Prelude for Experience Mining (Re-)Using Relevant Experience for Smart Decision Support

Jolán Velencei, Óbuda University, Budapest, Hungary
Viktor Dörfler, University of Strathclyde Business School, Glasgow, UK
Zoltán Baracskai, DoctuS Consulting, Budapest, Hungary
Jaszmina Szendrey, University of Strathclyde Business School, Glasgow, UK

Abstract

After 10 years in teaching a variety of topics in decision making one of the authors decided to develop a tool that can help decision makers to consolidate their knowledge around a particular decision problem. The three remaining authors have joined the R&D team over the next two decades; the work resulted in the Doctus knowledge-based system which is currently 25 years in development. In the most recent incarnation of Doctus we are able to identify relevant patterns from previous decisions by other decision makers, learning from which can be helpful to the decision makers with the decision situation at hand.

In this paper we introduce the latest incarnation of our Doctus KBS, featuring three distinct ways of reasoning; the newest 'third way of reasoning' supports reusing previous decision experience. As far as we know, Doctus is currently the only decision support tool capable of identifying relevant experience as well as learning from it. In this paper we provide a prelude for how the new solution should work in real-life decision support.

***Keywords**: Decision making, Decision support, Experience mining, Patterns of cognition, Smart decision*

Track: *Management*

Word count: *2.478*

1. Introduction

The four of us have spent many years working *with*, *for*, and *on* executive decision takers in various ways. We have worked as executive coaches, we have been doing research and published academic papers and books about executive decision takers and how to support them, we have acted as knowledge engineers and developed the knowledge-based expert system shell Doctus, we have been teaching decision takers in classroom environments and developing them in their organizational contexts. Our research, teaching, and consultancy mutually affect each other. We use the feedback from our knowledge engineer work as input for our research, we include our research findings in our teaching and also use them in our consultancy, etc. We have also used our multi-faceted experience to design post-experiential, post-MBA courses for decision takers. Three years ago, we started a drama-based course for executives and executive coaches (Baracskai, Dörfler, & Velencei, 2010) and based on this experience two years ago we have developed a more personalized approach for supporting and developing executives in their natural organizational context using dramas as tools for decision support (Baracskai, Dörfler, Stierand, Szendrey, & Velencei, 2012). This year we return to our knowledge-based expert system Doctus and merging it with the last few years' experience in using dramas we are developing what we now call *'smart decision support'* – a knowledge-based decision support that uses relevant experience. In this paper we are providing a prelude for *smart decision support*, i.e. we are focusing on the big picture in which such decision support approach makes sense and on the conditions under which we believe it will work. We also explain some of the technical details, but we do not go in great depth about these.

We regard the world in which organizations operate chaotic, meaning that changes are unexpected in terms of direction, size and impact (see e.g. Drucker, 1969; Friedman, 2009; Handy, 1995; Nordström & Ridderstråle, 2002; Prahalad, 1998). In this world we contrast the lean and mean organizations with those characterized by communityship (Mintzberg, 2006, 2009) based on the engagement (Mintzberg, Simons, & Basu, 2002) of the members of the community. With some admitted stretch of assumptions, we regard the executives of the lean and mean organizations are the action heroes. They have no doubts, they always 'know' what the right thing is (usually short-term profitability) and how to achieve it (most often by cost cutting and downsizing), and they deliver the performance to achieve it. (Cf Mintzberg, 2007) In our experience action heroes hardly need any other decision support than optimization or, at least, quantitative methods calculating the single right solution. Of course, we are not against calculations in general, only we believe that most of what the top executive does cannot be calculated, unless they are action heroes. We have no interest in supporting the action hero decision takers, and our tool is not adequate for them. Many executives are action heroes, but many are not. We watched executives struggling with doubts as they could not know the outcomes of their actions in the chaotic world of business. Their world is not deterministic, they cannot control everything, and often they do not even know how to distinguish good from evil in a particular situation. Gradually we realized that they are drama heroes (cf heroic leadership vs. engaging management introduced by Mintzberg et al., 2002). Drama heroes fight for something no one else fights for, doubt themselves and can never find out whether it was for a good cause or not. Yet, this struggle seems to be the essence of the drama hero. This kind of executive, in an organization characterized by communityship, is who we supported using our drama-approach and for whom we have now developed the *smart decision support* approach.

2. Knowledge-based Decision Support

The term *knowledge-based decision support* refers to using software tools called knowledge-based systems (KBS), expert systems (ES), or knowledge-based expert systems – as they utilize knowledge bases and are expected to perform at the level of a human expert. In contrast with the typical tools of operational research (OR), expert systems reason about a problem rather than calculate a solution and they are expected to provide explanations about their outcomes. (Brachman et al., 1983; Feigenbaum, 1982, 1992; Feigenbaum, McCorduck, & Nii, 1988; Gill, 1995) The Doctus KBS belongs to the broad category of *symbolic* systems, meaning that the knowledge representation it uses is based on symbolic logic in the form of "if… then" production rules. A knowledge-based system, such as Doctus, consists of two main parts: the software tool called *shell*, which contains the inference engine but is empty in terms of the content and the *knowledge base*, which is the representation of the expert knowledge. Although there are many shells available in the decision support market, most of them require the knowledge base to be entered in a form borrowed from logical programming languages, such as PROLOG or LISP; in contrast Doctus uses a fully graphical interface and does not require any coding.

The overall process of building an expert system is called *knowledge engineering*. The knowledge engineer builds the knowledge representation about the decision situation at hand by acquiring knowledge from domain experts and entering it in the form of "if… then" rules into the shell. As a general rule, the knowledge base is ready when the expert(s) whose knowledge has been modelled agree(s) with it – this does not satisfy the requirements of rigorous validation but works well in practice.

The most widespread use of expert systems is through *rule-based reasoning* (RBR) also called *deductive reasoning*. In RBR the domain experts are expected to explicitly articulate the aspects of the decision as well as the "if… then" rules between them; and these rules are then applied to the decision alternatives, providing an evaluation as well as an explanation. Often, however, the most experienced domain experts find it difficult to articulate their knowledge in the form of "if… then" rules. In such situations, it is sufficient if the domain experts articulate the decision aspects and then recall cases from their experience together with the evaluation; Doctus can then infer the rules from the cases of past experience. This use of expert systems is called *case-based reasoning* (CBR) or *inductive reasoning*. Doctus delivers CBR using an entropy-gain method based on a modified ID3 algorithm. (Quinlan, 1986, 1993) There are about 100 software packages delivering RBR and perhaps a dozen of these is also capable of CBR. As far as we know, Doctus is the only tool on the market to perform a 'third way of reasoning' we call *reductive reasoning*. Reductive reasoning always follows CBR and uses the informative attributes identified during CBR to generate a new rule-based knowledge base.

3. Smart Decision Support: What Makes Experience Relevant?

It is important to note that any decision situation will need its own knowledge base, as expert level knowledge can only be represented for a relatively small problem area, such as a particular decision, rather than for the entire domain of expertise (Davenport & Prusak, 2000). This is why expert systems, although regarded as a very sophisticated type of decision support, are relatively rarely used: each decision requires a new knowledge base, and building a knowledge base for a particular decision takes about 8 consultancy days, over a two month period (i.e. one day a week), normally involving 3 knowledge engineers. In other words, knowledge-based decision support is expensive. Although there were attempts for building instant knowledge bases that can be implemented directly, or with only minor customizations, these approaches have never been fruitful – the offered knowledge bases were either specific or not delivering the expert level of insights. For years we tried to figure out the conditions under which experience, and thus a knowledge base, from one decision could be adapted in another decision situation. As knowledge engineers we were able to see that two knowledge bases are similar – but this always happened post hoc or, at least, well into the second decision support process. To make it useful, we need to identify relevant experience at the beginning of the process. Furthermore, the relevant experience may be available in an entirely different context, which makes finding it particularly difficult.

The problem of identifying relevant experience is not specific to the area of decision support; for example, this is the central problem to the whole area of benchmarking. The original idea of finding the *best practice* did not work, as it is very difficult to conceptualize what it means that a particular process is conducted in the best way, when different organizations need to be compared. Thus the idea of *lead practice* was introduced, according to which we need to look at a leader in our particular area of business, and look at how they are conducting the process we are interested in. If they are the leader in the area, it seems reasonable to assume that they are good at conducting this particular process as well. This, however, imposes a great limitation to finding relevant experience, i.e. it is limited to our particular area of business, not to mention the problems of information availability. It is easily possible that an organization from an entirely different area of business is particularly good at what we are interested in.

Having introduced the case-based reasoning in Doctus some 20 years ago, we have soon realized that a similar CBR outcome (case-based graph), actually does represent the similarity of experience, and it is completely independent of the area of business. The reason is that the knowledge base, and thus the case-based graph is the representation of the expert knowledge localized at a particular decision. However, we faced two substantial difficulties: (1) we could not explain what we mean by 'similar' case-based graphs and (2) we had no means of transferring the relevant experience from one knowledge base to another. But we had the germ of the idea. Now, nearly two decades later, we have introduced *reductive reasoning*: we can use the case-based graph as a source from which a new rule-based knowledge base is constructed and this can then be refined for the new decision situation.

4. Commentary

Instead of a conclusion, we offer a commentary, as we believe that we are at the beginning of a new era in knowledge-based decision support, and there is nothing to conclude yet. We have a relatively detailed big picture of the generic context in which this type of smart decision support could work, and we have the initial technical solution necessary to make it work. We also have our first implemented smart decision support, carried out as a consultancy project. The client was a Foundation who needed smart decision support for evaluation and selection of innovation project ideas from a variety of disciplinary areas that it was going to finance. For this decision we re-used relevant experience of a previous R&D decision from the very different context of an oil company. In this decision support process the relevant experience was still identified by the involved knowledge engineers, but the re-use was supported with the reductive reasoning. Currently we are developing a meta-knowledge-base that would support the identification of the relevant experience, however, even the current semi-manual solution helped reducing the overall cost of the decision support project by 25%.

References

Baracskai, Z., Dörfler, V., & Velencei, J. 2010. Shakespeare's Witches: Drama-based Executive Coaching, *AoM 2010: The Seventieth Annual Meeting of the Academy of Management*. Montréal, QE.

Baracskai, Z., Dörfler, V., Stierand, M., Szendrey, J., & Velencei, J. 2012. Dramatic Dilemmas: Drama-based Decision Support, *AoM 2012: The Seventy-second Annual Meeting of the Academy of Management*. Boston, MA.

Brachman, R. J., Amarel, S., Engelman, C., Engelmore, R. S., Feigenbaum, E. A., & Wilkins, D. E. 1983. What Are Expert Systems? In F. Hayes-Roth & D. A. Waterman & D. B. Lenat (Eds.), *Building Expert Systems*: pp. 31-57. Reading, MA: Addison-Wesley Publishing.

Davenport, T. H. & Prusak, L. 2000. *Working Knowledge: How Organizations Manage What They Know* (paperback ed.). Boston, MA: Harvard Business School Press.

Drucker, P. F. 1969. *The Age of Discontinuity: Guidelines to our Changing Society*. London, UK: Heinemann.

Feigenbaum, E. A. 1982. *Knowledge Engineering for the 1980s*. Stanford, CA: Department of Computer Science, Stanford University.

Feigenbaum, E. A., McCorduck, P., & Nii, P. H. 1988. *The Rise of the Expert Company: How Visionary Companies are Using Artificial Intelligence to Achieve Higher Productivity and Profits*. New York, NY: Vintage Books.

Feigenbaum, E. A. 1992. A Personal View of Expert Systems: Looking Back and Looking Ahead. Stanford, CA: Department of Computer Science, Stanford University.

Friedman, T. L. 2009. *Hot, Flat, and Crowded: Why The World Needs A Green Revolution - and How We Can Renew Our Global Future*. London, UK: Penguin Books Ltd.

Gill, T. G. 1995. Early Expert Systems: Where are They Now? *MIS Quarterly*, 19(1): pp. 51-81.

Handy, C. 1995. *The Empty Raincoat: Making Sense of the Future*. London, UK: Arrow Business Books.

Mintzberg, H., Simons, R., & Basu, K. 2002. Beyond Selfishness. *Sloan Management Review*, 44(1): 67-74.

Mintzberg, H. 2006. The Leadership Debate with Henry Mintzberg: Community-ship is the Answer, *Financial Times*.

Mintzberg, H. 2007. Productivity Is Killing American Enterprise. *Harvard Business Review*, 85(July-August): 25.

Mintzberg, H. 2009. Rebuilding Companies as Communities. *Harvard Business Review*, 87(July-August): 5.

Nordström, K. & Ridderstråle, J. 2002. *Funky Business: Talent Makes Capital Dance* (2nd ed.). London, UK: Prentice Hall.

Prahalad, C. K. 1998. Managing discontinuities: The Emerging Challenges. *Research Technology Management*, 41(3): 14-22.

Quinlan, J. R. 1986. The Induction of Decision Trees. *Machine Learning*, 1(1): 81-106.

Quinlan, J. R. 1993. *C4.5 Programs for Machine Learning*. San Mateo, CA: Morgan Kauffman Publishers.