REDUCTIVE REASONING

ZOLTÁN BARACSKAI[*], JOLÁN VELENCEI[*], VIKTOR DÖRFLER[**]

Abstract: *Today one of the most challenging problems of an organization is how to make the decision taking faster in order to improve efficiency. Excellence resides in the differences of people's tacit knowledge. In Knowledge Management excellence can be achieved if the leader delegates the Routine Decisions to others, while the Original Decisions are taken on basis of tacit knowledge. With Expert System Shell DoctuS, we can reduce the expressed rules to those meta-schemata that actually affect the decision. These meta-schemata are derived from a blend of explicit and tacit knowledge. The process of reducing the number rules we call reduction, which is the third kind of reasoning beside deduction and induction. The achievement is that we can get the same decision using the values of fewer attributes. According to our experience the benefits include knowledge discovery in every application (i.e. a part of tacit knowledge is made explicit) and the process of decision taking becomes faster and its cost is reduced.*

(UDC: 338.246 (470); JEL classification: L 86, O 32; Original scientific paper;
Received: June 26, 2005)

1. Background Knowledge

This section contains the background knowledge that we have built upon when working our way towards the solution. It is grouped into to subsections, one describing the three types of decisions and the other presents the knowledge used by the decision taker. The background literature is mostly based on literature, though it also includes some of our previous results that are used in the paper.

1.1 Decision Types

Simon [10] distinguished the programmed from the non-programmed decisions. Programmed decisions are those that frequently occur thus one can have elaborated procedures how to handle them; these could literally be programmed. The non-programmed decision is a novel situation what one meets for the first time thus there cannot be any elaborated procedures available; such situations need tailored procedures, they can certainly not be programmed. The programmed and the non-programmed decisions are non-existing extremities (black and white) of a continuum (greyscale) in which the real life decisions can be found. As a further development of this conception we describe the decisions using three corner stones (Figure 1):

Reflex Decisions: We were interested whether there are thinking processes underlying every decision. We have observed that there are habitual activities that we do without thinking, as by instinct only; a private example could be buying a cigarette, and business examples are paying the salaries or controlling the stock.

Routine Decisions: There are decisions taken by managers following some set of rules, of which rules they have explicit knowledge. A private example could be buying a car (not the first one, of course), and business examples are to decide about the type of the framework contract we want with a customer or a supplier. The knowledge used in these decisions comes from the experience we got by taking similar decisions and we are aware of the decision aspects (attributes) and of the rules between the values of these attributes. Routine decision is the closest real-life resemblance of programmed decisions but it they are not the same. Although routine decisions incorporate a vast amount of attributes and rules, the importance of individual remains in focus. Sometimes we may need a new logical rule between the values of the attributes, or consider some new attributes or ignore some of the old ones.

[*] Doctus Co., Csikihegyek utca 8 II, 1118 Budapest, Hungary, {zoltan, jolan}@doctus.info

[**] University of Strathclyde, Graham Hills Building, 40 George Street, Glasgow, G1 1QE, United Kingdom, viktor.dorfler@strath.ac.uk

Fig. 2. Decision Types

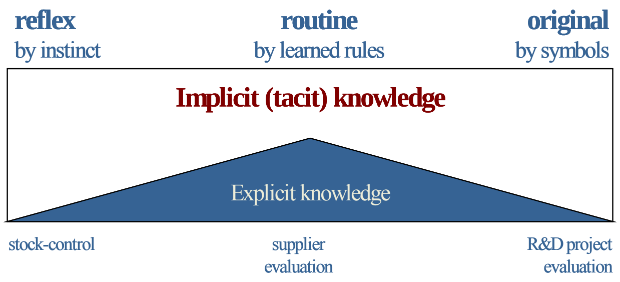
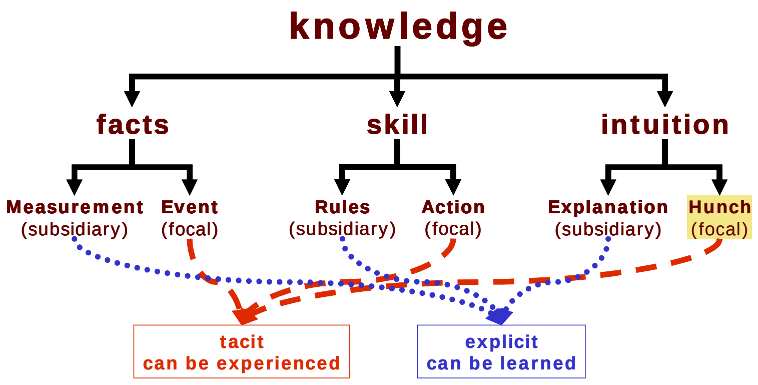


Fig. 2. Knowledge Types



Original Decisions: There is a first time for every decision – these we call the original decisions. Such is the loss of virginity for a private example, and for business examples we can think of all R&D projects' evaluations (i.e. not the small refinements but the genuine novelties). These situations come the closest to the non-programmed decisions, although, they are not the same as we always now something even about the novel situation. Typically, original decisions are the responsibility of the leader, who does not have any experience with it; however, it does not mean that she/he is novice decision taker. The experienced e-leader has meta-schemata that appear in almost every decision. This decision type has an additional feature: there are no well defied attributes. The attributes are defined using symbols and metaphors.

In our view e-leaders should make reflex decisions without spending any time. Excellence is up to original decisions while spending time to take reflex decisions has negative effects on competitiveness. Defining rules for routine decisions facilitates delegation of decisions. If all decisions are considered to be new challenges (i.e. original decisions) the e-business looses on efficiency; if all are considered to be routines the e-business becomes rigid and dies.

1.2 Knowledge Types

Ryle [9] divides knowledge into "knowing how" and "knowing that". The same categories appear at Anderson [1] as procedural knowledge and declarative or descriptive knowledge. "Knowing what" (meaning knowing what to do), the typical knowledge type of the leader, is included in "knowing how" not in "knowing that". Investigating "knowing what" Minsky [4] concludes that positive knowledge (knowing what to do) differs from negative knowledge (knowing what not to do). Both are essential. Minsky [3] also distinguishes the special knowledge from the common sense.

Knowledge is subjective: different people have different knowledge. Cognition does not exist without cognitive individual. [5] Knowledge can only be objective if it is about the reality. Thus the personal knowledge [6] is objective and subjective at the same time. A group or an organization cannot have knowledge, the organizational knowledge and the organizational learning are only metaphors describing the impact of the group and the organization on the change of personal knowledge. Tacit knowledge is at the edge of all kinds of creative acts; it is hard to verbalize but can be experienced. Polanyi [7] introduced the concept of tacit knowledge which underlies the explicit knowledge. Polanyi described the explicit knowledge, which systemizes and organizes the existing knowledge. However, the knowledge can never be conveyed utterly. It is impossible to explain how to kiss or to write a poem. It is necessary to recognize the unfamiliar signs, hunch the undiscovered paths, and to accept innovations.

Based on these previous investigations we distinguish three types of knowledge and we use Polányi's [6] distinction of focal and subsidiary awareness as subtypes (Figure 2):

Facts: The true-false dichotomy is useful to understanding knowledge where the facts are measured by means of standards. Measurement made by anyone at any time always provides the same results. The focal parts of facts are the events, while the subsidiary parts are the measurements. We emphasize that there is no need for knowledge management in cases where the well founded factual knowledge is taken as basis. Data management can be used instead.

Skills: There are difficult skills, for example to balance a bike; however they seem to be easy: we can use them without understanding the physical rules of balancing. Our subsidiary skill stores recognition of letters but we are not aware of it when reading. Speech would also be difficult if we checked the appropriateness of each word before articulation: *"Human competency cannot be copied. Everyone develops his/her own competency – through mistakes, thinking and repetition."* [11] Competency means to know how to interpret. *"Expertise or Competence in Polanyian sense implies the ability of know-how within a certain domain and the ability not only to submit to the rules but also by reflection influence the rules of the domain or the tradition."* [11] A competent individual is not only aware of the rules of discipline but knows how and when to use them. Competence is not a feature but a relation between knowledge and knower. Our concept is that pre-set competence does not exist, i.e. competence depends on both the know-how of the individual and the environment.

Intuition: The hunch cannot be explicit. We cannot define the process of judgment what we know is that the judgment has been made and it is viable. It is like going through a dark tunnel to people who are happy to live in the «light of facts». Very often, e-leaders take their decisions upon their hunch, which suggest satisfactory results. A hunch is not a statement or sentence but an idea, an intellectual vision, cognition and «aha» experience. The intuitive result can be explained in hindsight; this is the subsidiary part of intuition, which should be up to the bivalent formal logic. In our view intuition is useful when rules in our background knowledge are insufficient. The basis of our model is the tacit knowledge. We can achieve improvements in knowledge management if we identify and map some of tacit knowledge.

2. Solution

Our solution is presented in two parts: First we are introducing the solution, the DoctuS Knowledge-Based System Shell [2], in general terms, starting from the advantages and disadvantages, emphasizing the principles of how the different ways of reasoning can support the decision taker. After that comes an illustration of the application using a case study.

2.1 DoctuS Knowledge-Based System

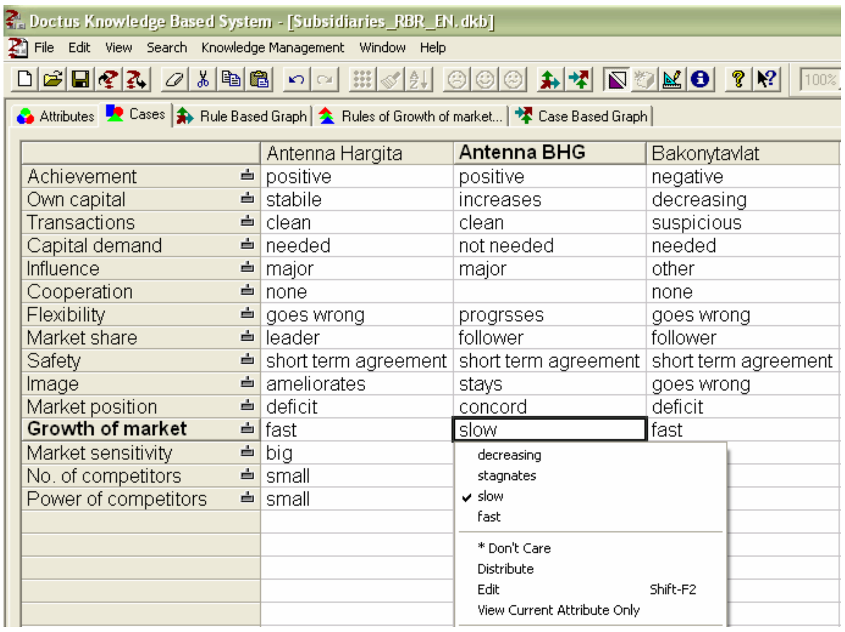
DoctuS, uses symbolic representation, that is to say symbolic artificial intelligence. The first advantage of symbolic representation of knowledge is that it's humane. The symbolic logic is the only solution that does not quantify the user's preferences. E.g. the person, whose knowledge is being modeled, thinks that the beautiful is a better value than the ugly. Nobody thinks that the beautiful is 3,6 times better than the ugly. Using symbolic logic we do state nothing like that. Into the symbolic knowledge base of an expert system we can put the knowledge in form as we talk or think about it. Therefore we get to the second advantage, which is the transparency, easy modification and fine-tuning of the knowledge base.

Numerical signs can be treated only as symbols, so if we want to use numerical data, first we have to transform them into symbols. There are several solutions. The easiest is if instead of saying numbers, the expert tells something like «too much», «not enough», etc. The trouble is that this cannot be automated. There are, however, statistical and fuzzy cluster analyzing algorithms that can be automated. So this disadvantage is eliminated.

If there are many symbols; there will be plenty of rules, which is a real disadvantage. Today, this is not a problem of computing capacity. The expert-level knowledge is few thousand of cognitive schemata that mean few thousand rules, which modern software can easily handle. Though it is hard to acquire lots of rules from the expert; the use of multi-step reasoning helps.

Further disadvantage is that the expert has to articulate the rules, thus there is no access to

Fig. 3. Cases described


Doctus Knowledge Based System - [Subsidiaries_RBR_EN.dkb]
File Edit View Search Knowledge Management Window Help
100%
Attributes
Cases
Rule Based Graph
Rules of Growth of market...
Case Based Graph
decreasing
stagnates
✓ slow
fast
* Don't Care
Distribute
Edit Shift-F2
View Current Attribute Only

	Antenna Hargita	**Antenna BHG**	Bakonytavlat
Achievement	positive	positive	negative
Own capital	stabile	increases	decreasing
Transactions	clean	clean	suspicious
Capital demand	needed	not needed	needed
Influence	major	major	other
Cooperation	none		none
Flexibility	goes wrong	progrsses	goes wrong
Market share	leader	follower	follower
Safety	short term agreement	short term agreement	short term agreement
Image	ameliorates	stays	goes wrong
Market position	deficit	concord	deficit
Growth of market	fast	slow	fast
Market sensitivity	big		
No. of competitors	small		
Power of competitors	small		

the tacit knowledge. In symbolic approach the representation of the common sense is unrealizable; we are all experts of it. We cannot articulate much of it; most of our common sense waggles between focal skills and focal intuition (thus tacit).

There are two basic ways of reasoning in DoctuS. If we use *deduction* also called *rule-based reasoning* an expert gives the attributes and its values, and puts the attributes in a multi-step graph and defines the «if... then» logical rules between the values of the attributes. Cases (decision alternatives) are described by case features, i.e. we choose one values for each attribute for every case. A knowledge engineer helps the expert to put in words what she/he knows. We call this process knowledge acquisition. This is part of long, complex process called knowledge engineering.

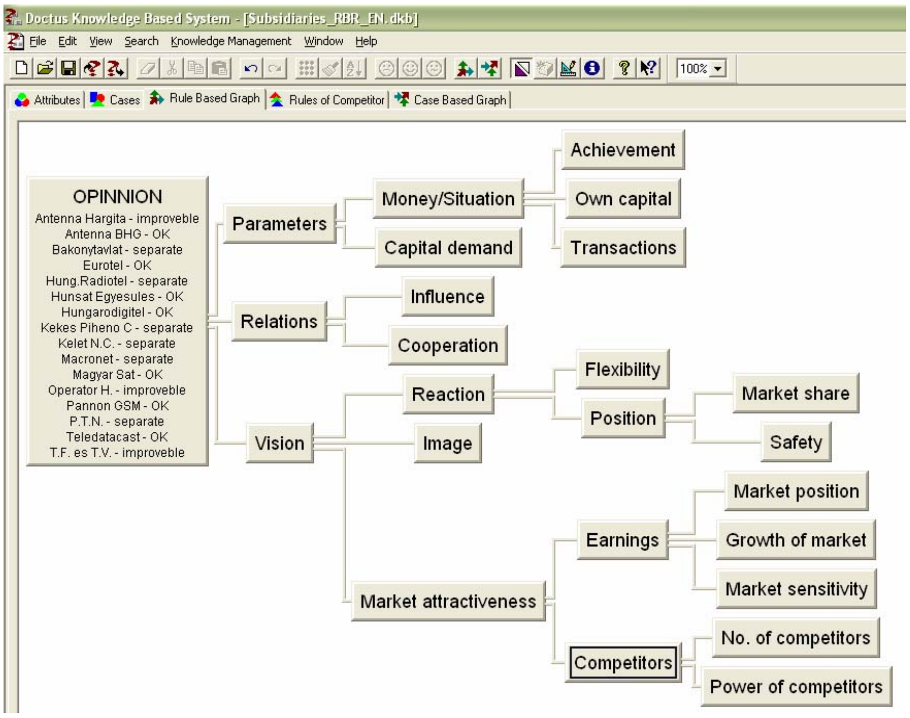
The impossibility of extracting the tacit knowledge is valid for the deductive reasoning, also called rule-based reasoning. There is a solution: instead of acquiring rules, acquire the cases of experience, from which DoctuS software deduces the rules. This is called *induction* or *case-based reasoning*. Originally the case-based reasoning was inherited from quantitative decision support. Its essence was – and in many computer programs it still peeps out behind the symbolic solution's veil – to define some metrics, and distance built on it, which will be the measure of similarity. For a new case the nearest one – the most similar – is searched from the case-base. In symbolic logic cases described with the same rules are taken as similar. If we use induction, the expert names the cases, for which she/he chooses attributes, than values for each attribute. For this we need cases that already occurred and the results are well-known. Through the process of knowledge acquisition and knowledge engineering the knowledge engineer assists in the expert's work. The knowledge acquisition's output is a decision tree (called Case Based Graph).

The resulting decision tree from the inductive reasoning can be converted into a single-level rule-based knowledge base. This is the third way of reasoning, which is not an entirely standalone type of reasoning; it mixes the previous two types. As during this conversion the number of the attributes is reduced to those appearing in the case-based graph, this type of inference is called *reduction*. Reduction gets its importance in the delegation of the decision, as it will be shown it the next section.

2.2 Case study

In rule-based reasoning, we acquired knowledge by building a knowledge base using DoctuS' deductive module. Here we present a case study in DoctuS on evaluation of business units of a Hungarian broadcasting company. The decision is, whether «to separate» the unit, to re-evaluate its position later («tomorrow»), it is

Fig. 4. Decision proposal in Rule-Based Reasoning



OPINNION
Antenna Hargita - improveble
Antenna BHG - OK
Bakonytavlat - separate
Eurotel - OK
Hung.Radiotel - separate
Hunsat Egyesules - OK
Hungarodigitel - OK
Kekes Piheno C - separate
Kelet N.C. - separate
Macronet - separate
Magyar Sat - OK
Operator H. - improveble
Pannon GSM - OK
P.T.N. - separate
Teledatacast - OK
T.F. es T.V. - improveble

«improvable», or it is «ok». The experts of well-defined domains tried to retrieve the rules they used, but was hidden in their, for long years accumulated, tacit knowledge. First, they defined several attributes, which they thought the decision was based on. For each attribute, then the possible values were determined, for specifying the rules and later the case features. These were filled into DoctuS. Decision takers had to establish relations between the attributes using «if… then» logic. For defining these, first a deductive graph is constructed, then decision takers define «if… then» rules between the linked attributes in each node of the graph. To acquire the rules, we follow the decision taker's activities for several days. According to the decisions taken, we built a prototype, which was later refined. Once this part was completed, the experts named cases where the modeled decision was taken, and described the circumstances using the attributes previously acquired. The cases they used were to be real ones that were already closed, in order to know the results of the decisions too.

While defining the case features, it often happens that experts realize they have forgotten about a relevant attribute or a possible value for one of the attributes. This time, for the «Growth of market» the expert originally defined only three values: «decreasing», «stagnates» and «fast»; while evaluating the second case, he realized that the growth of market was actually «slow». So a new value was added, and than used to describe the case as shown on Figure 3. Each new attribute and value helps to get a clearer picture, to describe the expert's knowledge more precisely. When the knowledge base is modified the rule-based graph and the rules too are to be adjusted.

This kind of knowledge base helps experts to extract their tacit knowledge. On the basis of the attributes, their relations (the rule-based graph and the rules) and the cases described, DoctuS can reproduce the rule-based reasoning of the expert, and suggest an evaluation for each case (a decision proposal), as seen on Figure 4.

If the suggested decision is not the same as the expert's decision, then the knowledge base should be refined. New attributes, relations, rules can arise from this process. The knowledge base is finished when the results are identical with the expert's opinion.

Experts and decision takers usually express more rules than they actually use, for the sake of reliability. With DoctuS, we can reduce the number of rules to those meta-schemata that really affect the decision. These meta-schemata derive from both explicit and tacit knowledge. We call this reduction. As result we get the same

Fig. 6. Case-Based Graph with polarization

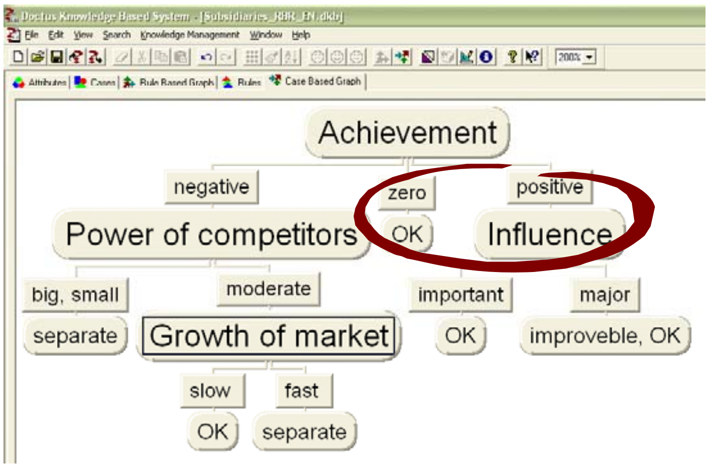
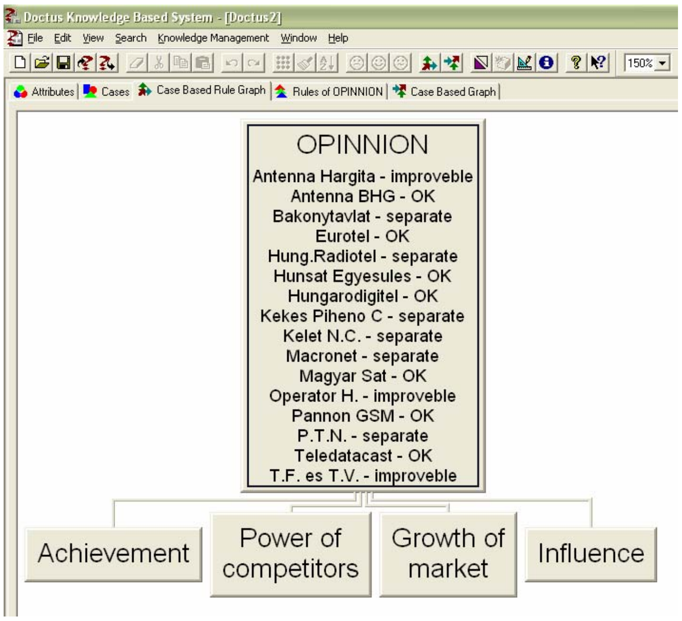

Doctus Knowledge Based System - [Subsidiaries_HBK_EN.dkb]
File Edit View Search Knowledge Management Window Help
200%
Attributes Cases Rule Based Graph Rules Case Based Graph
Achievement
negative
zero
positive
Power of competitors
OK
Influence
big, small
moderate
important
major
separate
Growth of market
OK
improveble, OK
slow
fast
OK
separate

Fig. 6. Case evaluations in the Case-Based Rule Graph


Doctus Knowledge Based System - [Doctus2]
File Edit View Search Knowledge Management Window Help
150%
Attributes Cases Case Based Rule Graph Rules of OPINNION Case Based Graph

OPINNION
Antenna Hargita - improveble
Antenna BHG - OK
Bakonytavlat - separate
Eurotel - OK
Hung.Radiotel - separate
Hunsat Egyesules - OK
Hungarodigitel - OK
Kekes Piheno C - separate
Kelet N.C. - separate
Macronet - separate
Magyar Sat - OK
Operator H. - improveble
Pannon GSM - OK
P.T.N. - separate
Teledatacast - OK
T.F. es T.V. - improveble

Achievement
Power of competitors
Growth of market
Influence

decision using fewer attributes. We will show here how this is done in DoctuS.

A decision depends not only on the quantity of the attributes, but also on the given values defined for each attribute. What we define as acceptable by defining the rules is heavily relying on these values. If the expert only defines «good» and «bad», the cases can hardly be compared. If she/he defines too many values – according to our experience with DoctuS, if it more than five –, that will also cause trouble in refining the knowledge base. The more the decision taker can stay between the two extremes, the more possibility there is for fine-tuning the evaluation.

DoctuS' second, inductive module can help finding relations and defining rules using case-based reasoning. The process is slightly different from the above discussed. First, the decision taker defines the attributes and their values, than names real cases, but this time the outcomes are also filled in. At this point DoctuS generates a case based graph classifying the cases. (See Figure 5)

In our case the case-based graph shows, that the most informative attribute for evaluating a business unit is the «Achievement». For example, if «Achievement» is «negative», and the «Power of competitors» is «moderate», then the «Growth of market» has also to be considered; if it is «slow», then the company is «ok». That is how you can read the branches of the case-based graph. (See Figure 5)

We have to emphasize that it is not the computer that decides or shows the right way; it is always the expert who has to decide whether the presented graph describes her/his process and way of thinking.

With the help of the «Polar» function of DoctuS, we can find the limit between the defined values for each attribute, above which the solution is acceptable, and under which it is not. So finally we get back to two categories, and can reduce the number of rules. Note that this creates a more refined model than «good»/«bad», because this does not automatically mean, that only the «excellent» would be accepted, or only the «bad» rejected. For example, on Figure 5 the «Influence» is polarized.

To find the most informative attributes in induction (case-based reasoning) Doctus uses a modified ID3 algorithm, originally developed by Quinlan [8]. During the refinement of the model the expert often modifies the attributes appearing in the nodes of the case-based graph; the knowledge engineer pays attention that only attributes with similarly high informativity are used. Otherwise new attributes and/or values can be defined, similarly to the rule-based knowledge base. Once the expert has agreed that the graph is displaying her/his thinking the inductive reasoning is finished.

Having the result of induction ready we can step into the third type of reasoning, to reduction. This can simply be done by using the «extract rules» command; the result is a new rule-based knowledge base with a single-level rule-based graph, which we call a case-based rule graph to distinguish it from the usual rule-based ones. (See Figure 6) When we extracted the reduced decision taking rules from the tacit knowledge of a decision taker, we experienced that Occam's razor always worked: there were only few used, complex rules. The attributes in the reduced knowledge base are those that were in the accepted case-based graph. This means, that the reduced knowledge base provides the same decision proposals (case evaluations) as the original deductive knowledge base but it uses far fewer attributes and far fewer rules. This increases the efficiency by reducing the required time and cost. The name «reduction» is more than once justified.

3. Conclusion

Most of the time decision takers have to give an accurate account of their decisions. If they used DoctuS during the process, they just need point at a decision node anywhere in the tree structure to explain the outcome. The transparency provided by DoctuS guarantees reliability and confidence both during the decision taking process and afterwards. We suggest using DoctuS when:

- the time you have is very short;
- the environment is unpredictable;
- the decision problem is ill-structured;
- you do not have all the necessary information to use OR methods;
- you do not aspire towards the best decision, but towards an acceptable one;
- the judgment calls for intelligent reasoning.

References

1. Anderson, J.R.: The architecture of cognition. Harvard University Press, Boston, MA (1983).

2. Doctus Knowledge Based System Homepage. URL http://www.doctus.info. Accessed 28 October 2004.

3. Minsky, M.: Why People Think Computers Can't. AI Magazine, Vol. 3 No. 4 (1982). URL http://www.ai.mit.edu/people/minsky/papers/ComputersCantThink.txt, Accessed 29 October 2004

4. Minsky, M.: Negative Expertise. International Journal of Expert Systems, Vol. 7 No. 1 (1994) 13-19.

5. Neisser, U.: Cognitive Psychology. Meredith Publishing, New York, NY (1967).

6. Polanyi, M.: Personal Knowledge: Towards a Post-Critical Philosophy, Routledge, London (1962).

7. Polanyi, M.: The Tacit Dimension. Peter Smith, Gloucester, MA (1966).

8. Quinlan, J. R. The Induction of Decision Trees, Machine Learning, Vol. 1 No. 1 (1986) 81-106.

9. Ryle, G.: The Concept of Mind. Hutchinson and Co., London (1969).

10. Simon, H.A.: The New Science of Management Decision. Prentice-Hall, New Jersey, NJ (1977).

11. Sveiby, K.E.: The New Organizational Wealth. Berrett-Koehler Publishers, San Francisco, CA (1997).

REDUKCIONO ZAKLJUČIVANJE

Rezime: *Jedan od najizazovnijih problema neke savremene organizacije je kako donijeti odluku što brže u cilju poboljšanja efikasnosti. Savršenstvo je u razlikama podrazumijevanog znanja ljudi. U Menadžmentu znanja savršenstvo može biti dostignuto ako lider delegira rutinske odluke drugima, dok se originalne odluke donose na bazi podrazumijevanog znanja. Sa Expert System Shell DoctuS-om možemo redukovati pravila izražavanja na onaj meta-silogizam koji stvarno utiče na odluku. Ovaj meta-silogizam se izvodi iz mješavine eksplicitnog i podrazumijevanog znanja. Proces redukovanja broja pravila nazivamo redukcijom, što je treća vrsta zaključivanja nakon dedukcije i indukcije. Postižemo ga tako što možemo donijeti iste odluke korišćenjem vrijednosti sa manje atributa. Prema našem iskustvu, prednosti uključuju poznavanje otkrića u svakoj aplikaciji (tj. dio podrazumijevanog znanja je učinjen eksplicitnim) i proces donošenja odluka postaje brži i njegovi troškovi se smanjuju.*