A Review of Explanation and Explanation in Case-Based Reasoning[1]

Dónal Doyle, Alexey Tsymbal, Pádraig Cunningham

Department of Computer Science
Trinity College Dublin
Donal.Doyle@cs.tcd.ie

1. Introduction

With the arrival of more powerful computers and improved algorithms in machine learning, the availability of computer-based Knowledge Based Systems (KBSs) has increased rapidly. There is a wide range of KBSs available in many different types of domains, including medicine, finance, industry and technical diagnosis (Armengol et al, 2000; Ong et al, 1997; Rowe & Wright, 1993; Mark et al, 1996; Ye, 1995). Sales of KBS development tools are growing at a rate of about 16% per annum since 1988 (Durkin, 1996). For example, the French Bank "Evalog" has decreased its costs of processing loans by tenfold, helped minimise risks and increased processing capacity by using a KBS called "EvEnt" (Mao and Benbasat, 2000). However, case studies show that, in contrast to the availability of these systems, many KBSs that are installed in organisations are not being used (Majchrzak and Gasser, 1991).

Some possible reasons for the lack of usage of available KBSs is that the users do not understand the models used in the particular system (Majchrzak and Gasser, 1991), or they might not be convinced by the prediction given by the system (Brézillon and Pomerol, 1996). Other possible reasons could include a fear by system users that the system will eventually replace the user (Berry and Hart, 1990), or a fear that the use of KBSs may lead to a dehumanisation of some work practices (Shortliffe, 1992).

Consider one possible scenario occurring from the use of KBSs in the medical domain of the future.

> *"The semiconscious patient lies in a futuristic intensive care unit, tubes protruding, wires emerging from under the sheets and connecting to a host of monitor carts or wall-mounted devices, and intravenous fluids with computer-controlled infusion pumps circling the bed. The beeps of the monitors are not interrupted by footfalls of nursing staff, for health workers seldom have to enter the room. Instead, intelligent devices measure every pertinent physiological parameter, deciding how to adjust infusion rates, when to alter the respirator settings, and whether to sound alarms for the intervention of nurses or physicians."* (Shortliffe, 1992)

To some people this scenario for the clinical use of computers may appear unrealistic, due to current limitations of automation and ethical or legal constraints

[1] This material is based upon works supported by the Science Foundation Ireland under Grant No. S.F.I.-02IN.1I111.

(Druzdzel, 1996). It creates an image that if expert systems are developed in the medical informatics domain, it will result in a dehumanisation of the health care system.

In designing KBSs it must be remembered that verification is not enough, KBSs need to justify and be accountable for their predictions (Richards, 2003). The inclusion of explanations with predictions helps to make them more credible. After all, a physician using a diagnostic system is not supposed to follow an oracle's advice blindly, even if the system has shown reliable performance over time (Druzdzel, 1996). Belief in a system can be increased not only by the quality of its output but more importantly by evidence of how it was derived (Swartout, 1983).

Explanation is one of the most common methods used by humans to support decisions (Schank, 1986). Explanation and the ability to customise explanations are considered to be among some of the most important capabilities of KBSs (Stylianou et al., 1992). Research has shown that the use of explanations make KBS predictions more acceptable to the user [Ye and Johnson, 1995; Mao and Benbasat, 2000; Ramberg, 1996). Since MYCIN (Hayes-Roth and Jacobstein, 1994; Shortliffe, 1996) most KBSs provide some form of explanation.

Without understanding a systems prediction the user might accept or reject the systems advice for the wrong reasons. The use of explanation helps prevent incorrect acceptance or rejection of a KBS output. In order to prevent dehumanisation due to KBSs, users must have a sense of control over KBSs, rather than feeling that the KBSs are in control (Shortliffe, 1992). Including explanations keeps the user in control of the system.

Including explanations as part of a KBSs output does not automatically guarantee user acceptance for the KBS. It can be argued that bad explanations from a KBS could be considered worse than no explanation at all. Explanations that are very long and complicated can often confuse the user. Explanations can be unnecessarily long by including information that adds nothing to the user's belief (Wolverton, 1995).

This review paper discusses the use of explanations in KBSs, in particular in CBR systems, and the benefits of the use of explanations. Section 2 discusses the principles of CBR and the usefulness of explanations in KBSs before describing some important aspects of a good explanation. The next section discusses how some current KBSs attempt to improve the quality of their explanations. Section 4 moves on to focus on the task of explanation in CBR. The final section (Section 5) considers whether users in practice, find predictions more convincing with predictions, than without predictions.

2. Background

There are a vast number of KBSs currently in use by companies and organisations throughout the world. For example all of the top five Japanese companies have between 20 and 30 KBSs in use (Mizoguchi and Motoda, 1995). Many financial and insurance companies also use KBSs in their operations (Rowe & Wright, 1993; Meyer et al., 1992). Multinational companies such as General Electric, General Motors, use KBSs, allowing information to be accessible in all their companies worldwide. For example GE use a KBS for recommending plastic samples that might be useful for their customers (Cheetham, 2003). Other KBSs in use include DAI-DEPUR which is an environmental decision support system for the control and supervision of Municipal Waste Water Treatment Plants (Cortés et al., 2002) and the Alarm Correlation Module (MCA), a traffic monitoring and control system (Bandini, 2002).

However a lot of KBSs that are installed in companies are not in operation. According to Majchrzak and Gasser (1991), more than 50% of systems, which are installed in companies, are not being used. This section will take a look at possible reasons why KBSs are not being used and how the possible use of explanations supporting a systems prediction might increase a user's acceptance of the system. As the main emphasis of this paper is on Case Based Reasoning (CBR) in KBSs, the first part of this section will provide an overview of CBR.

2.1 Case Based Reasoning

The use of CBR predominately came from the research by Roger Schank and his students into cognitive science (Schank and Abelson, 1977; Schank, 1982; Kolodner, 1983; and Hammond, 1988). The inspiration for CBR came from a desire to understand how people remember information and are in turn reminded of information; and that subsequently it was recognised that people commonly solve problems by remembering how they solved similar problems in the past (Watson 1998). Unlike other forms of reasoning such as Rule-Based Reasoning, CBR does not draw conclusions by chaining together generalised rules. In CBR the main source of knowledge is in a memory of stored cases. Solutions for problems are derived by retrieving the most similar cases from its memory and adapting them to fit the given problem. A classic definition given by Riesbeck and Schank (1989) is

> *"A case-based reasoner solves problems by using or adapting solutions to old problems."*

The CBR approach is based on two observations of real world problems. The first is that similar problems tend to have similar solutions. The second is that the types of problems encountered tend to reoccur over time. When the two of these observations hold, it is worthwhile to remember and reuse prior reasoning. Therefore case based reasoning is an effective reasoning strategy (Leake, 1996).

The general model of a CBR process is often referred to as the CBR-cycle. This cycle generally comprises four activities:

Retrieve: Retrieve similar case(s) to the problem case.
Reuse: Reuse the solution(s) provided by the most similar case(s).
Revise: If necessary adapt the solution(s) to the retrieved case to better fit the problem.
Retain: If necessary add the new solution to the Case-Base once it has been confirmed or validated.

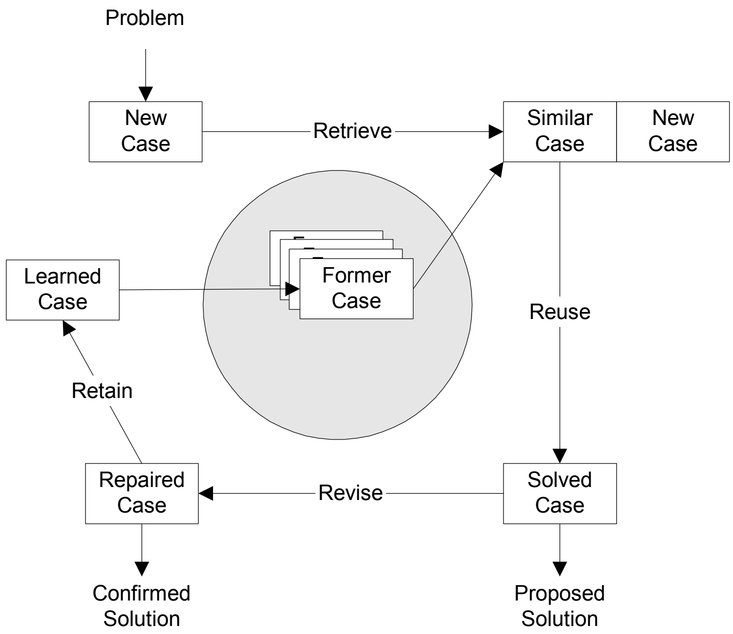


Figure 1: General model of Case-Based Reasoning (Aamodt and Plaza 1994)

There are numerous different techniques available to retrieve the most similar case in a case-base. One approach is to form a complete problem description and use this to select a relevant case. Alternatively an incremental approach can be taken where more discriminating features are used to discriminate between cases in a case base (Cunningham, et al., 1998; Cunningham et al., 1995; Owens 1991).

Possibly the most widely used technique used for retrieval in CBR are nearest neighbour techniques (Watson, 1997). Nearest neighbour algorithms all use a similar technique. The similarity between a target case, *q,* and a case in the case-base, *x,* are determined by calculating the similarity δ between each feature, *f,* in both cases. This similarity may then be scaled using a weighting factor, w_f. A sum of all scaled similarities is calculated to provide a measure of similarity between the two cases (the target case and the case in the case-base). This can be represented by the following equation:

$$Sim(\mathbf{q},\mathbf{x}) = \sum_{f \in \mathbf{F}} w_f \cdot \delta(q_f, x_f)$$

Similarities are usually normalised to fall within the range [0, 1], where zero is dissimilar and one being an exact match. Most CBR tools that use nearest neighbour techniques use algorithms similar to this, for example the Wayland System (Price and Pegler, 1995).

Once the similarity between the target case and all of the stored cases has been calculated, the most similar cases are retrieved (i.e. the cases with the highest similarity value). These cases are then reused to help solve the target case. In generating a solution for the target case, the retrieved cases are often revised and adopted to suit the target case.

Case adaptation in CBR is usually achieved by using rule-based systems on the retrieved cases. However an important reason for using CBR is because there is a lack of reliable domain knowledge. As a result, developers must acquire domain knowledge to implement the adaptation rules. This makes developing suitable adaptation techniques generally a difficult challenge in CBR (Kolodner, 1991; Leake 1994) as a major motivation in using CBR is often a lack of domain knowledge.

These difficulties have led many developers of CBR systems to not implement the adaptation stage of the cycle. These systems generally present the most similar cases to the human user, allowing them to perform the adaptation and evaluation on the presented cases. This framework is the basis for some successful CBR applications such as Clavier, Caber and Expert Help Desk System (Mark et al., 1996).

The final stage of the CBR cycle, "Retain", used to be performed by early CBR systems by simply storing each new case they generated. However this technique of retaining cases tends to make case-bases grow unnecessarily big in size, which generally slows down case retrieval. More modern approaches add new cases to the case-base when a failure occurs. This technique is known as "failure-driven learning" (Leake, 1996). When an inaccurate solution is provided by a CBR system, the failed solution can be repaired and added to the case base, preventing the same error happening again. In some systems information about the failure itself can be stored, so that it could provide warnings about possible future solutions (Bareiss, 1989; Kolodner and Simpson, 1989).

After examining CBR and how it works, how exactly does it compare in performance against other techniques in KBSs? According to Leake (1996) there are five main problems identified in real world KBSs that can be ameliorated by using CBR. These are:

1. *Knowledge Acquisition.* A common problem in expert systems is how to provide the rules on which the system depends. In complicated domains, rules may be difficult to create and the number of rules required may be unmanageably large, with no guarantee that the rules are reliable. However some of these complicated domains may naturally suit a CBR procedure for solving problems. But in domains not naturally suited to CBR, due to unavailability of cases, or cases may be in a hard to use format (e.g. cases described with natural text) after the initial case engineering effort it is often simple to add new cases to the case base and to maintain the case base.
2. *Knowledge Maintenance.* In most KBSs the domain is not fully understood when the initial knowledge base is set up. Very often in KBSs when new knowledge is discovered the system needs to be retrained on this new data. This process generally requires a knowledge engineer. Alternatively with CBR a user may be able to add new cases to the case-base without expert intervention. Also, because of the incremental learning performed by a CBR system, they can often be deployed with only a limited number of important cases or "seed" cases. New cases can be added to the case-base if and when the case library is insufficient in practice (Mark et al, 1996). This cannot be achieved with other KBSs as they should take into account all possible theoretical problems before deployment.
3. *Increasing Problem-Solving Efficiency.* Reusing prior retained solutions helps a CBR System to increase its problem solving efficiency. This is achieved by saving failed solutions as well as successful solutions, to warn of future potential problems.
4. *Increasing Quality of Solutions.* In a complicated domain where the theory is not well understood, it is difficult to build rules and the rules developed may be imperfect. However cases are real world solutions to problems, so they

generally have a higher quality solution. Also due to the ability of CBR systems to expand its case-base, over time solutions will become more accurate.

5. *User Acceptance.* As already mentioned a big problem with KBSs is for user acceptance. There is no benefit of having an elaborate KBS if users will not use it. This is a big problem with other systems: neural network systems generally cannot provide explanations to support their decisions, and rule-based systems must explain their decisions by using rules that the user may not fully understand or accept (Riesbeck, 1988). However with a CBR system an explanation can be given by simply presenting an actual prior case to the user as support of the systems prediction.

2.2 Explanation in Expert Systems

As already mentioned, there is a significant number of KBSs installed in companies but not being used. There are numerous arguments put forward that try to explain the lack of use of KBSs. One simple reason for the lack of usage could be a fear by the end user of their personal job security. That if the system is successful, the company would not need as many workers. However most KBSs are intended as assistants to professional people and not at human replacing (Berry and Hart, 1990).

The introduction of a new KBS into an organisation generally results in modification of work practices, which is generally met with resistance from the workers. There are often also problems with interaction of the KBS with classical software in the organisation itself and also that KBSs are often imposed on the people who know little about the technology (Majchrzak and Gasser, 1991). To help reduce these problems it is important to involve the end user in the entire design process of the KBS. Not only does this result in a system that works to the liking of the end user and should integrate more smoothly with existing software and work procedures, but it should also create a sense of owning for the end user hopefully reducing anxiety about their job security.

It is also important in a KBS that the user understands the models that are used by the system and is confident in the system outputs (Brézillon and Pomerol, 1996). A common approach to attempt to increase confidence in a system's output is to include an explanation of the result in the system's output (Moore and Swartout, 1988). Explanations have been noted as an important feature of KBSs that are often not offered with other types of computer systems (Richards, 2003). They help to justify predictions in complicated domains where the domain is not fully understood or complicated heuristics are used (Lee and Compton, 1995; Swartout and Moore, 1993).

However many initial KBSs that used explanations were not very efficient at convincing users. Explanations were often only from the system to the user, without the user intervening in the building of the explanation (Karsenty and Brézillon, 1995; Cawsey, 1993; McSherry, 2001) and very often explanations were rejected as the user could not understand the explanation.

Producing more understandable explanations is a more difficult task than it may appear. The main complication is that different people understand different types of explanations better than others. One common technique for simplifying the type of explanation to give different people is by creating one type of explanation for experts and another for novices. Some research has shown that experts tend to prefer low-level explanations, while novices prefer a combination of higher-level explanations, background information and low level explanations (Ramberg, 1996; Wolverton, 1995). Research has also soon that novice users, due to their lack of domain knowledge, have a greater need for explanations to understand KBS outputs than experts, they generally use explanations for understanding and prefer a justification (why) style of explanation. Alternatively, domain experts tend to use explanations in anomalous situations, as they are more likely to recognise anomalies than novice users. Expert users tend to use a reasoning-trace (how) style of explanation more than novice users (Mao and Benbasat, 2000).

The development of user models for customised explanations can be a time-consuming process. Swartout and Moore (1993) describe an approach using stereotypes. Using stereotypes the level of detail is customised to each stereotype. Rules have different levels associated with them and are fired if the user belongs to that level. Although this system is easier to implement than user models, there is a problem that if the user does not fit the stereotype the explanation will be unsatisfactory.

Studies have shown that the use of explanations generally improves a user's acceptance of a KBS (Ye, 1995). However Cawsey (1993) makes the point that explanations wrongly tailored for different users are probably worse than providing no explanation. But what makes a good explanation? The next subsection discusses some aspects of a good explanation.

2.3 Improving Explanations

In order to improve the quality of explanations given by a KBS, the properties of good explanations must be considered. Stephan Toulmin, a philosopher developed a generic approach to providing an argument. His six-element argument structure (Figure 2) can be used in a broad range of situations. (Toulmin, 1958; Toulmin et al., 1984; Ehninger and Brockriede, 1978). The six elements of Toulmin's argument structure as given by Shanker (1999) is:

1. *Data*: The particular facts about a situation on which a claim is made.
2. *Warrant*: The knowledge that justifies a claim made using the data.
3. *Backing*: The general body of information or experience that validate the warrant.
4. *Qualifier*: The phrase that shows the confidence with which the claim is supported to be true.
5. *Rebuttal*: The anomaly that shows the claim not to be true.
6. *Claim*: The assertion or conclusion put forward for general acceptance.

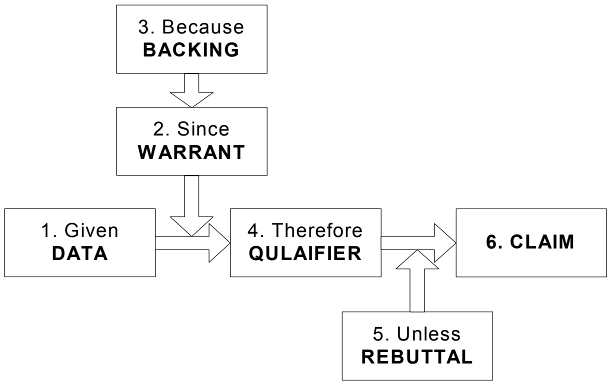


Figure 2: Toulmin's Argument Structure (Ehninger and Brockriede, 1960)

Toulmin's model of argument is significant in that it highlights the discrete response steps that a KBS explanation should follow, in order to answer user queries in a convincing way. Wick (1992) points out how research in explanation has, without stated intent, evolved to engulf Toulmin's argument structure.

Although Toulmin's model provides a method for generating good arguments, other factors need to be considered when developing explanations in a KBS. Swartout and Moore (1993) discuss five aspects of a good explanation in a KBS

1. Fidelity – how accurate is the representation? Fidelity is aided by a simple inference engine.
2. Understandability – are the content and context understandable? The components to be considered are terminology, user sensitivity, abstraction into different levels, summarization (the amount of detail offered in an explanation), the possible system perspectives, linguistic competence and feedback.
3. Sufficiency - is there enough knowledge to provide explanation in different contexts? This will require knowledge to be stored that allows explanation about the system's behaviour, justification, preferences, domain explanations and terminology definitions.
4. Low construction overhead – how time consuming and difficult is it to build explanations?
5. Efficiency - what will the system's response time be like? Will it take a long time to generate an explanation?

3. Explanation in existing expert systems

There are numerous existing KBSs that use explanations to help increase credibility in their predictions. This section examines some approaches to help improve the explanations given by KBSs. The first approach is viewpoints (Finch, 1999). Viewpoints help to change the type of explanation given to different users. The next approach is the use of Satisficing Conclusion-Substantiating Explanation (Wolverton, 1995), which builds up explanations so that they are as detailed as possible, but still understandable by both novice and expert users. Some expert systems based on neural networks or simulations can prove extremely difficult for creating explanations. The final approach in this section looks at adding explanations to KBSs that use simulation as their method of making predictions (Greer et al., 1995).

3.1 Viewpoints

As discussed by Finch (1993), a viewpoint is a way of looking at an underlying concept, allowing it to be presented differently for different users. This allows one expert system to be created, which can provide explanation for different classes of user, by using different viewpoints on the knowledge base. This is an important feature as it gets over the problem discussed in Section 2, where different users prefer different types of explanations and varying levels of detail in explanations. A simplistic approach to viewpoints is to consider them as a filter that can be passed over the knowledge, only allowing certain facts and concepts to be seen through it and holding back the facts and concepts that the user may not understand or are "insignificant" in explaining the conclusion that is found. Perhaps the simplest way to describe viewpoints is with an example taken from Finch (1999).

This example uses a rule-based system called IMVEX that is coded as a suite of Perl modules, and uses a backward-chaining inference engine to achieve goals based on facts which the user (or some other external agent) provides and a rule base. The rules are expressed in the following form:

<label>: <conjunction of conditions> => <consequences>.

For example, the following rule (label is '4') states that any man, under the age of 65, who is healthy, is not entitled to parish relief:

4: age < 65 & man & healthy => relief' no.

Imvex also allows text to be associated with symbols and their values. This is used when the rule is displayed to the user, for example, the above rule could become:

Rule 4 states that:

[1] If you are under the age of 65
[2] AND you are a man
[3] AND you are healthy
[4] THEN you will not be granted parish aid.

Now consider two participants in an application for parish relief, the claimant and the claim adjudicator. The claimant makes an application for parish relief and the adjudicator decides whether the claim is valid. Viewpoints allow both participants to use the same rule base, but to interact with it and get information in a way suited to their needs.

The simplest way that IMVEX displays the information according to the viewpoint used, is the presentation of the information. For example a result from firing a rule that has a conclusion "relief = yes", then this might be presented to the claimant as: *"You will receive parish relief"*. While the adjudicator would see: *"The claimant will receive parish relief"*.

Although this is a useful facility, it is only a minor aspect of the viewpoints mechanism, affecting only the superficial presentation of information. The major advantage of the viewpoints mechanism is the ability to associate viewpoints with parts of the rule base, allowing alternative chains of reasoning to be formed, depending on the user. Consider the following rule, which states that someone who is unhealthy should be granted parish relief:

9: healthy' no => relief.

For the claimant, this is sufficient. They will be asked whether they are healthy. If they say 'no', they will be told they deserve parish relief. Alternatively the adjudicator would need some proof of the claimant's ill health. Possible proof could be production of a valid doctor's certificate. Therefore the above rule would be sufficient for the claimant, but the adjudicator's viewpoint will have to be able to incorporate the need for a doctor's certificate. So rule 9 would become for the two viewpoints:

9/claimant: healthy' no => relief
9/adjudicator: healthy' no & doctor_cert => relief

So when the adjudicator is using the system, he will be asked about the patient's health and also about the production of a valid certificate. The claimant on the other hand will only be questioned about his health. Thus, by using viewpoints, a single rule base can be used by both a benefit claimant and the claim adjudicator. This has obvious advantages in the maintenance and development of the rule base, since there is only one rule base to maintain.

3.2 Satisficing Conclusion-Substantiating Explanation

Wolverton (1995) presents the notion of Satisficing Conclusion-Substantiating (SCS) Explanations, which aims to have explanations as understandable as possible, but maintaining a high level of belief. This is achieved by keeping the explanations free of:

1. Any portion of an explanation that the user cannot understand, as this is presumed not to add to their level of belief in the conclusion, and could possibly reduce the users level of belief.
2. Portions of an explanation that add very little to the user's belief in the conclusion, provided they are not required to push the user's belief in the conclusion over the user's belief threshold.

This approach requires the system to have a user model. A definition of SCS Explanations is given in Definition 1.

Definition 1: Satisficing Conclusion-Substantiating Explanation (Wolverton, 1995)

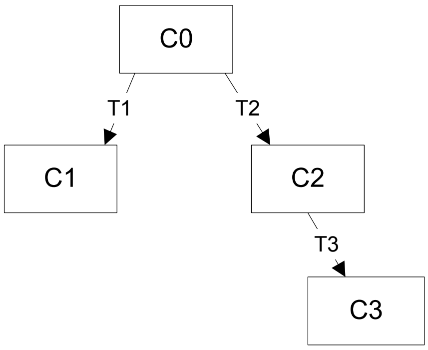
Given a system S, some conclusion produced by that system C_S, and a user U with domain theory τ_U, an SCS explanation of Cs to U with belief threshold p, E (C_S, U, p), is the minimal set of knowledge elements such that either

1) *Prob $[C_S | \tau_U \cup E(C_S, U, p)] > p$, or if no explanation meeting this criterion exist,*
2) *There is no other explanation E' $(Cs | \tau U)$ such that*
$Prob\ [C_S | \tau_U \cup E'(C_S, U, p)] > Prob\ [C_S | \tau_U \cup E'(C_S, U, p)]$

Here, *Prob* need not represent a strict Bayesian probability, but can also correspond to any numeric method of representing likelihood or degree of belief.

This definition says that a SCS explanation is the shortest sequence of facts that can convince the user that the conclusion is true. Or if no sequence of facts can give the user a confidence greater than *p*, then the set of facts that will give the user the greatest confidence in C_S is used as an explanation.

We will now examine an example of explanations from a SCS explanation system, taken from Wolverton (1995). Figure 3 shows the reasoning sequence of the system for a diagnosis of a problem with a digital signal processing system.



C0: Signal output of system is faulty
T1: Verify values stored in hardware registers
C1: No problem in initialization software
T2: Oscilloscope signal level check of gain boards
C2: Analogue gain board faulty
T3: Observe time response of T&C signals
C3: Timing component should be replaced
Prob (C3|T1 ^ T2 ^ T 3) = .95
Prob (C3|T2 ^ T 3) = .92
Prob (C3|T3) = .75
Prob (C3|T1 ^ T 2) = .7
Prob (C3|T2) = .5

Figure 3: Example reasoning of digital signal processing expert system (Wolverton, 1995)

In this example, the system is attempting to diagnose the problem (C0) that the observed output signal of the system is too weak. The system firsts tests the hypothesis that the problem is caused by a fault in the initialization software by verifying the values stored in the hardware registers (T1). This test indicates that the software is initializing the registers correctly (C1), so the system can give up on this branch of the hierarchy. Next, the system tests the hypothesis that there is a hardware fault by performing an oscilloscope signal level check of the two (analogue and digital) gain boards (T2). This test reveals that the analogue gain board is faulty, while the digital gain board is fine (C2). Finally, the system attempts to refine its current diagnosis by observing the time response of the analogue gain board's timing and control signals (T3). This test reveals that the board's timing component is faulty (C3), and this is returned as the expert system's diagnosis.

Outputting the system's entire line of reasoning as an explanation for the conclusion C3, without using a SCS Explanation, might produce the following explanation:

> *I verified the values stored in the hardware registers. The result was that all hardware register values were correct. This test indicated that the hardware initialization software is working correctly.*
>
> *Next, I did an oscilloscope signal level check of the gain components of the hardware. The result was that the signal level on the analogue gain stage was too weak, and the signal level on the digital gain stage was fine. This test indicated that the analogue gain board is faulty, and the digital gain board is fine.*
>
> *Next, I observed the time response of on-board timing and control signals of the analogue gain board. The result was that the rising edge of the START timing signal occurs after the raw signal event*

of interest. This test indicated that the hardware-timing component of the analogue gain board is faulty and should be replaced.

This explanation is an accurate representation of the tests that were carried out on the system and the conclusions reached in the process of producing the final diagnosis, but it also is probably unnecessarily long and detailed for most users. And this is only a simple example of a small portion of a fault hierarchy; explanations of realistic complex diagnoses can quickly grow to be completely unmanageable.

Now let's examine two different SCS explanations of the same diagnosis. The first explanation will be generated for an expert user, he is assumed to know about all the concepts and all the tests that the system knows about, and therefore this user's model is identical to the system's knowledge base. Assume that the belief threshold for an explanation is set to 0.9. For this user and this belief threshold, the results of test T1 constitute insignificant information. Knowing the results of tests T2 and T3 already support a .92 level of belief in conclusion C3, and additionally knowing the result of test T1 raises that value only a very small amount to .95. It was necessary to perform that test during the diagnosis in order to eliminate software as the potential source of the problem and to focus on the hardware, but in retrospect, the result of that test adds very little to the expert user's belief in the truth of C3. So an SCS explanation for this user would remove the first paragraph of the previous explanation.

Now consider a user with less expertise. This user has an understanding of the higher-level components of the system, but doesn't understand the detailed workings of those components. In particular, he does not know about the timing and control components of the gain boards. Thus, his user model includes tests T1 and T2, but does not include T3. Therefore, the SCS explanation method will not present the detailed description and results of test T3. It will present the results of T1 and T2, rather than T2 alone, because (1) both tests taken together give more confidence in the conclusion than T2 by itself, and (2) since the confidence level for T1 and T2 (.75) is less than the belief threshold (.9), the method wants to find the explanation that maximizes confidence. So the explanation produced is the same as the first one presented in this section, except with a simplified final paragraph, stating the findings of T3 but not the actual test itself:

I verified the values stored in the hardware registers. The result was that all hardware register values were correct. This test indicated that the hardware initialization software is working correctly.

Next, I did an oscilloscope signal level check of the gain components of the hardware. The result was that the signal level on the analogue gain stage was too weak, and the signal level on the digital gain stage was fine. This test indicated that the analogue gain board is faulty, and the digital gain board is fine.

A more detailed test of the analogue gain board indicated that the hardware-timing component of the analogue gain board is faulty and should be replaced.

3.3 Simulation Explanation

With some KBSs, such as simulations and neural networks, it is often very difficult to produce explanations to support predictions. This is due to many of these systems being treated as "black boxes" (Beck et al, 1989; Bernatzki, 1996), with outputs being produced without any explanation of how the output was derived. Greer et al (1993) examine the use of simulation models in the agricultural sector for making decisions for farmers. In the agricultural community there exists a general distrust in KBSs. In order to overcome initial distrust, a system must give plausible explanations for any decisions it makes, and must use terminology and logic understood and trusted by the user. Greer et al. (1995) approach the "black box" problem of simulations, by developing a system that creates explanations that are suitable for the agricultural community.

The architecture of the system (which is referred to as EXPLAIN) consists of a User Interface, a Simulation Model, an Explanation Model and a User Model as shown in Figure 4. The User interface has three responsibilities: querying the user for suitable inputs to the simulation, acquiring knowledge about the current user, and outputting the results and explanation of the simulation to the User. The Simulation Model carries out the simulation on the users input, and outputs the results to the Explanation Module. Using the input from the Simulation Model along with the input from the User Model, which contains information about the user, the Explanation Module creates an individualised explanation of the Simulation. This explanation is then passed back to the user via the Interface.

The Explanation Module creates this individualised explanation by first creating an internal representation of the simulation. Then using the user model it personalises the representation, which is then passed through a Natural Language Generator before returning the result to the User.

Using the Natural Language Generator and the User Model to create the explanation of the simulation creates an explanation that is both in an easy to read language, and at a standard that is understood by the user. This should lead to an overall increase in the level of belief in the prediction for the user.

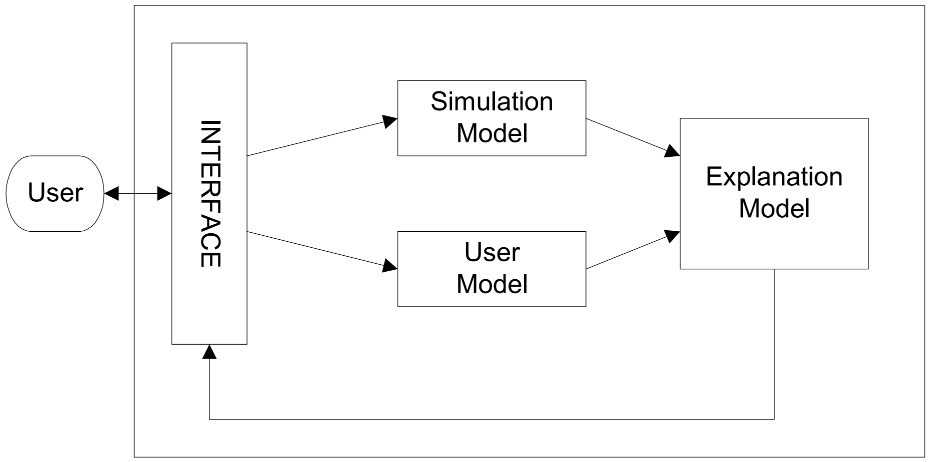


Figure 4: Architecture of the Explain System (Greer et al. 1995)

4 Explanation in Case-Based Reasoning

CBR systems offer a very different approach to constructing explanations than in other types of expert systems. The input data to a case-based system are not converted to rules as in a rule based system, or into a complicated interconnected network of perceptrons as in neural networks, but they are kept in their original form, just as they were entered. The benefit of maintaining the cases in their original form (as pointed out by Riesbeck (1988)) is that “a solution to a new problem can be explained simply by presenting the case that was adapted to create the solution”. Unless the adaptation process is complicated, the differences between the solution and the problem case should be minimal, so the user can easily judge the validity of the solution.

Research work on Case-Based Explanation (CBE) can generally be categorised into knowledge-light and knowledge-intensive approaches (Cunningham et al, 2003). A knowledge-intensive approach to CBE will incorporate mechanisms such as rule-based or model-based inference that can be used to generate explanations. Amongst the earliest examples of this approach is the work of SWALE (Kass and Leake, 1988) and its descendents; more recently is Model-based Case Adaptation System (MoCAS) (Pews and Wess, 1993), the Plan Abstraction and Refinement in an Integrated System (PARIS-System) (Bergmann et al., 1993), and finally DIRAS system for assessing the long-term risks of Diabetes (Armengol et al., 2000).

On the other hand the knowledge-light approach is usually used in retrieval only systems or interactive adaptation. A good example of the knowledge-light approach to CBE is the CARES system for predicting recurrence of colorectal cancer (Ong et al., 1997). CBR has been used successfully in developing diagnosis systems based on a feature-value representation. In CBR systems that use a simple feature-value based representation, the retrieved cases can be used in explanation as follows (Cunningham, et al. 2003):

> *“The system predicts that the outcome will be* X *because that was the outcome in case* C1 *that differed from the current case only in the value of feature* F *which was* f2 *instead of* f1.
>
> *In addition the outcome in C2 was also* X *…”*

Explanation in these terms (i.e. expressed in the vocabulary of the case features) will not always be adequate but in some situations, such as in medical decision support, it can be quite useful.

4.1 MoCAS

MoCAS (Pews and Wess, 1993) uses explanation-based similarity for case retrieval and adaptation. In explanation-based similarity, the similarity of two cases is based on the similarity of their explanations. To achieve this, the domain knowledge is modelled on several layers of abstraction. This modelling must be able to switch between different levels by transforming representational terms from one level to another. An explanation on a lower level of abstraction is more detailed and contains more rules than an explanation on a higher level of abstraction. Therefore the explanations of two cases can differ very much on a low level of abstraction, but may be identical on a higher level of abstraction. The lower the level of abstraction on which two explanations are identical, the higher the assessment of their similarity.

4.1.a Single-level explanations

An explanation on an isolated level of abstraction can be represented in a graph structure with two different kinds of labelled nodes: *rule-nodes* and *fact-nodes* as in Figure 5. Two single-level explanations are called identical if the graphs are identical except for the labelling of the fact-nodes, which can vary, however the labelling of the rule nodes must be the same, but the instantiations of the rules can be different.

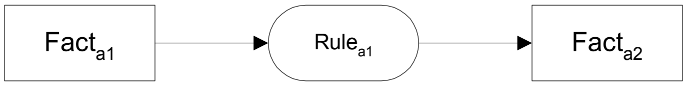


Figure 5: Single-level Explanations (Pews 1993)

4.1.b Multi-level explanations

On two adjacent levels, the explanations are linked by two different kinds of abstraction mappings. The fact abstraction relates several level-*n* fact nodes on level *n*+1 (Figure 6). It is assumed that the available domain knowledge includes information about the different possibilities for fact abstraction. Rule abstraction occurs when a sub graph containing several rule- and fact-nodes can be mapped onto a single more abstract rule-node on the next higher level of abstraction.

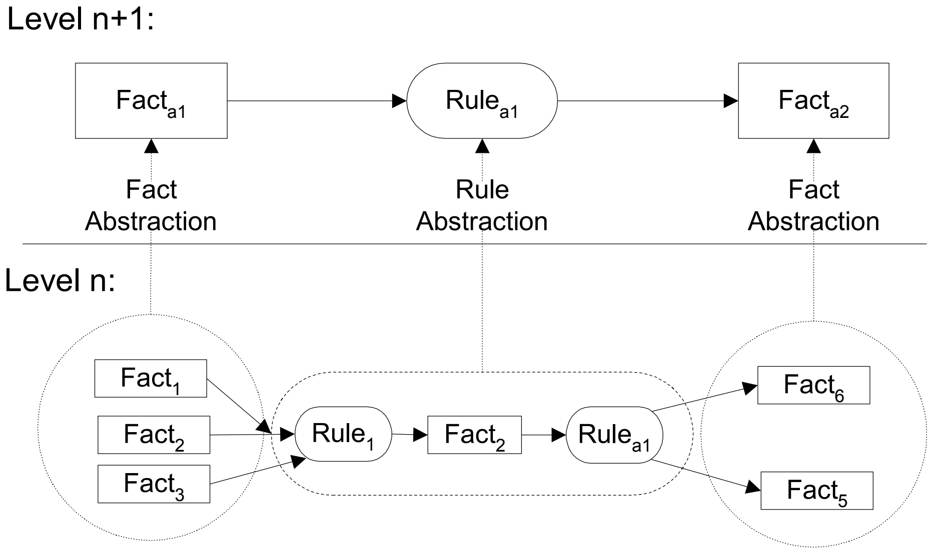


Figure 6: Multi-level Abstractions (Pews and Wess, 1993)

4.1.c Similarity between a complete case and a problem

To compare the similarity between a complete case and a problem case an attempt is made to match the explanations of the two cases at the highest level of abstraction. If this matching is successful, the process moves to the next lower level of abstraction and attempts to match the explanations again. The lowest level of abstraction at which the explanations can be mapped, indicates the degree of similarity between the case and the current problem.

4.1.d Example case

Consider a simple example of a diagnosis task from a technical domain as shown in Figure 7. In Case$_1$ a generator G_1 supplies a light bulb L_1 via Wire$_{18}$ and Relay$_7$ with a voltage. Suppose Wire$_{18}$ is broken, therefore the lamp stays dark even if the generator provides a voltage and the relay is closed. Then in $Case_2$ instead of a light bulb there is a motor M_1. In this case assuming that $Wire_{65}$ is broken the motor will not turn even if the relay is closed and G_2 is providing a current. The problem in this case is the broken wire. Therefore the explanation would be a diagnosis of the unintended behaviour, here it would be the rule for the broken wire.

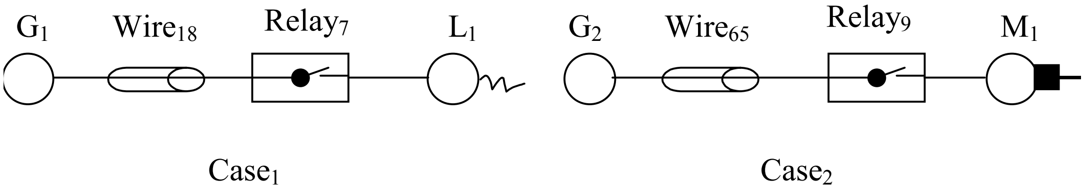


Figure 7: Two cases from a technical diagnostic domain (Pews and Wess, 1993)

For these two cases, the modelling leads to explanation graphs as shown in Figure 8. The explanations of $Case_1$ and $Case_2$ are not identical on the lowest level of abstraction, because different rules describe the behaviour of the motor and the bulb. But at a higher level of abstraction (level 2), the behaviour of the two different components can be abstracted into the rule that reflects ”doesn’t operate” behaviour. On this level of abstraction the two cases are identical.

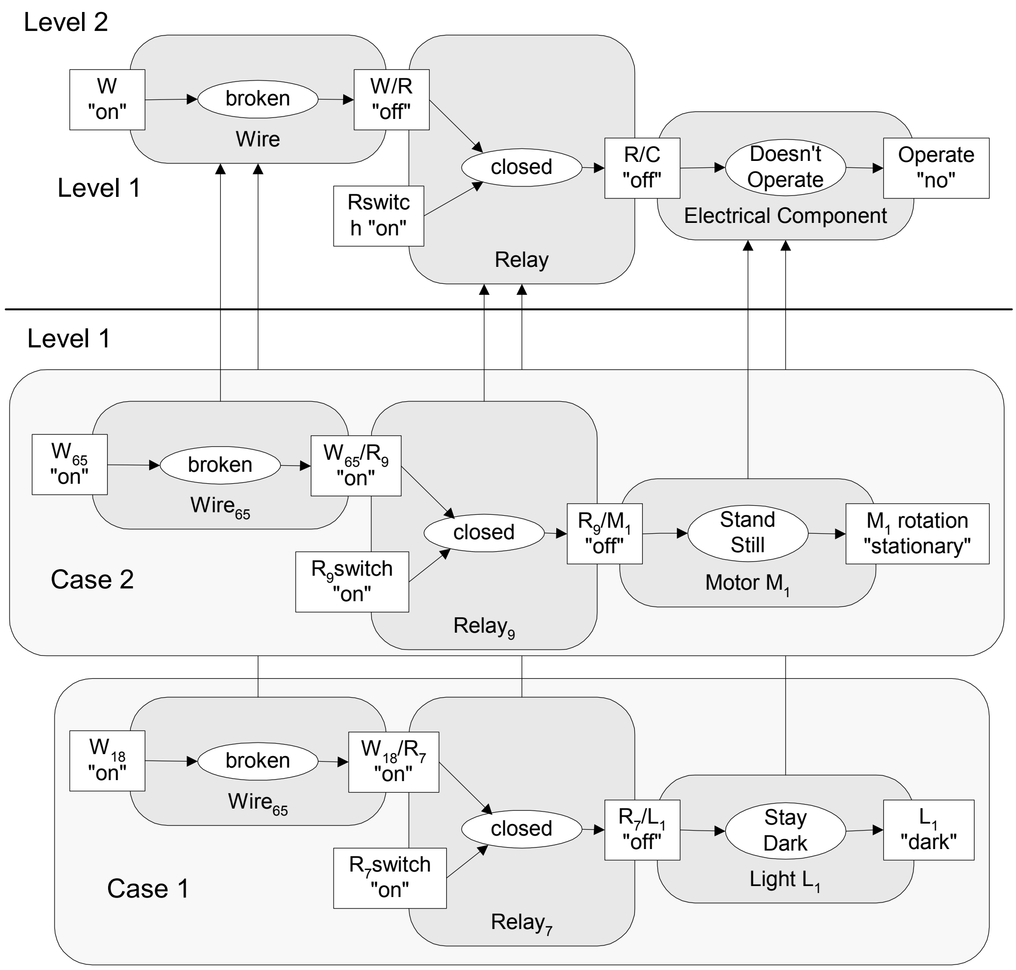


Figure 8: Explanation structures for two diagnostic cases (Pews and Wess, 1993)

The next step is to refine these abstract solutions towards a full solution to the original problem at the required level of detail. A mapping of $Case_1$ to the problem of $Case_2$ (generator works, relay closed, but the bulb is still dark) a mapping of the faulty component $Wire_{18}$ from $Case_1$ to the related component in $Case_2$, $Wire_{65}$, indicates that $Wire_{65}$ is faulty.

However, in general, a mapped abstract solution needs to be refined towards a single abstract rule. This refinement can be achieved by standard hierarchical search-based methods, which can employ exactly the same knowledge that was already utilised during similarity assessment. The abstract solution that is already available imposes strong constraints on the search space, so that only a small amount of sub-problems have to be solved. However if the degree of similarity between the two cases is too low, the search space can become so large, that it is not possible to achieve a solution in a reasonable amount of time. This could be a strong indication that the new case has to be added to the case base.

For example, consider a new case, similar to $Case_1$ only that instead of a wire, there is a more complex component. For example this component could be an infrared-sender and receiver including several amplifiers. An explanation mapping will now come up with the mapping of the wire to the compound component. However, no decision can be made as to what part of the component is faulty. But the required refinement search can now be focused on the respective sub-components. Model-based diagnosis techniques using the model knowledge of the relevant components and their interaction can be used to determine a potential refined diagnosis.

4.2 DIRAS

Diabetes is one of the most common diseases in the world. At present it affects approximately 3% of the European population. Although diabetes is manageable if certain lifestyle changes are made, a bad management of the disease will produce microcomplications (blindness, renal failure or polyneuropathy), and macrocomplications (gangrene and amputation, aggravated coronary heart disease or stroke). The aim of DIRAS is the prediction of individualised risk assessment for a patient with diabetes of developing one of these complications (Armengol et al., 2000).

DIRAS uses Lazy Induction of Descriptions (LID) (Armengol and Plaza, 2001) to infer the risk relating to each particular type of complication. LID works by selecting the most discriminatory feature (using the López de Mántaras (RLM) distance). This feature value is added to the description. From this point only cases with the same feature value are used. The system checks to see if all the remaining cases have the same risk level for the complication being investigated. If not - the next most discriminatory feature of the remaining cases is selected. This process continues until the set of remaining cases have the same risk level. The description used for explaining the classification contains all the most discriminatory features used during the process and there values (Armengol and Plaza, 2001).

DIRAS also produces a report that can be useful for non-expert physicians to be able to manage diabetic patients. This report is made up of 4 sections:

1. Personal data (e.g. age, diabetes type, year of diabetes diagnostic).
2. Assessments about the measures (e.g. cholesterol level).
3. Information about macrocomplications.
4. Information about microcomplications.

An example output of the risk of macrocomplications is given in Figure 9.

3- MACROVASCULAR COMPLICATIONS

Ischemic heart : No
 Infarct (or coronary bypass or angioplasty): No
 Anginal chest pain : No
Stroke : No
Low extremities vasculopathy : Yes
 Amputation above ankle : No
 Amputation below ankle : No

	Right foot	Left foot
Leg claudication : No		
Bypass or angioplasty :	No	No
Feet pulse present :	No	No
Healed ulcer :	No	No
Acute ulcer :	No	No

Global risk progression: HIGH because the value of total cholesterol is high

---

Risk for Specific Macrocomplications :

 Stroke : VERY-HIGH because the blood pressure is very high and the patient has macrocomplications
 Infarct : HIGH because the value of total cholesterol is high
 Lesion/amp. : HIGH, because the patient has polyneuropathy with normal sensitivity and vasculopathy

Figure 9: Sample report showing the possible risk of macrocomplications for a patient using DIRAS (Armengol et al., 2000)

4.3 CARES

MOCAS and DIRAS are both examples of knowledge-intensive systems. However some people argue that “knowledge intensive CBR is missing the point of CBR, which is the potential CBR has to finesse knowledge engineering effort by manipulating cases that are compiled chunks of knowledge” (Cunningham et al., 2003). The National University of Singapore in conjunction with Singapore General Hospital developed the CARES system, to be able to predict the risk of recurrence of colorectal cancer, and suggest an appropriate follow up regime for the patient. Without this prediction, a patient may be encountering unnecessary, painful and expensive check ups. An accurate prediction system would reduce these.

CARES uses a matching process known as nearest-neighbour matching. On entry of a new case the system looks at all the existing cases and retrieves the most similar cases. Given a new case x and a retrieved case *q*, for each feature *f*, the system computes a degree of match between the feature values using a similarity function. An example of a similarity function is given in (1).

$$sim(q_f, x_f) = \begin{cases} 0, & \text{if } f \text{ discrete and } q_f \neq x_f \\ 1, & \text{if } f \text{ discrete and } q_f = x_f \\ 1 - \dfrac{|q_f - x_f| - f_{\min}}{f_{\max} - f_{\min}}, & \text{if } f \text{ continuous} \end{cases} \qquad (1)$$

This is then repeated for all the features (*F*), which the user has selected to use. Each feature has an associated weight, which represents the overall importance of that particular feature in calculating the similarity between the two cases. The overall similarity between the two cases is calculated using (2).

$$Sim(q, f) = \frac{\sum_{f \in F} w_f \times sim(q_f, x_f)}{\sum_{f \in F} w_f} \qquad (2)$$

During the retrieval process the system maintains a list of the ten most similar cases in the case base. These cases are then displayed to the doctor for browsing. From these most similar cases, the system also produces its prediction and a follow-up regimen and explains its decisions using the retrieved case as a basis.

It should also be noted that the doctors could select and adjust which feature's to include when doing the similarity check and also what weight to give the feature. This allows the system to base its decisions on similar criteria as the doctor, producing more useable findings for the doctor. In the CARES system all 10 retrieved cases need not have the same outcome as the conclusion presented by the system. More than likely they won't, however by being able to browse the retrieved cases the doctor is able to consider any anomalies that are present in the 10 cases.

5. Usefulness of Explanations

The previous sections have discussed the use of explanations in KBSs. But how useful are these explanations? Are users just as convinced with only a prediction? Do they really need an explanation? And if they do, what type of explanations do they like best? Do they prefer Case Based Explanations (CBEs), due to its consistency with much that psychologists have observed in the natural problem solving that people do, as expressed in (Kolodner, 1991) or is a Rule- Based Explanation (RBE) more acceptable?

In (Cunningham et al., 2003) we investigate the usefulness of explanations in both CBR and Rule Based Reasoning (RBR). Our paper uses the domain of Blood Alcohol Levels (BAL) where a system was set up which could predict a persons BAL by using such features as amount drank, duration of drinking, amount to eat, gender and weight. In the experiment 37 subjects were presented with eight unique problem cases. Each of the problem cases was presented three times, once with prediction and case-based explanation, once with prediction and rule-based explanation and once with predictions only with no explanation. The rule-based and case-based explanations were presented together but the order was varied to avoid any bias due to familiarity. The subjects were asked to score how convinced they were by the explanations on a 5-point scale (No, Maybe No, Maybe, Maybe Yes, Yes). In the evaluation of the results these scores were interpreted as numeric values 1-5.

An incorrect prediction coupled with a poor explanation was included in each category to help assess the attention paid by the subjects to the evaluation. The average rating for these poor predictions was 1.5 while the average for the other predictions was 3.9 (on a scale of 1-5). The ratings for these poor predictions were not considered further in the evaluation. Figure 10 shows the averages of the remaining ratings.

Two things to note are the strong performance of the CBE and the fact that the predictions without explanation were still found to be quite convincing. Statistical tests were run on the data and a paired t-test showed that the CBE was better than the RBE (P value = 0.0005) and better than "No Explanation" (P value = 0.005).

Counting the wins and draws between the rule-based and case-based alternatives showed that CBE wins 105 times, RBE wins 48 times, and there are 106 draws.

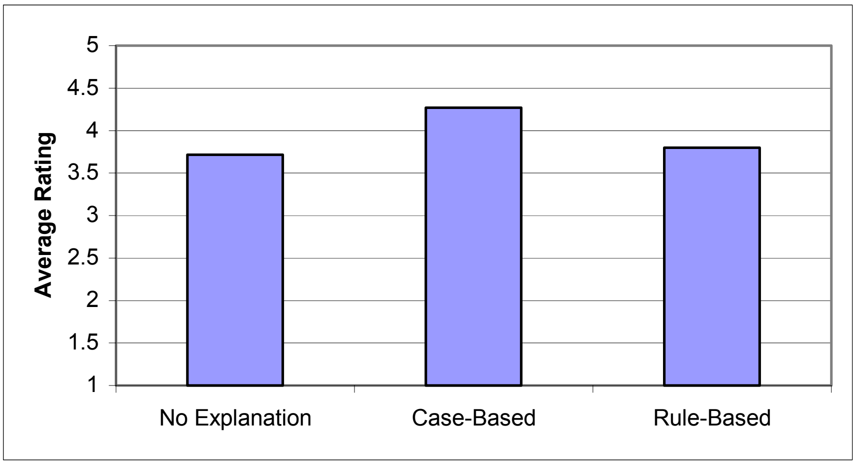


Figure 10: The average ratings of three alternative prediction and explanation systems

Even though the evaluation favoured the case-based approach, that does not mean that this approach will always be the best. It must be remembered that although case based explanation did best, it was still only slightly better than rule-based explanations, which itself was only marginally better than no explanation. As stated in Cunningham et al, (2003) it must be noted that:

- Because of the inherit instability of decision tree building algorithms there are alternative decision trees that would have produced different rules that might have scored better.
- Case- Based Explanation may inherently suit the task because it considers all features in the decision making process. The Rule-Based Explanation only considers a subset of features and this may be more acceptable in other domains.
- The comparatively simple case representation may favour CBE. It might fare less well with more complex cases (i.e. more features)
- Results may be different in domains where the subjects have more insight or less insight into underlying mechanisms.

6. Conclusions

In this review paper, we have shown that although it is often difficult to generate good explanations that by including them as part of a KBSs output, it can help improve users credibility in the system. Benefits of using CBR as KBSs that produce explanations have been examined. Such benefits of CBR include easier knowledge maintenance, easier knowledge acquisition and easier acceptance of CBE's, as their explanations are made up of actual real life cases.

The generation of explanations in some CBR systems is discussed. As not all domains are naturally suited to CBR, techniques for improving explanations in other types of KBSs are analysed. Users preference of case based explanations compared to rule based explanations is also considered in the paper. Attention has also been given to the fact that different classes of users prefer different types of explanations.

Although it has been shown that use of explanations help increase users confidence in a KBS, a lot of work still needs to be done in the area of generating valuable explanations. More research needs to be carried out in the type of explanations different classes of users prefer, for example the level of detail provided by the KBS and the type of explanation provided. Possible work could focus on trying to make the generation of explanations more interactive with the user. Such interactivity could include the ability for users to easily switch between different types or levels of detail of explanations or systems that allow users to control features that they themselves feel are more discriminating. Another area for further work could include how making it easier for users to revise their inputs for certain features might increase their acceptance of the KBS as they can see the actual effects particular features have on a KBSs prediction.

References

Aamodt A., Plaza E., "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches", *AI Communications*, Vol. 7(i), pp39-59, 1994.

Armengol E., Palaudàries A., Plaza E., "Individual Prognosis of Diabetes Long-term Risks: A CBR Approach. Methods of Information in Medicine", *Special issue in prognostic models in Medicine*, Vol. 40, pp46-51, 2000.

Armengol E., Plaza E., "Lazy Induction of Descriptions for Relational Case-Based Learning", *Machine Learning: EMCL 2001 Lecture Notes in Artificial Intelligence 2167*, L. De Raedt and P. Flach (eds.), Springer-Verlag pp13-24, 2001.

Bandini, S., Bogni, D., Manzoni S., "Alarm Correlation in Traffic Monitoring and Control Systems: A Knowledge-Based Approach" *15th European Conference on Artificial Intelligence 2002. Prestigious Applications of Intelligent Systems (PAIS 2002)*, IOS Press, pp 638-642, 2002.

Bareiss R., Exemplar-Based Knowledge Acquisition: A Unified Approach to Concept Representation, Classification, and Learning, San Diego, 1989.

Beck H., Jones J., "Simulation and Artificial Intelligence", *Knowledge Engineering in Agriculture*, pp119-135, 1989.

Bergmann R., Pews G., Wilke W., "Explanation-based Similarity for Case Retrieval and Adaptation and its Application to Diagnosis and Planning Tasks", *Proceedings of the 1st European Workshop on Case-based Reasoning*, pp182-196, 1993.

Bernatzki A., Eppler W., Gemmeke H., "Interpretation of Neural Networks for Classification Tasks", *Proceedings of EUFIT 1996*, pp1420-1424, 1996.

Berry D., Hart A., Expert Systems: human issues .Cambridge, MIT Press, 1990.

Brézillon P., Pomerol J., "Misuse and nonuse of knowledge-based systems: The past expreiences revisited", *Implementing Systems for Supporting Management Decisions*, P. Humphreys, L. Bannon, A. McCosh, P. Migliarese and J. Pomerol (eds.), Chapman and Hall, pp44-60, 1996.

Cawsey A., "User modelling in interactive explanations", *Journal of User Modelling and User Adapted Interaction,* Vol. 3 (1), pp1-25, 1993.

Cheetham W., "Global Grade Selector: A Recommender System for Supporting the Sale of Plastic Resin", *Proceedings of the 5th International Conference on Case-Based Reasoning (ICCBR 2003)*, Springer, pp 96-106, 2003.

Cortés U., Rodríguez-Roda I., Sánchez-Marrè M., Comas J., Cortés C., Poch M., "DAI-DEPUR: an Environmental Decision Support System for the Control and Supervision of Municipal Waste Water

Treatment Plants" *15th European Conference on Artificial Intelligence 2002. Prestigious Applications of Intelligent Systems (PAIS 2002)*, IOS Press, pp 603-607, 2002.

Cunningham P., Doyle D., Loughrey J., "An Evaluation of the Usefulness of Case-Based Explanation", *Proceedings of the 5th International Conference on Case-Based Reasoning (ICCBR 2003)*, Springer, pp122-130, 2003.

Cunningham P., Smyth B., Bonzano A., "An incremental retrieval mechanism for case-based electronic fault diagnosis", *Knowledge-Based Systems*, Vol.11, pp239-248, 1998.

Cunningham P., Bonzano A., Smyth B. "An incremental case retrieval mechanism for diagnosis", Technical Report TCD-CS-95-01, Department of Computer Science, Trinity College, 1995.

Durkin J., "Expert Systems : A view of the field", *Intelligent Systems and their Applications*, Vol. 11(2), pp56-63, 1996.

Druzdzel M.J., "Explanation in Probabilistic Systems: Is it Feasible? Will it Work?", Proceedings of the *Fifth International Workshop on Intelligent Information Systems (WIS-96)*, pp1-13, 1996.

Ehninger D., Brockriede W., Decision by debate (2nd ed.), New York: Harper and Row, 1978.

Finch, I., "Knowledge-Based Systems, Viewpoints and the World Wide Web", FLAIRS Conference, pp37-41, 1999.

Greer L.E., Falk S., Greer K.J., Bentham M.J., "Explaining and Justifying Recommendations in an Agriculture Decision Support System" *Computers and Electronics in Agriculture*, Vol. 11, 195-214, 1995.

Hammond K.J., "Case-Based Planning: Viewing planning as a memory task", *Proceedings of the DARPA Case-Based Reasoning Workshop,* Morgan Kaufmann, 1988.

Hayes-Roth F., and Jacobstein N. "The state of knowledge-based systems", *Communications of the ACM*, Vol. 37(3), pp27-39, 1994.

Leake D.B., "CBR in Context: The Present and Future", *Case-Based Reasoning: Experiences, Lessons and Future Directions*, D.B. Leake (ed), MIT Press, pp3-30, 1996.

Leake, D.B., "Towards a computer model of memory search strategy learning", *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, pp549-554, 1994.

Lee M., Compton P., "From heuristic knowledge to causal", *Explanations proceedings of eighth Australian joint conference on artificial intelligence* (AI'95), X. Yao (ed.), World Scientific, pp83-90, 1995.

Karsenty L., Brézillon P., Cooperative problem solving and explanation, *Int. J. on Expert Systems With Applications*, Vol. 8(4), pp445-462, 1995.

Kass A.M., Leake D.B., "Case-Based Reasoning Applied to Constructing Explanations", *Proceedings of 1988 Workshop on Case-Based Reasoning*, J. Kolodner (ed.), Morgan Kaufmann, pp190-208, 1988.

Kolodner J., "Improving human decision making through case-based decision aiding", *AI Magazine,* Vol. 12(2), pp52-68, 1991.

Kolodner J., "Reconstructive Memory, a computer model", *Cognitive Science*, Vol. 7 (2), pp281-328, 1983.

Kolodner J., Simpson R., "The MEDIATOR: Analysis of an early case-based program". *Cognitive Science*, Vol. 13(4):507-549, 1989.

Majchrzak A., Gasser L., “On using Artificial intelligence to integrate the design of organizational and process change in US manufacturing”, *AI and society*, Vol. 5, pp 321-338, 1991.

Mántaras R.L., “A distance-based attribute selection measure for decision tree induction”, *Machine Learning*, Vol. 6, pp81-92, 1991.

Mao J., Benbasat I., “The Use of Explanations in Knowledge-Based Systems: Cognitive Perspectives and a Process-Tracing Analysis”, *Journal of Management Information Systems,* Vol. 17(2), pp153-180, 2000

Mark W., Simoudis E., Hinkle D., “Case-Based Reasoning: Expectations and Results *Case-Based Reasoning: Experiences, Lessons and Future Directions*, D.B. Leake (ed), MIT Press, pp269-294, 1996.

McSherry D. “Interactive Case-Based Reasoning in Sequential Diagnosis”, *Applied Intelligence,* Vol. 14, pp 65-76, 2001.

Meyer M.H., Detore A., Siegel S.F., Curley K.F., “The strategic use of expert systems for risk management in the Insurance Industry”, *Expert Systems With Applications*, Vol. 5, pp15-24, 1992.

Mizoguchi R., Motoda H., “Expert systems research in Japan”*, IEEE Expert Magazine*, Vol. 10 (4), pp14-23, 1995.

Moore J.D., Swartout W.R., “Explanation in expert systems: a survey”, ISI Research Report, RR-88-228, Information Sciences Institute, University of Southern California, 1988.

Ong L.S., Sheperd B., Tong L.C., Seow-Choen F., Ho Y.H., Tong L.C., Ho Y.S, Tan K. “The Colorectal Cancer Recurrence Support (CARES) System”, *Artificial Intelligence in Medicine*, Vol. 11(3), pp 175-188, 1997.

Owens C., Indexing and retrieving abstract planning knowledge, Ph.D. Dissertation, Yale University, 1991

Pews G., Wess S., “Combining model-based approaches and case-based reasoning for similarity assessment and case adaptation in diagnostic applications”, *Preprints of the First European Workshop on Case-Based Reasoning (EWCBR-93)*, Vol. 2, pp325-328, 1993.

Price C.J., Pegler I., “Deciding Parameter Values with Case-Based Reasoning” *Progress In Case-Based Reasoning*, I. Watson (ed.), Springer-Verlag, pp121-133, 1995.

Ramberg R., “Constructing and Testing Explanations in a Complex Domain”, *Computers in Human Behaviour*, Elsevier Science Ltd, Vol. 12-1, pp 29-48, 1996.

Richards D., “Knowledge-Based System Explanation: The Ripple-Down Rules Alternative”, *Knowledge and Information Systems*, Vol. 5 (2), pp 2-25, 2003.

Riesbeck C., An Interface for Case-Based Knowledge Acquisition, 1988.

Riesbeck C., Schank R., Inside Case-Based Reasoning, Erlbaum, 1989.

Rowe G., Wright G., “Expert Systems in Insurance: A review and analysis”, *Intelligent Systems in Accounting, Finance and Management*, Vol. 2, pp129-145, 1993.

Schank R., “Explanation: A first pass”, *Experience, Memory, and Reasoning*, J. Kolodner and C. Riesbeck (eds), pp139-165, 1986.

Schank R., Dynamic Memory: A Theory of Learning in Computers and People, Cambridge University Press, 1982

Schank R., Abelson R., Scripts, Plans, Goals and Understanding, Erlbaum, 1977.

Shankar R., Musen M., “Justification of Automated Decision-Making: Medical Explanations as Medical Arguments”, *Proc AMIA Symposium*, supplement, pp395-399, 1999.

Shortliffe E.H., Computer-Based Medical Consultations: MYCIN, New York: Elsevier/North-Holland Inc., 1976.

Shortliffe E H., “Will Information Technology Dehumanize Health-Care Delivery?”, Proceedings of The American Philosophical Society, Vol. 137(3), pp390-398, 1992.

Stylianou A.C., Madey G.R, Smith R.D., “Selection criteria for expert systems shells: A socio-technical framework”, *Communications of the ACM,* Vol.35(10), pp30-48, 1992

Swartout W., Moore J., “Explanation in second generation ES”, *Second generation expert systems*, J. David, J. Krivine, R. Simmons (eds), Springer, Berlin, pp 543-585, 1993.

Swartout W., “XPLAIN: A system for creating and explaining expert consulting programs”, *Artificial Intelligence*, Vol. 21(3), pp285-325, 1983.

Toulmin S., The uses of argument, Cambridge University Press, 1958.

Toulmin S., Rieke R., Janik A., An introduction to reasoning (2nd ed.), Macmillan, 1984.

Watson I., Applying Case-Based Reasoning: techniques for enterprise systems, Morgan Kaufmann, 1997.

Watson I., “CBR is a methodology not a technology”, *Research & Development in Expert Systems XV*. R. Miles, M. Moulton, M. Bramer (eds.), Springer, London , pp213-223, 1998.

Wick M., “Expert system explanation in retrospect: a case study in the evaluation of expert system explanation”, *Journal of Systems and Software*, vol. 19(2), pp159-169, 1992.

Wolverton M., “Presenting Significant Information in Expert System Explanation”, *Proceedings of the 7th Portuguese conference on artificial Intelligence*, pp 435-438, 1995.

Ye L.R., “The Value of Explanation in Expert Systems for Auditing: An Experimental Investigation”, *Expert Systems with Applications*, Elsevier Science Ltd, Vol. 9(4), pp543-556, 1995.

Ye L.R., Johnson P.E., “The impact of explanation facilities on user acceptance of expert systems advice”, *MIS Quarterly,* Vol. 19(2), pp157-172, 1995.