Modeling*. Amsterdam: IOS Press.

Noy, N.F., & McGuinness, D.L. (2001). *Ontology development 101: a guide to creating your first ontology.* Accessed at http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html on June 8, 2007.

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*. Cambridge, MA: MIT Press.

David C. Brown is a Professor of computer science and has a collaborative appointment as a Professor of mechanical engineering, both at Worcester Polytechnic Institute. He attained a BSc from North Staffordshire Polytechnic in 1970, an MSc from the University of Kent in 1975, an MS from The Ohio State University in 1977, and a PhD from The Ohio State University in 1984, all of which are in computer science. He is a member of the Association of Computing Machinery, American Association of Artificial Intelligence, IEEE Computer Society, and International Federation of Information Processing Working Group 5.1. Dr. Brown is the Editor in Chief of *AI EDAM* and is on the editorial boards of several journals, including *Concurrent Engineering: Research and Application*, *Research in Engineering Design*, and *International Journal of Design Computing*. He is a Vice Chair of the Design Computing and Cognition conferences and has been a member of the program and advisory committees for many conferences, as well as for international workshops on topics such as computational creativity, configuration, and design rationale.