Ripple down rules: possibilities and limitations

P.Compton, G.Edwards*, B.Kang, L.Lazarus*, R. Malor*, T.Menzies, P.Preston. A.Srinivasan, S.Sammut.

School of Computer Science and Engineering, University of New South Wales,
PO Box 1, Kensington NSW, Australia 2033,
email compton@spectrum.cs.unsw.oz.au
and
Department of Chemical Pathology, St Vincent’s Hospital
Darlinghurst NSW, Australia 2010

ABSTRACT

A major problem with building expert systems is that experts always communicate knowledge in a specific context. A knowledge acquisition methodology has been developed which restricts the use of knowledge to the context in which it was provided. This method, "ripple down rules" allows for extremely rapid and simple knowledge acquisition without the help of a knowledge engineer. An expert system based on this approach and built by experts is now in routine use. This paper reviews what has been achieved using the approach, its problems and potential.

A PHILOSOPHY OF KNOWLEDGE

Ripple down rules are based on a specific philosophical view of the nature of knowledge (Compton and Jansen 1990). We make no apology for this, for we would suggest that approaches to building knowledge based systems that eschew any philosophical consideration are in fact strongly dependent on and limited by such assumptions. The prevailing assumptions are implicitly Platonic, in that it is assumed knowledge has some concrete reality, it is some sort of stuff that can be "extracted" or "mined" from experts' heads. If you can't get enough of this knowledge stuff to make the expert system work properly it is because the expert can't properly report on what is in his head. An implicit corollary of this approach is that knowledge is in some sense right or correct. This follows from the belief that the limitations of a knowledge base are due to inadequate knowledge acquisition - the problems would disappear if only acquisition could acquire deep (true) expert knowledge. Clear cut evidence that this is not the right way to approach knowledge is provided by Shaw(Shaw) who has demonstrated that experts may have quite different and apparently inconsistent knowledge about a domain but who are able to freely communicate with each other. Observation of experts during the maintenance phase of an expert system development suggests that experts never provide information on how they reach a specific judgment. Rather the expert provides a justification that their judgement is correct. The justification they provide varies with the context in which they are asked to provide it (Compton, Horn et al. 1989; Compton and Jansen 1990; Compton and Jansen 1990). The context will vary with the questioner. An expert will justify their expert judgement in quite a different ways when queried by fellow experts, trainees, knowledge engineers, lay people. The context depends on the framework in which questions are asked. If the knowledge engineer believes that "real" knowledge is causal the expert will provide justifications in terms of causality. If the knowledge engineer favours heuristics, the expert will provide heuristics. It does not seem to be correct to approach knowledge acquisition as the task of getting to the deep and more inaccessible knowledge. Rather experts always justify their judgement with what seems like equal ease. The problem is that this justification varies with the context and has to be "engineered" to fit in with the other knowledge in the knowledge base. The experts do not become better at giving "ready to use" knowledge unless they become in effect knowledge engineers.

The best way of understanding knowledge seems to be to view knowledge and knowledge bases as models (Clancey 1989). As Clancey points out the basic feature of a model is that it can simulate reality, behave like reality according to the expectations of the users. From this point of view a knowledge base is not a model of the expert's knowledge, both the knowledge base and the expert's knowledge are models of the domain. This is essentially traditional philosophy. Knowledge is different from reality or the thing in itself, knowledge is the way in which the knower relates to or "understands" the known. The concept of model captures this same dichotomy. The model is not the thing but it behaves like the thing, not in a absolute sense but according to the expectations of the model or knowledge users. Clancey points out that even a set of rules which merely capture heuristics about the domain, in some sense model the domain and behave like the domain. There are of course many different kinds of models which one can build, and may different criteria as to what is an acceptable simulation of reality.

From the modelling point of view the expert's "knowledge" or justification in context is the creation of a model to explain how reality works to fit with or replace other models of reality within the constraints of certain modelling paradigms.   This modelling view is in turn consistent with Popper's falsification approach to the development of knowledge (Popper 1963). Knowledge is always a hypothesis which can never be proven correct, at most it can be proven incorrect and be replaced by another perhaps "less false" hypothesis. The hypothesis is a model and as such is intrinsically different from reality. Obviously the model is also constructed within a particular paradigm, particular context and so will have to changed or replaced to cope with other points of view. None of this implies a relativist position. The "truth" of knowledge is found in Lonergan's concept of "insight" (Lonergan 1959). Essentially "knowledge" is the model we make of reality to express our insight into reality, our experience of some intelligibility in reality. Conversely "insight" is the process or act of recognising that the knowledge we have constructed is a model of reality, that it makes sense of reality in some way.

Clancey [Clancey, 1989] and Gaines (Gaines 1991) provide excellent introductions to these ways of thinking about knowledge from the perspective of AI practitioners rather than the opposition to AI perspective of Dreyfus (Dreyfus and Dreyfus). They touch on issues beyond the scope of this paper, suggesting that knowledge does not reside in the head but in its expression in some medium, and that knowledge has an important social dimension. The important aspect of these different approaches to knowledge is not that they question the possibility of artificial intelligence   but that they open up different approaches to acquiring, representing and using knowledge. Ripple down rules which will be outlined in more detail below is one attempt to handle knowledge acquisition and use form this perspective. Repertory grid methods such as KSSO can also be viewed as capturing similar philosophical concerns (Gaines and Shaw 1990).

One of the consequences of the argument above is that any attempt to understand how an expert reaches decisions by asking questions or interviewing in essence asks the expert to provide a justification for his decisions within the context of the interviewing or questioning paradigm. The model the expert helps the knowledge engineer construct is always influenced by the framework the knowledge engineer is working in, the tools available etc. Since this always happens it can be argued that there may be advantage in having fairly strong tools, which allow the expert very limited but simple choices in providing knowledge, but which are adequate to model the domain. The strong tool may have considerable advantage over the weak methods which leave it to the expert and knowledge engineer to structure the knowledge. In the weak methods the structure for the knowledge is implicit and therefore difficult to maintain and little advantage has been gained in trying to model the experts knowledge because the expert has been in fact justifying his or her judgement within the framework of the knowledge acquisition tools and techniques and the knowledge engineer's concerns. (Alain Rappaport, personal communication).

We have developed a knowledge acquisition methodology "ripple down rules" which attempts to recognise that the knowledge the experts provide is only a justification in context and that this justification will be most reliable if used only in the same context (Compton and Jansen ; Compton and Jansen). The method described below can be considered independently of the analysis above and it stands or falls on its performance. However, this section started with the strong claim that philosophical assumptions influence the development of practical methodologies. The ripple down rules method was explicitly based on philosophical assumptions which were noted in 1987 before the method was developed (Compton). The philosophical analysis above, although now expanded is not post hoc.

RIPPLE DOWN RULES

Most of the experiments we have carried out have been in the domain of providing clinical interpretations for pathology reports. That is, a comment is appended to the report explaining what the results mean, what tests should or should not be ordered in the future etc. The crucial component of the context in this domain is that the expert is justifying why their new interpretation is better that the interpretation given for the case (perhaps no interpretation). If the expert was justifying why their interpretation was better than another person's, they would try provide the justification in terms of what they assumed to be the other person's beliefs and assumptions etc. If one considers the justifications provided to a patient versus when a colleague queries the expert's interpretation of a report, this attempted guessing of the other person's knowledge or reason for the query is obvious. In the case of the expert system, the knowledge behind the interpretation does not have to be guessed, it is all the knowledge in the knowledge base that has determined the interpretation provided by the expert system. This includes both rules that have actually been satisfied by the data, and rules that have been candidates to fire during the inference process but have not been satisfied by the data. In other words we want the new rule that is being added to be applied to a set of data only if the data is able to satisfy the rules that previously provided the wrong interpretation and if the data does not satisfy other rules also not satisfied in providing the earlier wrong interpretation.

FIGURE 1 - A ripple down rule knowledge base. Each node in the tree is a rule with any desired conjunction of conditions. Each node also has a case associated with it, the case that prompted the inclusion of the rule. In this example rules are numbered as are the leaves of the tree to indicate which rule the classification comes from when a case traverses the tree. The classification comes from the last rule that was satisfied by the data. The heavy line illustrates a path through the knowledge base for a case giving interpretation 9. If this is the wrong interpretation and a rule has to added to give interpretation 11, it would be joined onto the unsatisfied branch of rule 10 as shown, as this was the exit point from the knowledge base for that particular case differences

RIPPLE DOWN RULE STRUCTURE

To achieve this the expert system is built as a tree with a rule at each node with two branches depending on whether the rule is satisfied or not, by the data being considered. Any new rule that is added in response to a wrong interpretation, is attached to the branch at which the expert system terminated, thus making a new node. For example, in a one rule expert system, if the rule gives an interpretation which the expert thinks is wrong the new rule will be attached to the satisfied branch of the first rule. On the other hand if the one rule expert system fails to give an interpretation the new rule added will be attached to the unsatisfied branch. The tree then evolves over time as the expert corrects wrong interpretations. Note that each node of the tree is a rule of any desired complexity and that the interpretation produced from the tree is that from the last rule that was satisfied by the data. (Fig 1)

This knowledge base structure greatly facilitates rule addition. Normally when rules are added or changed the performance of the expert system has to be checked against cases. One approach to this is to use "cornerstone cases", cases for which the rules were previously changed (Compton, Horn et al. 1989)). One has to check all such cases. Here this is unnecessary. The only cornerstone case that can be misinterpreted by the new rule is the case associated with the rule that last fired, the rule that gave the wrong interpretation, which the new rule is going to correct. Rather than check this case, the expert can select conditions for the new rule from a list of differences between the case for which the rule is added and the cases associated with the last rule that fired, giving the wrong interpretation.

For example:

old case	**new case**
TSH high	TSH high
T3 low	T3 low
FTI normal	
	TT4 high

The expert must choose either or both the conditions

FTI NOT normal

TT4 high

as conditions in the rule and can optionally chose any of the common conditions to make the rule intelligible. Such a rule is guaranteed to work on the new case but not the old case, so no further checking is required or relevant. Once a condition from the difference list has been selected the expert can optionally add any other conditions from the new case which help make the rule intelligible. These extra conditions will not affect the performance of the rule on the new or old case, but may improve its performance on yet unseen cases by narrowing the rule's scope. Note that the raw pathology data has been preprocessed to "high" "low" etc. before the differences are calculated.

It can be noted that this method of differences relates to other knowledge acquisition methodologies based on personal construct theory, in turn based on the assumption that experts are good at identifying differences (Gaines and Shaw 1990). In contrast to these other methods which ask the expert to think of a difference, here the expert only has to identify which differences are important.

EXPERIENCE

Our initial experiments were concerned with redeveloping GARVAN-ES1. We found that no knowledge engineering was required and that 'ripple down rules' could be simply added to the knowledge base as provided by the expert. This results in knowledge acquisition at least 40 times as fast as that required for a conventional version of the same knowledge base, with the same knowledge engineer/expert involved.(Compton and Jansen 1990a & b). Other versions of "ripple down rules" have been developed in Prolog and in Hypercard. Brian Gaines has developed a further version in a knowledge representation server based on a term subsumption language (Gaines 1991)

We have developed a more general purpose ripple down rule system which has recently been put into routine use in the Department of Chemical Pathology St.Vincent's Hospital, Sydney. It is intended that this system will be able to deal with interpretative reports for all of chemical Pathology. Chemical Pathology covers hundred of different analytes and individual reports may include time series data on twenty or more analytes. The St. Vincent's system has two modules, a batch mode version that runs automatically to append comments to reports and a maintenance version which shows the case differences and which the expert uses to add new rules. Both versions are written in C and run on a Vax under VMS. This system is more. An example case interpreted using the system is shown in Fig 2.

```
 05-Jul-91              ST. VINCENT'S HOSPITAL                12:58 PM
                     DEPARTMENT OF CHEMICAL PATHOLOGY
                   ----------------------------------------
  Patient ....... : LAIDUP,I.M. Male, 72 years

  M.R.N. ........ : 02-62-64  Ward : CH ITU
```

Test	Range	Units	BLOOD 746406 12:55 03 Jul 1991	BLOOD 746573 07:12 04 Jul 1991	BLOOD 746590 08:46 04 Jul 1991	BLOOD 746986 07:14 05 Jul 1991	BLOOD 747004 09:15 05 Jul 1991
Sodium	137-146	mmol/L	....	....	139	....	140
Potassium	3.5-5.0	mmol/L	....	....	4.0	....	3.8
Chloride	95-105	mmol/L	....	....	103	....	104
Bicarbonate	24-31	mmol/L	....	....	23*	....	22*
Urea	3.0-8.5	mmol/L	....	....	8.7*	....	8.6*
Creatinine	0.06-0.12	mmol/L	....	....	0.14*	....	0.13*
Inorg. Phos.	0.70-1.40	mmol/L	....	....	....	....	0.90
Magnesium	0.85-1.05	mmol/L	....	....	....	....	0.86
Calcium	2.10-2.60	mmol/L	....	....	....	....	2.16
Albumin	36-47	g/L	....	....	33*	....	34*
Tot. Protein	66-82	g/L	....	....	62*	....	65*
Bilirubin	< 18	umol/L	....	....	15	....	12
Alk. Phos.	30-100	U/L	....	....	52	....	64
ALT	< 30	U/L	....	....	33*	....	40*
GGT	< 35	U/L	....	....	46*	....	55*
pH	7.35-7.45		7.51*	7.40	....	7.43	....
PCO2	32-45	mmHg	28*	37	....	34	....
PO2	75-105	mmHg	107*	93	....	68*	....

```
  Bicarb.        24-31   mmol/L        22*     23*   ....      22*   ....
  Base Excess    -3-+3                 +1      -1    ....      -1    ....
  ------------------------------------------------------------------------
Results consistent with:
  Resolution of respiratory alkalosis.
  Evolving hypoxaemia.
```

Rule trace:

```
1:       rule false
...
67 rules fail to fire
...
301:    rule false
314:     true
 CURR(BLOOD_PH) is NORMAL
AND
 CURR(BLOOD_PO2) > 59.00
AND
 CURR(BLOOD_PO2) is LOW
370:    rule false
397:    rule false
485:     true
 MIN(BLOOD_PCO2) < NORMAL
rule: 485
```

Fig 2 illustrates a typical case interpreted by the expert system. The interpretation of the data directly follows the data. The rule trace was provided by the maintenance module, the batch run time system produces only the interpretation. This feature of the maintenance module is not normally used in arriving at a new rule. Two rules fired on this case with the interpretation provided by the last rule that fired rule 485. The CURR function returns the value for the most recent measurement for that analyte. For the analytes above this is the 7.14 sample on the 5th July. The minimum BLOOD_PCO2 is the first result. Notice that the expert has chosen to specify that BLOOD_PO2 is low but above a certain value.

The rule shown in Fig2 and the rules in Fig 3 show indicate a number of features of the system. Firstly, although the data is initially classified in terms of the laboratories normal ranges, this is not always sufficient to discriminate between cases. For example although the upper limit of normal for the analyte TSH is 6 mU/L, experts will often consider results of up to 10 mU/L or perhaps more as normal. In a normal expert system this has to be handled by borderline regions as in GARVAN-ES1 (Horn, Compton et al. 1985) or by some sort of fuzzy or probabilistic classification. With ripple down rules these strategies are not required and the cut-off values can be redefined in the context. For example in the two cases given above the TSH was high. If in the old case the value was 12 mU/L and the new case 7mU/L, the expert can incorporate into the new rule TSH < X where X is chosen so that 7 <= X < 12. The feature extractor which classifies TSH as high is not changed, but only cases with TSH values below the new cutoff will be interpreted by rules in the new context. Within this context the TSH cut-off can be further refined with any cases satisfying all the rules along the pathway having their TSH values in a very narrow range. This technique is both more powerful and more simple than attempting to arrive at a universal classification or feature extraction system that will apply to every context.

```
RULE 369

IF ((CURR(AGE)*(-0.27)) + 94) - CURR(BLOOD_PO2) > 0.00 & MIN(BLOOD_BIC) <
NORMAL THEN

RULE 379

IF  CURR(BLOOD_PH) is NORMAL & CURR(BLOOD_PCO2) is LOW & CURR(BLOOD_PO2) is
HIGH & CURR(BLOOD_BIC) is LOW  THEN

RULE 410

IF TDIFF(CURR(BLOOD_TSH),MAX(BLOOD_TSH) < 30.00 THEN

RULE 424

IF AVG(BLOOD_PO2) < 60.0 THEN

RULE 452

IF VAL(BLOOD_RGLU, MIN(BLOOD_ST)) < 5.80 & CURR(BLOOD_DYNT) is "GT100" &
MAX(BLOOD_RLU) < 10.50  THEN
```

Fig 3. Example rules showing some of the built functions used. Note in RULE 369 a mathematical expression that the expert has included. The CURR function returns the value for the most recent measurement of the analyte while MIN, MAX and AVG return the values implied in their names. The VAL function return the value for the first parameter at the time specified by the second parameter; in rule 452 it returns the value of BLOOD_RLU at the time of the minimum BOLLD_ST. TDIFF returns the time difference between the two time points specified. Notice in RULE 369 that ordinal relationships exist not only between numerical values but between the primary classifications of the data.

The second feature of the rules in Figs 2 and 3 are the functions used to deal with temporal data. These functions allow the expert to make up rules about:

- The most recent value available for an attribute
- The maximum and minimum values of attribute over the data
- The average value of an attribute over a time period, and the nett change in its values over that period (to obtain rates of change).
- The value of an attribute at the time instant specified by the current/maximum/minimum value of another attribute (or the nearest available one to this time instant).
- The time difference between a range of possible options. For example, that between the nearest available value of an attribute and the time instant specified by the current/maximum/minimum values of another attribute. Alternatively, the time difference between the current/maximum/minimum values of one attribute and the current/maximum/minimum values of another.

The experts can also include in rules complex numerical expressions relating the values returned by the functions.

These functions are all built-in and so a programmer is required if further functions are to be included. The particular functions included here were arrived at after a conventional knowledge acquisition exercise of discussing with experts what features they need to identify in the data. With ripple down rules in their current development this remains the major task for a knowledge engineer, identifying and programming such functions. As will be discussed later this is the first limitation of ripple down rules. However it should be noted that this is still a major advance on conventional expert systems. Feature extraction only appears as a problem because knowledge engineering has become so simple. In a conventional system it is extremely difficult to deal with temporal data like that shown in Fig 2 (Coiera 1989). The system here has been able to deal with a wide range of temporal data using these very simple functions because the expert is able to redefine critical values for these functions in context. It should be noted that this system is already in routine use and seems to work for all of chemical pathology and that the expert is not assisted by a knowledge engineer.

Fig 1 shows a ripple down rule knowledge base as a binary tree, however this should not be viewed as a conventional tree. Firstly each node contains a rule. The complexity of the rules is shown in Fig 4. Rules to data have up to 5 conditions. The majority of the rules have only a single condition. This does not imply the knowledge and knowledge base is simple. A rule in a conventional expert system is equivalent to a ripple down pathway from root to leaf. The number of satisfied rules along pathways is shown in Fig 5. To be fully equivalent to a conventional rule both the positive and negative branches along a pathway need to be taken into account, indicating that the knowledge in ripple down rules is fairly complex. Figs 4 and 5 also indicate the advantage of ripple down rules. Most of the individual rules added are very simple for in context it is simple to identify the one or two key conditions that discriminate between the old incorrect interpretation and the correct interpretation. Ripple down rules are very general and un-encumbered by the inclusion of the all the conditions required to make the rule work in a conventional expert system.

AN important feature of fig 5 is shown differently in Fig 6. These figures indicate that there are not many correction rules added to earlier rules. Fig 6 indicates that the knowledge base should be viewed as a conventional flat expert system with corrections added to rules. The difference from a conventional expert system is that as soon as a rule is fired only rules that are corrections to that rule can fire and the traversal down the left most branch, equivalent to the flat expert system is left behind. The longest branch is the one where all rules have failed. This indicates that although experts give knowledge in context so that ripple down rules are required, they do make reasonable good general statements and that more new rules are required than corrections to old rules. With hindsight this must be true otherwise conventional expert system would not work at all. In reality they obviously do work but are difficult to build and maintain on a large scale, because knowledge is taken out of context. The fact that we can communicate at all means that we do generate reasonably good models, reasonably good hypotheses of the world, but just as all hypotheses can be falsified because they are only true in a particular albeit fairly general context, so too, rules no matter how apparently general require further refinement in context.

Fig 6 shows the entire ripple down rule tree whereas Fig 5 indicates only the number of satisfied rules in a pathway. In contrast to normal tree diagrams all branches are the same length. Although this results in parts of the tree being overwritten. However this representation accurately indicates the number of satisfied rules and unsatisfied rules in any pathway, but because of the overlay the particular sequence of satisfied and unsatisfied rules is obscured. This figure illustrates that the tree is extremely unbalanced. The first half of the tree going form top to bottom is mainly thyroid rules while the remainder of the tree is mainly blood gas rules..

At the time of writing this system has been in routine use for about two months and so far and so far 600 rules have been added. Development commenced with thyroid data and is now mainly concerned with acid/base balance. The system also has some catecholamine and diabetes rules. The thyroid development was completely separate from the previous Garvan system to provide a further evaluation of the method. No knowledge from GARVAN-ES1 was used and the knowledge was added by an expert who was not involved in the original development (or any other expert system project). Fig 7 illustrates the rate of growth of the knowledge base. (Note that Fig 7 shows the full 600 rules but figs 4,5 &6 the first 500). At about 230 rules the thyroid component of the system is about 95% correct which is consistent with earlier Garvan results and suggests that it will at least double in size (Compton and Jansen 1990). The Garvan system was 96% correct when introduced into routine use, however it doubled in size before reaching 99.7% acceptance by experts. In the redevelopment of GARVAN-ES1 as a ripple down rule system 550 rules were required to reach approximately 99% accuracy. The rate of thyroid knowledge acquisition has slowed down to a rule every few days, so that the system seems to be in the maintenance phase. Note that on Garvan archival data about 72% of the results are normal so that an interpretation level of over 95% represents a significant development. We are not sure what to expect with the long term development of this system. The expert may eventually decide that further changes are not warranted, but since the knowledge acquisition is so easy, the system may be continually refined, incorporating new tests, new styles of interpretation, the latest fad as it arises. This is what expert's do but is it what an expert system should do?

The rate at which acid/base rules are added also appears to be starting to slow down. The expert is at present is deciding what domain to tackle next. These are still very preliminary results, but it appears clear that the ripple down rule approach is viable in the lab environment and that such systems can be built by experts without knowledge engineering help. Since the system went into routine use there have been no queries in any form to knowledge engineers and there have been no knowledge engineers available at St.Vincent's. The commencement of the acid base part of the knowledge base was entirely an expert decision as were the ways in which this was approached. The only queries have been from knowledge engineers to the experts wondering how the system is progressing. It should also be noted that the expert has no particular computer skills or interest beyond using simple wordprocessing, statistics and drawing packages as part of his normal chemical pathology work. The expert add new rules to the system in response to picking up incorrectly interpreted reports when signing out reports, which is one of normal duties. It remains an expert task to sign all reports which are issued by the laboratory, providing a perfect opportunity to check interpretations and add rules if necessary. A number of experts sign reports but all reports that are misinterpreted are currently given to the one expert to add new rules to the system.

Note that new areas can be opened up by simply adding a rule to give an interpretation for the new area. However once the new rule is entered the expert or experts must be able to follow up all misinterpretations and add correcting rules. This seems to be a manageable task in that a single expert has coped with adding rules as required for the high volume acid/base area while continuing with his normal duties. He has noted that the major problem, has not been creating the rule but deciding on the most helpful form of interpretation to provide. Different subdomains have different requirements for rules. Fig 8 illustrates that the initial rules added for the blood gas domain are much more complex than those added for thyroid data. However fairly soon the complexity of the blood gas rules dropped to the level of the thyroid rules. One would expect refinement rules are more simple than the initial rules developed, if for no other reason than conditions have been "used up" in earlier rules in the path. Some of the later rules added were added to the left most branch of the tree. We are not certain whether such rules tend to be simple because the earlier rules pick up most of the cases and the later rules just identify special features in more unusual cases, or whether the expert was nervous with the start of a new domain and so tended to make up more specific rules.

Fig 7 indicates the growth of the knowledge base. The x axis indicates the number of working days the system has been under development, about 9 weeks. The system started with approximately 200 rules because the expert was developing rules off line while interface problems with the main reporting programs were ironed out. However these rules were developed without the help of a knowledge engineer and were produced by the same process of identifying cases that required interpretation at report signing and then running them on the expert system by hand. The steep rise in the curve indicates when acid base knowledge base was commenced. Note that this is the only figure which covers the 600 rules current at the time of writing. Figs 4,5 &6 illustrate the first 500 rules.

Fig 8 indicates the number of conditions in each rule against rule number. The dots indicate the actual number of conditions in each rule while the continuous line indicates a smoothed average of the number of conditions per rule.

RIPPLE DOWN RULE LIMITATIONS

Repetitious knowledge acquisition

The most obvious problem of ripple down rules is the likelihood of repetitious knowledge acquisition. It is obvious that a ripple down rule system may end up with knowledge repeated in many places throughout the tree. We have not yet evaluated the St.Vincent's data but with the Garvan ripple down rule development repetition was not a significant problem, and there appeared to be no more than 13% of the knowledge repeated. Because the knowledge acquisition is so easy this is a small additional task. Suggestions that repetitious knowledge will be a problem are normally allied with suggestions that the tree should be periodically reorganised as it may be come very messy and unbalanced. In fact it seems that the tree's lack of balance shown in Fig 6 comes from the same cause as the small amount of repetition observed. Although experts' rules do need correcting they are fairly good rules so Fig 6 indicates more rules are constructed because the system fails to give a diagnosis, (the left most branch) than because rules need corrections. However because the acquisition is so simple the rules tend to be simple and general (Fig 4). The expert is able to make up the most general rule he or she wishes without having to add conditions because of the engineering requirements of trying to make the rule work. Thus Figs 4 and Fig 6 indicate that ripple down rules provide very simple general rules which require a fairly small number of corrections, hence the repetition is fairly small.

We have also investigated machine learning methods such as ID3 (Mansuri et al. 1990) since they produce a more optimal tree. For training sets 8000 and 9445 case we found similar sized trees to the ripple down rule tree for ID3 and C4.5.(Mansuri, Compton et al. 1991) Brian Gaines carried out similar studies using his Induct algorithm and produced slightly smaller trees (personal communication). The use of C4.5 was not optimised so that it too may produce smaller trees. Also the Garvan ripple down rule system which was used as the basis of comparison was not complete and still had a 1.1% error rate. The inductive system may thus produce a more compact tree than the ripple down rules although these studies suggest that the difference will not be major e.g. 394 rules for Induct versus 550 for ripples. Even a two or three fold difference would be acceptable in terms of the ease of building ripple down rules. These studies also indicated an important advantage of ripple down rules. The Garvan ripple down rules were built by sequentially going through a large set of cases and making up new rules for cases which were misinterpreted by the rules already in the system. When inductive methods were applied to the same data 6000 cases were required in the training set before the inductive systems performance caught up to the ripple down rules (Mansuri, Compton et al. 1991). At small numbers of cases ripple down rules outperformed inductive systems by a large margin. These results are not surprising, Gaines showed that the inclusion of rules provided by an expert greatly reduces the requirement for training cases for inductive learning (Gaines 1989). This was also a difficult task as some 60 different classification of the data were required. These results do not imply that ripple down rules are "better" than inductive methods. If large data bases of training cases are available, inductive methods are obviously easier, however if these training cases are not available ripple down rules will produce a knowledge base not too much larger than inductive methods working on large training sets. Sooner or later repetitious knowledge acquisition will be a problem, but it is not a significant problem yet. A possible solution is outlined below.

The suitability of a training set for inductive methods does not depend just on the numbers of training cases, but on how well classified the cases are. This is a major problem in the domain of pathology interpretation. Ripple down rules allow the expert to construct free text interpretations, as the expert would do if adding an interpretation manually. It is very difficult to assign these interpretations to specific classes. Even if the expert is provided with tools to check the existing interpretations to decide which of these are being repeated there is no guarantee he or she will bother to do this carefully and not just resort to the addition of anew classification. If one wishes to produce a primitive system it may be possible to get the expert to provide coarse classifications, but expert interpretations are normally very subtle - GARVAN-ES1 produced 60 interpretations for example. We are currently working on tools to assist experts in identifying the relationships between interpretations. The aim of this work is to facilitate exploration of the knowledge base for educational purposes, it is not required for ripple down rules. To build a training set for inductive learning the expert would have to accurately classify thousands of cases. With ripple down rules, the expert has to add rules only periodically and the system can be introduced into routine use immediately.

Multiple classifications

A far more important problem than repetitious knowledge acquisition is that multiple classifications may be required for example if a patient has multiple independent diseases. At present the ripple down rule approach provides a single interpretation, although that conclusion may contain a number of parts. This may lead to large portions of the tree being repeated. The obvious solution of producing separate trees is not attractive because the domains may not be clearly separated. Data from one domain may be used in another and the artificial separation into sub-domains may be too crude resulting in the recurrence of the problem within the separate trees.

Another solution is to move away from the tree representation with its single path a case. We have proposed a an approach whereby an in context knowledge acquisition methodology may be applied to a flat expert system in which multiple rules could provide interpretations which would then allow for multiple classification and help to reduce repetitious knowledge problems [Compton, 1991 #69]. In this approach rules can be modified, but the case for which the modification is made is kept resulting in perhaps more than one case per rule in contrast to the ripples down rules. When a rule is narrowed to exclude a case a difference list can be produced of the intersection of all the cases connected to the rule and the new case. Generalising a rule is more complex in that all the other cornerstone cases have to be considered in arriving at a list of conditions which will allow the new case to fire on the rule but prevent any other case firing.

With ripple down rules the exert picks from a single list. Similarly here the expert should be presented with a single list of conditions to choose from. Concepts of generalisation and specialisation should be hidden and the expert only deal with the difference list. A simpler approach may be to allow the ripple down rule interpreter to backtrack and explore any false branches. This is a legitimate strategy because the false branches are don't care rather than false. A rule is made up by the expert and automatically added to the false branch of a tree because the case failed to satisfy the rule and go down the true branch. However the expert has no notion of true and false so that a rule made up and attached to a false or don't car branch may be perfectly appropriate for some cases which satisfy the rule. An advantage of this approach over the flat system is that it may not have to deal with all the cases in the tree individually. It may be possible just to consider cases associated with the rules at the same level in the tree, that is attached to the new rule only by false branches. The rule will have to include a NOT condition from the intersection of the cases in the sub-tree attached to the satisfied branch of each of these rules.

The backtracking suggested will change the scope of the knowledge acquisition task but not its nature; the expert will still identify differences to develop rules to produce the right interpretations, but will deal with multiple interpretations rather than single interpretations and the difference list may be more complex. The selection of conditions will also indicate where in the tree a new rule should be added. We hope that if the system is built like this from the ground up it will not greatly increase the load for the expert. A prototype ripple down rule system with backtracking is currently being evaluated. We propose to minimise the problem at present at St.Vincent's by judicious choice of initial domains for interpretations.

Feature extraction

A further area of research for this system is in feature extraction or data reduction. A common assumption with knowledge based systems is that the data is going to be presented to the system in terms of the conditions the expert used in rules. This is only the case when the expert is actually the person who enters the data into the system. This problem is not normally focused on, because the major knowledge acquisition problem is knowledge engineering. Ripple down rules solves the knowledge engineering problem and leaves one confronted with the feature extraction problem. The problem is obvious in terms of pathology reports. The expert is able to look at the data and identify the appropriate classification for a report. However for the rule to apply to as many cases as possible, it must abstract from the individual features of the data in this case, and the rule for the abstraction must be known, so that it can be applied appropriately to other cases. We have proposed above that fairly simple functions such as maximum, minimum and average etc. together with the option of changing cut-off values in context are sufficient feature extractors for much of pathology; however, there will be limits to their applicability and it would be preferable for experts to define their own feature extractors. The current system allows the expert to redefine cut-off values in context. The features are not redefined however, the rule simple eliminates cases from that pathway which do not have values in the specified range. It may be better however to actually redefine the feature, particularly if the feature is more complex than those used here . This new definition would then be taken out of the context to be applied globally *to all rules developed after this point.* The definition of the feature would not be changed for any earlier rules, only for rules still to be added. It remains to be seen whether this is a useful strategy. Menzies has suggested applying this approach more generally to all procedural knowledge [Menzies, 1991]. Ideally experts should to be able to define features themselves. For complex features it would be more appropriate to use a programmer or knowledge engineer, but experts may be able toe deal with simple feature extraction or minor extensions to complex feature extractors.

The distinction we have made between feature extraction and other rules is not clear cut as feature extractors can be expressed in rules and are context dependent. The flat nature of the ripple down rules whereby each rule provides a classification provides a basis for a distinction. A feature can be defined as any aspect of the data that is used in a rule to provide a classification. A feature reduces the data in that a number of different data patterns may show the same feature and so using this feature in rules enables the rules to be more general. This distinction is not so clear cut in expert systems which use an number of levels of intermediate variables in their reasoning. The rules that define features will be context dependent and should be able to be refined in context in the same way as ripple down rules, but it is implicit that may have a more general scope than a particular context in the ripple down rule tree. The solution seems to be to maintain trees both for feature refinement and the actual knowledge used to reach final conclusions. However links are maintained between the trees so that newly defined or modified features are only used in rules added to the main tree after the feature has been changed. This suggestion remains to be evaluated.

It should be noted that feature extraction only emerges as a problem because the knowledge acquisition has been simplified. Feature extraction is normally inextricably interwoven with the problems of knowledge engineering. Feature extraction is also quite different form attribute identification. In the type of domains we are considering the data to be used in the expert system has all been identified, generally because it is available on a computer. Feature extraction is concerned with identifying the features of the data that experts wish to use in rules. Attribute identification is concerned with identifying what data is to be used by the expert system and is required in domains where it is not known what data is used by the expert. The attributes identified by repertory grid tools such KSSO (Gaines and Shaw 1990) will normally also be features in that they will normally be used directly in rules. The importance of feature extraction does not apply in cases where humans who can identify the features required enter data into the expert system.

POSSIBILITIES

Ripple down rules makes possible a particular type of expert system. The traditional expert system is normally set up as a once only effort with the hope that its expertise will be useful for a long time to come. Maintenance problems generally confound this hope. Repertory grid tools in contrast seem to be often used for creating "disposable" knowledge bases. The development of the knowledge base is used to clarify and support decision making when complex and important decision are being made. The 500 or 600 hundred knowledge bases at Boeing which seem to have been used and then discarded have been used in this way (John Boose, personal communication). Ripple down rules opens up the possibility of the evolving knowledge base. AS knowledge changes further refinements are added to the knowledge base. It should be noted that knowledge rarely changes instantaneously. As new sources of data are brought into play in decision making it takes a considerable time before experts fully explore how the data should be used and move away from earlier inadequate data. Ripple down rules provides a way of allowing the knowledge to gradually move with the changes. It also provides a way of tailoring knowledge to very specific local concerns - just add further refinement to the knowledge base. In the pathology area knowledge base may be passed from lab to lab but these will then be tailored to local preferences and it is probably more likely that they will be entirely local.

A question arises with expert systems as to whether they perform well according to objective tests. This is not actually a problem for the expert system but for the expert. If the expert system accurately reflects the expert, then the question is whether the expert performs reliably. This distinction becomes more obvious with ripple down rules as the expert refine the system to his or her personal idiosyncrasies. The current expectation at St.Vincent's is that there will never be a final expert system for the interpretation of all of chemical pathology, but that it will constantly evolve as lab practice evolves. None of this diminishes its role in providing automated interpretations of laboratory reports to assist clinicians receiving the report. However, it evolves as does the pathologists expertise.

The current application areas of ripple down rules are for classification tasks where all the data required is available and where there is an expert or some suitable person able to detect errors in the system makes in classifying cases. This final requirement is not onerous. If any change is to be made to acknowledge base it will be made in response to a case being miss-classified. This implies the existence of both the case and someone to detect the missclassification and so provides the basis for ripple down rules. Secondly any significant expert system will be tested on enough data to evaluate all aspects of its knowledge before being put into use and again the output will be checked by someone. Such a system could have been built using ripple down rules. The conventional approach to building an expert system is to build the knowledge base and then validate it. Ripple down rules combines the validate and knowledge acquisition processes.

It is likely that ripple down rules would also be suitable for problems requiring backtracking. Backtracking is used whenever one wishes to minimise data collection. The tree form of ripple down rules and the small number of conditions in most rules suggest that a fairly minimal number of conditions will be asked for, but it may be possible to further reduce this. One always starts at the top most rule in using ripple down rules. However in asking for data for this rule one could determine the order in which data is requested according to the frequency with which this condition appears in all the rules in the left most path of the tree, so that as many rules are eliminated as possible if the condition is not present in the data. If the top rule fails the process is repeated on the next rule down until a rule fires. Once a rule is found that fires the process is repeated on level down. This is only a preliminary suggestion which needs expansion to deal with ripple won rule with backtracking.

A final proposal is that ripple down rules may provide a particularly useful rule trace. A conventional rule trace provides little explanatory power because the experts knowledge is obscured by conditions added to rules for engineering reasons. With ripples the rule is exactly as expressed by experts. Further, each rule is a correction of an earlier rule and indicates the conditions under which the earlier rule does not apply. Each rule also has the case that promoted its inclusion stored. This may be a important educational resource, particularly in medicine where much of the training is cased based. A pathway may also includes a number of rules which failed. The conditions in some of these rules may not contradict the conditions in the rules that were satisfied. These rules thus indicate conditions which if they were present would alter the classification being made and may also be used for training.

SUMMARY

It has been proposed that expert knowledge is always provided in context and should therefore be used in context. A knowledge acquisition methodology has been proposed to capture and use knowledge in context. In this method the only knowledge acquisition task, is for the expert to select from a list of conditions. The expert thus has a very restricted task and no input into the way the knowledge base is structured. However this greatly simplifies the expert's task and he or she need understand nothing of the knowledge base structure. We have implemented a system based on this approach which is now in routine use in a pathology laboratory with knowledge added by experts without the intervention of a knowledge engineer. This approach seems to simplify problems such as handling temporal data, and dealing with probabilities. Some limitations remain such as dealing with multiple classifications and further research in these areas is under way.

References

Clancey, W. (1989). "Viewing knowledge bases as qualitative models." IEEE Expert **Summer**: 9-23.

Coiera, E. (1989). Intelligent patient monitoring. Proceedings of the Fifth Australian Conference on Applications of Expert Systems, Sydney.

Compton, P. (1989). Expert systems for the clinical interpretation of laboratory reports. Clinical Chemistry: An Overview. Proceeings of the 1987 International Congress of Clinical Chemistry. O. Van der Heiden , N. den Boer and J. Souverijn . New York, Plenum**:** 615-628.

Compton, P., R. Horn, et al. (1989). Maintaining an expert system. Applications of Expert Systems. J. R. Quinlan. London, Addison Wesley. **2:** 366-385.

Compton, P. and R. Jansen (1990). Knowledge in context: A strategy for expert system maintenance. Proc AI 88. C. Barter and M. Brooks. Berlin, Springer-Verlag**:** 292-306.

Compton, P. J. and R. Jansen (1990). "A philosophical basis for knowledge acquisition." Knowledge Acquisition **2**: 241-257.

Dreyfus, H. and S. Dreyfus (1988). "Making a mind versus modelling the brain: artificial intelligence back at a branchpoin." Daedalus **117**(Winter): 15-43.

Gaines, B. (1989). Knowledge acquisition: the continuum linking machine learning and experise transfer. Proceedings of the third European workshop on knowledge acquisition for knowledge-based systems, Paris.

Gaines, B. (1991). Between Neuron, culture and logic: explicating the cognitive nexus. Proceedings of ICO'91: Cognitive Science: Tools for the development of organisations.

Gaines, B. (1991). Integrating rules in term subsumption knowledge representation servers. Proceedings of AAAI-91.

Gaines, B. and M. Shaw (1990). Cognitive and Logical Foundations of Knowledge Acquisition. 5th AAAI Knowledge Acquisition for Knowledge Based Systems Workshop, Bannf.

Horn, K., P. J. Compton, et al. (1985). "An expert system for the interpretation of thyroid assays in a clinical laboratory." Aust Comput J **17**(1): 7-11.

Lonergan, B. (1959). Insight. London, Darton, Longman and Todd.

Mansuri, Y., P. Compton, et al. (1991). A comparison of a manual knowledge acquisition method and an inductive learning method. Australian workshop on knowledge acquisition for knowledge based systems, Pokolbin.

Popper, K. (1963). Conjectures and refutations. London, Routledge and Kegan Paul.

Shaw, M. (1988). Validation in a knowledge acquisition system with multiple experts. Proceedings of the International Conference on Fifth Generation Computer Systems, Tokyo.