Finding Conceptual Models to Assist Validation

Debbie Richards and Paul Compton

Department of Artificial Intelligence
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
Email: {debbier, compton}@cse.unsw.edu.au

Abstract

Many approaches to knowledge based systems (KBS) development attempt to build complete systems that are mostly considered final before they are put into production. Verification and Validation (V&V) is seen as an activity performed before the system is installed and little consideration is given to incremental KA, maintenance or validation of the system after installation. We take a situated view of knowledge as something that is created to suit the situation and that knowledge evolves as contexts change. Therefore we do not regard KA and maintenance as two distinct phases. For actual KA and maintenance we use a technique, known as ripple down rules (RDR) that is simple, yet reliable. RDR provides online validation of rules by ensuring that no new rule results in a previous rule giving the wrong conclusion. We seek to further improve the validation capabilities of RDR by retrospectively finding the models inherent in our knowledge base. We have added tools based on Formal Concept Analysis to RDR to assist the user with KA by showing the user whether the new rule and/or conclusion are consistent with existing concepts in the knowledge base and where they fit into the subsumption lattice of concepts.

1 Introduction

Many approaches to knowledge based systems (KBS) development attempt to build complete systems that are mostly considered final before they are put into production. These approaches are based on Newell's (1982) Knowledge Level which advocates the modeling of knowledge at a level above its symbolic representation and includes modeling of problem solving methods (Chandrasekaran 1986, McDermott 1988, Puerta et al 1992, Schreiber, Weilinga and Breuker 1993 and Steels 1993) and ontologies (Guha and Lenat 1990, Patil et al 1992 and Pirlein and Struder 1994). The need for complex modeling for KA has resulted in the development of verification (Cragun and Streuduel 1987, Preece, Shinghal and Batarekh 1992, Suwa 1982) and validation (O'Keefe and Leary 1993) (V&V) techniques that are designed for use before the system goes into routine use. There is little consideration for incremental validation of such systems and maintenance is often a neglected problem (Kang, Gambetta and Compton 1996, Menzies and Compton 1995 and Soloway, Bachant and Jensen 1987). While methods do exist, validation of systems is often left to vague heuristics such as: If the result is similar to an experts then it is okay. Perhaps validation poses such a problem due to the difficulty of trying to validate models that are by their very nature inaccurate (Clancey 1991). If models are at best imperfect representations that vary between users and the same user over time (Gaines and Shaw 1989) then an emphasis on modeling as a prerequisite for building KBS seems to impose a structure that does not actually exist.

The approach we have taken is based on a situated view of knowledge as something that is constructed to suit the particular situation (Clancey 1991b). We have chosen a KA and representation technique, known as ripple down rules (RDR) that incrementally captures knowledge in context. Refinements to existing rules always occur within the same context. The technique has been shown to address the KA bottleneck and maintenance problems. This has been achieved without the need for *a priori* modeling or the intervention of a knowledge engineer. We believe that starting with models is problematic because experts are often unaware of their own conceptual models and have difficulty describing why they would make a particular conclusion. However, we also believe that it is possible to build systems that exhibit behaviour similar to an experts and then extract the models underlying the knowledge captured. As explained by Clancey (1988) concerning the process of extracting conceptual and procedural abstractions from MYCIN into NEOMYCIN, the most famous reuse of knowledge, "we are stating a model that goes well beyond what experts state without our help" (Clancey 1991b, p261). By finding these inherent models we are also providing a means of improving validation of the knowledge. As a new rule is entered or a new conclusion is chosen we show the user how the new concept fits in with the existing knowledge and notify the user if the knowledge is inconsistent. We have added ideas from Formal Concept Analsysis (FCA) (Wille 1982) that allow us to generate the concepts associated with a rule base and then we use these concepts for comparison.

We first briefly describe how KA, maintenance and validation are currently performed in RDR. Then we describe the way FCA develops concepts and how we have added tools using these ideas. We finish with an evaluation of what we have done thus far and what we still plan to do.

2 Ripple Down Rules

Ripple down rules were proposed in answer to the problem of maintaining a large clinical pathology KBS. It was observed that experts did not offer an explanation of why they took a particular course of action but rather they offered a justification of their action and that justification depended on the situation (Compton and Jansen 1990). To avoid the problem of side-effects that occur when maintaining a typical production rule KBS (Grossner et al 1993), rules are never changed or deleted in an RDR KBS. If a case is misclassified a new rule is added as a refinement to the rule that gave the wrong conclusion. This new rule is reached only if the same sequence of rules is followed. The last true rule is the conclusion given. The utility of RDR has been demonstrated by Pathology Expert Interpretive Reporting System (PEIRS) (Edwards et al 1993) which went into routine use in a large Sydney hospital with about 200 rules and grew to over 2000 rules over a four year period (1990-1994).

In single classification RDR we can define an RDR as a triple <rule,X,N>, where X are the exception rules and N are the if-not rules (Scheffer 1996). When a rule is satisfied the exception rules are evaluated and none of the lower rules are tested. This study has used Multiple Classification RDR (MCRDR) (Kang, Compton and Preston 1995) which is defined as the triple <rule,C,S>, where C are the children rules and S are the siblings. All siblings at the first level are evaluated and if true the list of children are evaluated until all children from true parents have been exhausted. The last true rule on each pathway forms the conclusion for the case. MCRDR was chosen since the ability to provide multiple conclusion for a given case is more appropriate for many domains and, more importantly, because the problem of how to handle the false "if-not" branches (Richards, Chellen and Compton 1996) does not exist.

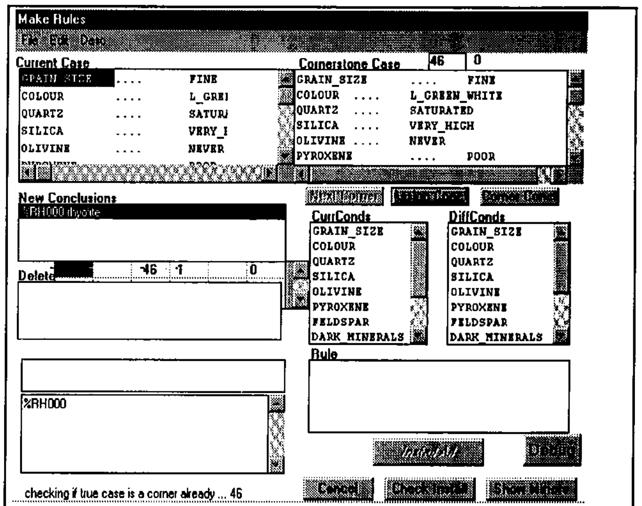

Figure 1: The MCRDR screen for making rules

RDR handles the issue of context by its exception structure and the storing of the case that prompted a rule to be added. This case is known as the cornerstone case and assists the expert in identifying the features in the current misclassified case that not only apply to the new classification but also differentiate it from the case associated with the rule that fired incorrectly. In Figure 1 we see the MCRDR for Windows screen for making new rules. The user must select the conclusion and then specifiy the attribute-value pairs (rule clauses) that distinguish the current case from the cornerstone cases associated with the rule that incorrectly fired. In this way the new rule is validated online against the case when it is added. With MCRDR multiple cases are involved in this evaluation, but it has been shown this is efficient (Kang, Compton and Preston 1995). The compactness and efficiency of both RDR and MCRDR have been demonstrated in simulation studies (Kang, Compton and Preston 1995).

RDR does not differentiate between initial KA and system maintenance. RDR develops the whole system on a case by case basis and automatically structures the KBS in such a way to ensure changes are incremental. Most approaches seek to define some globally applicable

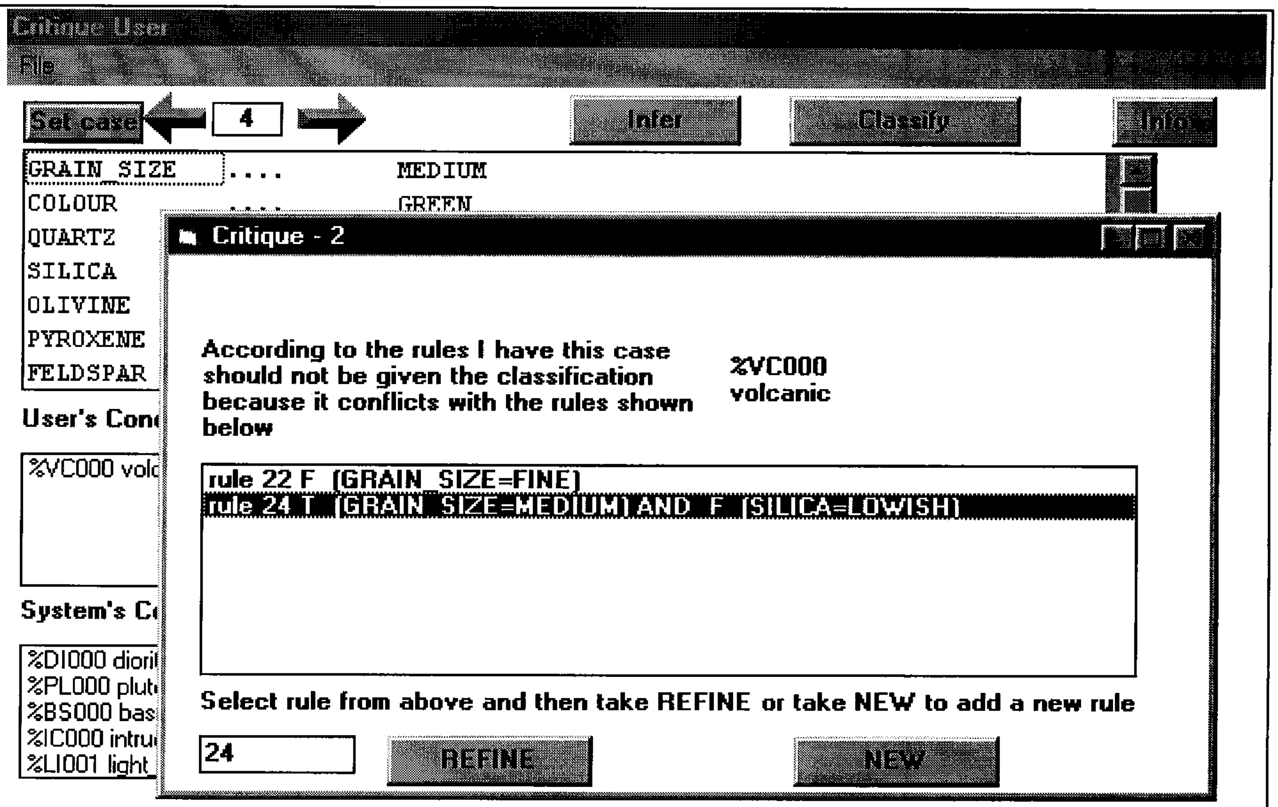

Figure 2: The critiquing screen in MCRDR/FCA, the user is shown inconsistent pathways

rules and require some consideration of the whole domain. This is true of methods that begin with modeling and is also true of methods, such as FCA and Repertory Grids (Gaines and Shaw 1989), that attempt to minimise modeling and allow the user to enter the knowledge directly. With the latter methods, incremental maintenance is often addressed by regenerating implications associated with the revised data set. The validation of rules provided by RDR and MCRDR is not total validation, but it ensures that the new rule is sufficiently specific and different to the wrong rule to assign the new classification to the current case but not to cover previously correctly classified cases.

We have also been looking at providing KA in a critiquing mode so that when a user selects a certain conclusion it is evaluated against all other paths (rules) and if those rules are inconsistent with the current case the user is notified of the discrepancy. In Figure 2, using the data from the SISYPHUS III (Shadbolt 1996) experiments, we have attempted to add a rule that states that a particular rock, already classified as plutonic, should be classified as volcanic. These conclusions are mutually exclusive, therefore it is inconsistent to assign both conclusions to the same case. This inconsistency has been detected by the system. The user is shown the rules in conflict and can then change their conclusion, select an existing rule to modify or add a new rule. Once the changes are made the case is run again and the user is given all conclusions for that case. This should now show the new conclusions and any conclusions that were not being altered.

3 Formal Concept Analysis

Formal Concept Analysis, first developed by Wille {1982), is "based on the philosophical understanding of a concept as a unit of thought consisting of two parts: the extension and intension (comprehension); the extension covers all objects (entities) belonging to the concept while the intension comprises all attributes (or properties) valid for all those objects" (1992, p.493). A *formal context* is a mathematical model in crosstable form that shows a set of objects and their attributes, known as the extension and intension respectively, and how they are related. A cross indicates that a particular object in a row has the corresponding attribute in a column, see Figure 3. The following description of FCA follows Wille (1982).

A formal context ($\mathbb{K}$) has a set of objects G (for *Gegenstande* in German) and set of attributes M (for *Merkmale* in German) which are linked by a binary relation I which indicates that the object g (from the set G) has the attribute m (from the set M) and is defined as: $\mathbb{K} = (G,M,I)$. Thus in figure 3 we have the context $\mathbb{K}$ of animals with G = {bird, reptile, amphibian, mammal and fish} and M = {has wings, flys, suckles young, warm-blooded, cold-blooded, breeds in water, breeds on land, has scales}. The crosses show where the relation I exists, thus I = {(bird,has wings), (bird,flys), (bird,cold-blooded), (bird,breeds on land), (reptile, cold-blooded),...,(fish, has scales)}.

A formal concept is a pair (X,Y) where X is the *extent*, the set of objects, and Y is the *intent*, the set of attributes, for the concept. The derivation operators:

$$X \subseteq G: \; X \mapsto X' := \{m \in M \mid gIm \text{ for all } g \in X\}$$
$$Y \subseteq M: \; Y \mapsto Y' := \{g \in G \mid gIm \text{ for all } m \in Y\}$$

are used to construct all formal concepts of a formal context, by finding the pairs (X″,X′) and (Y′,Y″). We can obtain all extents X′ by determining all row-intents {g}′ with g ∈ G and then finding all their intersections. Another approach it to find all intents Y′ by determining all column-extents {m}′ with m ∈ M and then finding all their intersection. This is specified as:

$$X' = \bigcap_{g \in X} \{g\}' \qquad\qquad Y' = \bigcap_{m \in Y} \{m\}'$$

Less formally, we take the set of objects, G, to form the initial extent X which also represents our largest concept. We then process each attribute sequentially in the set M, finding the intersections of the extent for that attribute with all previous extents. Once the extents have been found for all attributes, the intents X′ for each extent X may be found by taking the intersection of the intents for each object within the set. Thereby we determine all formal concepts of the context $\mathbb{K}$ by finding the pairs (X,X′).

The next step is to find the subconcept-superconcept relation between concepts so that they may be ordered. We can use the subsumption relation ≤ on the set of all concepts formed such that $(X_1,Y_1) \leq (X_2,Y_2)$ iff $X_1 \subseteq X_2$. For a family (Xi,Yi) of formal concepts of $\mathbb{K}$ the greatest subconcept, the join, and the smallest superconcept, the meet, are respectively given by:

$$\bigvee_{i \in I} (X_i,B_i) := \Big(\big(\bigcup_{i \in I} A_i\big)'', \bigcap_{i \in I} B_i\Big)$$
$$\bigwedge_{i \in I} (X_i,B_i) := \Big(\bigcap_{i \in I} A_i, \big(\bigcup_{i \in I} B_i\big)''\Big)$$

From Lattice Theory, we are able to form a complete lattice, called a concept lattice and denoted $\mathbf{B}(\mathbb{K})$, with the ordered concept set. The concept lattice provides "hierarchical conceptual clustering of the objects (via the extents) .... and a representation of all implications between the attributes (via its intents)" [Wille 1992, p.497]. The line diagram in figure 5 is our implementation, called MCRDR/FCA. Each concept has various intents and extents associated with it and is shown as a circle. The lines represent the sub/superconcept relations. For easier reading we have reduced the labeling. All extents are reached by descending paths from the concept δ and all intents of a concept δ are reached by ascending paths from δ.

	a	b	c	d	e	f	g	h
Bird	X	X			X		X	
Reptile					X		X	X
Amphibian					X	X		
Mammal			X	X			X	
Fish					X	X		X

Figure 3: Context of "Vertebrates of the Animal Kingdom". Columns a-h represent has-wings, flies, suckles-young, warm-blooded, cold-blooded, breeds-in-water, breeds-on-land and has-scales.

FCA shares a number of similarities with RDR. They both see KA as a task that should be primarily performed by experts and they see knowledge as only applying in a given context (Compton and Jansen 1990, Wille 1996). In FCA, KA consists of the expert defining a formal context. The subsumption relation provided by the concept lattice can be used to derive implications for use in a knowledge base (Wille 1992). We investigate starting from the opposite direction by using existing rules in an RDR KBS to define contexts and then generate the concepts using FCA

4 Finding Concepts in RDR Using FCA

As discussed in section one, we wanted to understand the underlying relationships and models inherent in the RDR rules to assist us with validation of new rules and conclusions. To test the benefits and suitability of FCA for such a purpose, MCRDR for Windows was enhanced with FCA tools. The following discussion refers to this implementation, known as MCRDR/FCA.

To derive concepts we need to use the rules to generate a context. The first step was to convert the RDR KBS into a flat structure made up of rule pathways. Pathways were found by reading each rule and picking up the conditions from the parent rule until the top node with the default rule was reached. From this flattened KBS the user chooses either the whole KB or a more narrow focus of attention from which to derive a formal context. When the whole KB is chosen the rules and rule clauses form the extents and intents, respectively. Selecting all rules is only feasible for small, if not very small, KBS. The line diagram became too cluttered to be readable as the number of rules being modeled grew. This is a limitation of graphical representation. Therefore, to limit the concepts to a manageable size the user was asked to narrow their focus of attention. Our approach is similar to that proposed by Ganter (1988) where the context is shortened to find subcontexts and subrelations. The decomposition of a concept lattice into smaller parts is a strategy that has previously been found useful (Wille 1989a).

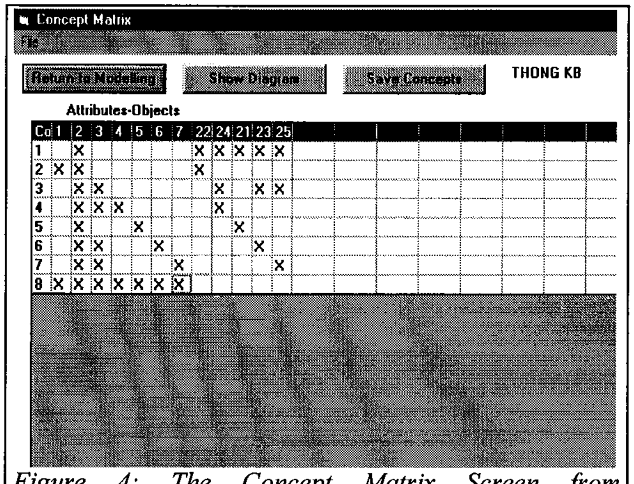

Figure 4: The Concept Matrix Screen from MCRDR/FCA. Here the user is shown the concepts for the context generated from the rules which conclude PL000 or VC000 for the SISYPHUS III domain.

There are thirteen possible ways that a context can be derived. It is also possible to combine contexts. The two most common methods are to create a context based on a selected rule or conclusion If a conclusion is chosen, all rules using that conclusion are selected and added as objects to the set G, forming the extents of the context. As each extent is added the clauses of the rules are added to the set M of attributes to form the intents of the context, first checking to see if any attributes have already been added by previous rules. Where the relation I held, that is object g had attribute m, a cross was marked in the appropriate row and column. If the user chooses a particular rule then that rule is added as the first object with the rule clauses as the initial intension. Every clause in each rule in the flattened RDR rule base is searched for a match on the initial set of attributes. If a match is found, that rule is added to the extension and all new attributes (clauses) found in the matching rule are also added to the intension.

Thus we are treating the rule clause, which is actually an attribute-value pair, as an attribute. This is similar to the technique known as *conceptual scaling* (Ganter and Wille 1989) which has been used to interpret a many-valued context into a (binary) formal context. A many-valued context, such as that represented in an RDR KBS, is a quadruple (G,M,W,I) where I is a ternary relation between the set of objects G, the set of attributes M and the set of attribute values W (merkmalsWerte in german). Basically, each attribute is treated as a separate formal context with the values as attributes associated with each of the original objects. A scale is chosen, such as a nominal scale (=) or an ordinal scale (≥), to order these attributes. From the many contexts, one for each attribute, the concepts are derived.

Having generated a formal context we can then construct all formal concepts of the formal context, using the process described in section 3. The algorithm used for ordering will affect the appearance of the concept matrix, see Figure 4, but does not affect the calculation of predecessors and successors or the graph layout. Appropriate ordering of concepts can be difficult as a given concept may be a subconcept of different superconcepts. This can be seen in the concept lattice in Figure 4 where we can see a number of sets of concepts.

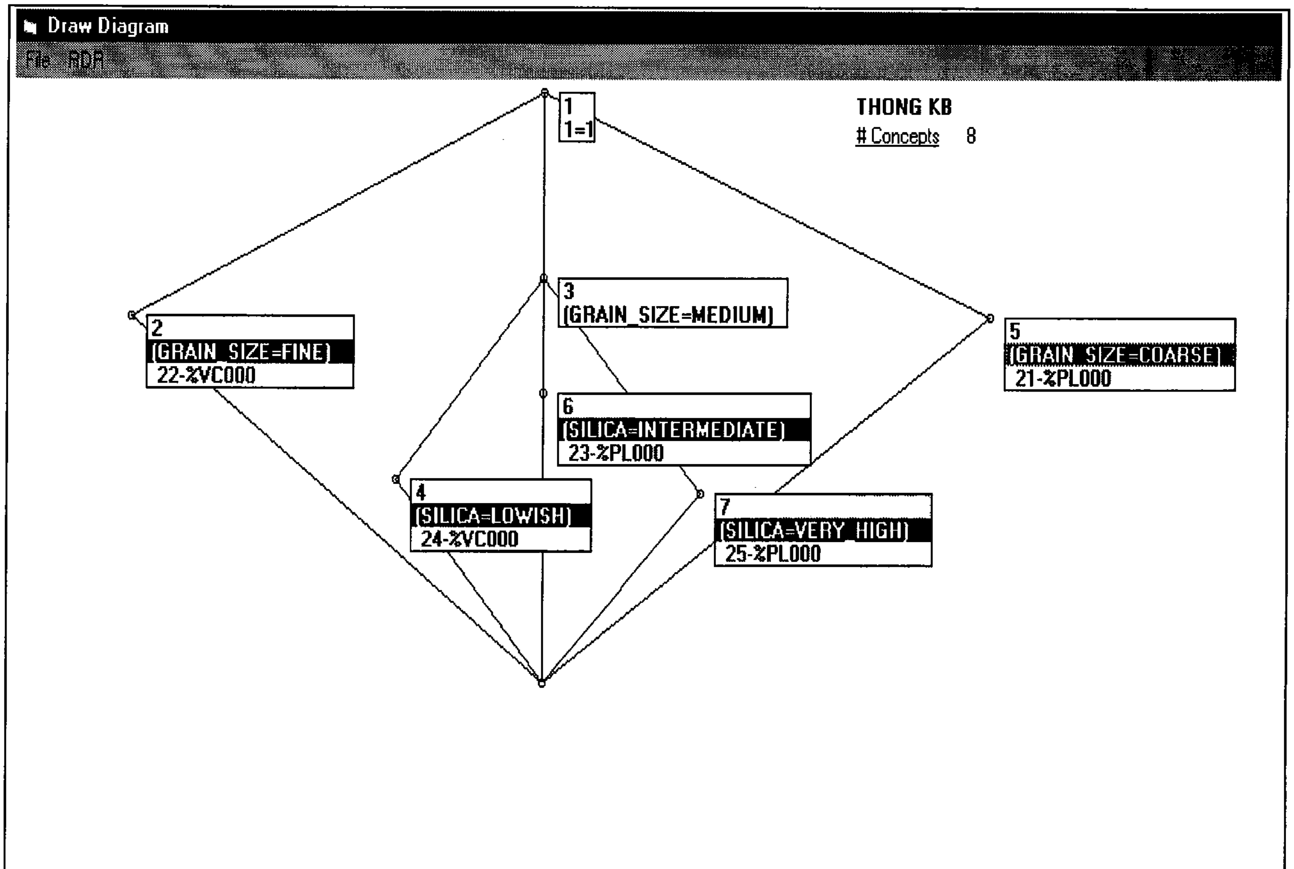

Figure 5: The Hasse Diagram for the PLUTONIC (%PLOOO) and VOLCANIC (%VCOOO) conclusions using MCRDR/FCA for Windows

From our set of ordered concepts we compute the predecessors and successors of each concept so that we can draw the *Hasse* diagram. Predecessors were determined by finding the largest subconcept of the intents for each concept. Successors were determined by finding the smallest superconcept of the intents. The successor list was used to identify concepts higher in the diagram, the parents, and the predecessor list identified concepts lower in the diagram, the children. There are many ways that a line diagram can be drawn. It may be desirable to provide a number of different layouts because concepts can be viewed and examined in different ways depending on their purpose and meaning (Wille 1992). In addition, the user has the ability to move a node anywhere they like providing the node is not moved higher than any of its parents or lower than any of its children.

So far we have used this system on three different domains. The first was a 60-rule Blood Gases KBS, known as 105, that had been developed from the cornerstone cases associated with the 2000+ PEIRS rules. The second domain was known as LOTUS and concerned the adaptation and management of the *Lotus Uliginosis cv* Grasslands Maku for pastures in the Australian state of New South Wales (Hochman, Compton, Blumenthal and Preston 1996). The knowledge was recorded into four KBS by four independent agricultural advisors. We used the concept matrices and line diagrams to compare the conceptual models of the advisors. This technique was seen as a useful way to identify and reconcile any differences as well as identify the main concepts associated with this domain.

The third domain also involved knowledge from multiple experts. Initially we have used the tools to understand the key concepts of the domain. We can see in Figure 5 that when GRAIN-SIZE = COURSE a rock is plutonic (%PL000) and if GRAIN-SIZE = FINE a rock is VOLCANIC (%VC000). However, when GRAIN-SIZE = MEDIUM then if SILICA = LOWISH it is a volcanic rock otherwise if SILICA = VERY-HIGH or INTERMEDIATE it is a plutonic rock. The line diagram has shown us what attribute-value pairs are the critical ones for these conclusions.

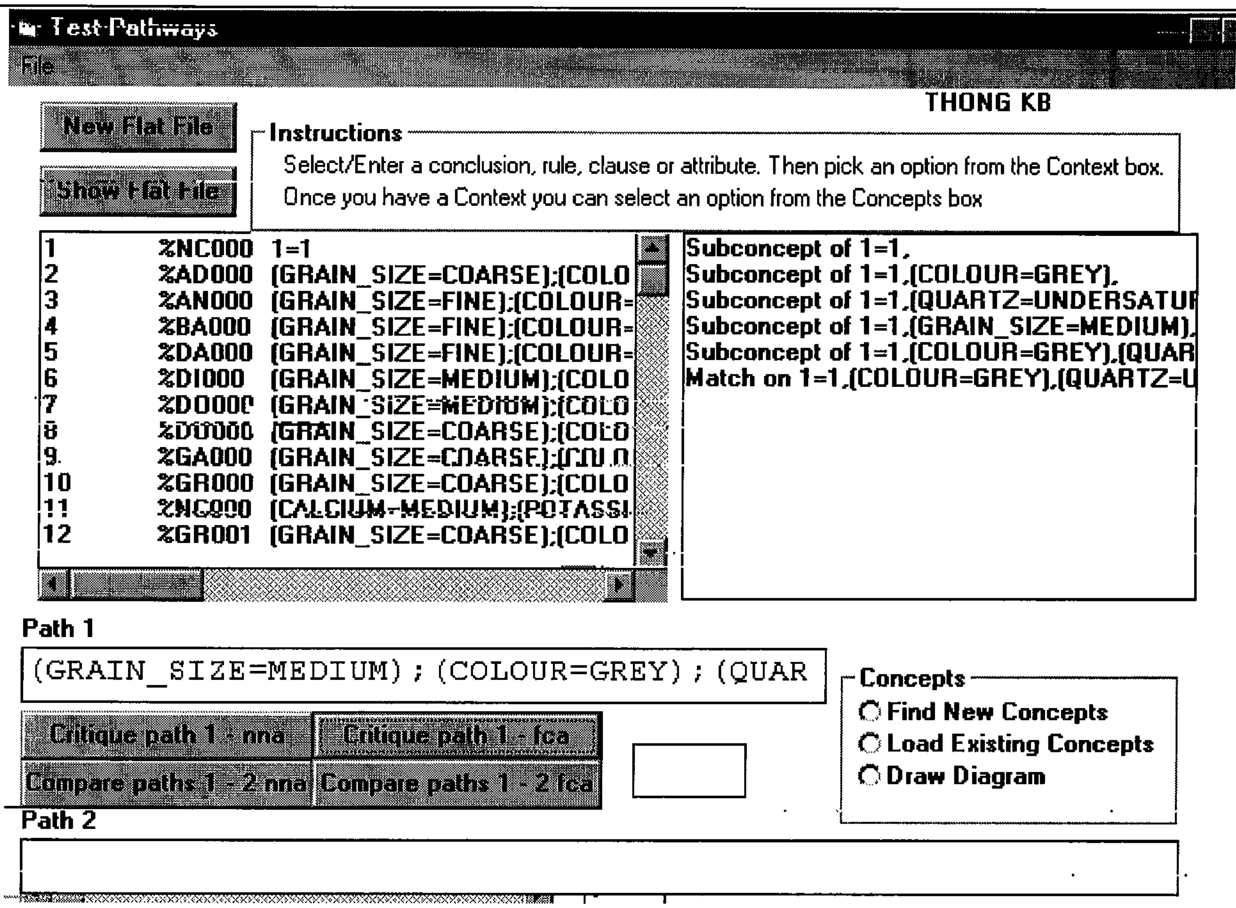

Figure 6: The user can compare pathways in the knowledge base using FCA and a nearest neighbour algorithm

We have also sought to the use the concepts for assisting the user with KA/validation. In addition to our work on critiquing conclusions, mentioned in section 2, we want to assist the user in forming the rule. Figure 6 shows how rule pathways can be compared. When the user selects the conditions for the rule the user is presented with a listbox of all the other pathways in the knowledge base that are matches, sub or superconcepts. For this purpose we are only using the intensional definition of the concepts derived using Wille'e technique. This is because an intensional definition implies an extensional definition but the converse is possibly but not necessarily true (Zalta 1988). Thus the extensional definition was too restrictive. The purpose of showing the user this information is to give them an understanding of how the new rule fits in with the existing knowledge. If the new rule is identified with concepts that seem inappropriate this is a warning to the user that the knowledge in the new rule or an existing rule is incorrect.

5 Future Directions

We can see from the examples that the ability to find and compare concepts in our knowledge base is useful for validation purposes. Such an approach goes beyond verification but attempts to identify that the knowledge is an accurate representation of the expert's knowledge. As a next steps we want to improve the user friendliness of our screens and test our approach using real experts, seeing if giving the user more understanding of the knowledge already captured assists them when adding new rules.

We have also demonstrated that the incorporation of FCA into RDR allows models to be found and compared without the need for prior understanding or explication of that model. This is particularly useful in domains where knowledge is emerging or in the common situation where it is difficult for experts to describe how they arrive at a conclusion. We see that KA using RDR offers a more realistic and reachable goal than approaches that depend on the user to predefine a model.

We are not the first to implement the ideas of FCA. Software, such ConImp (Burmeister 1996) and Toscana (Vogt and Wille 1995), have been available for many years. The main difference is that rather than using a Formal Context as the starting point we are trying to reduce the need for specification of the essential elements at the start and allow rules to be acquired and validated on-line. The rules in an RDR KBS lend themselves well to conversion to a formal context because each rule represents a rule pathway, which corresponds to a row in the formal context. We let the user select, from a variety of views of the knowledge, what parts of the KB should be included in the Formal Context.

While these preliminary results appear promising there is still much more work to be done. As mentioned in section 4, the formulation of concept lattices from many-valued contexts requires their interpretation into a formal context. While we have taken a simple approach by treating a clause as an attribute, currently a rule that should be part of the context for a selected focus of attention may be missed if the clause does not match on a conclusion or attribute already selected. The use of different *conceptual scales* (Ganter and Wille 1989) may provide a solution and needs further investigation. Some work has been done using a distance-weighted nearest neighbour algorithm to assign a score to clauses to find if clauses are related at all and to what extent. This approach has been implemented in Figure 6 where a nearest neighbour algorithm is used to assign a score of relative closeness of one pathway to others. It may also be possible to incorporate these techniques in determining which rules should be added to a context.

We continue to look at methods for discovering relationships in the knowledge base. Some work (Richards, Gambetta and Compton 1996) has already been done on the use of rough sets for this purpose and for verification of KBS. A comparison will be made between the dependencies and concepts generated using FCA and those found in the cores and reducts computed using rough sets. Other investigations include: a comparison of concept lattices to concept maps (Gaines and Shaw 1995); the use of *attribute exploration* for acquisition of formal contexts (Wille 1989b) and review of work which combined repertory grids and FCA (Spangenbery and Wolff 1988). Of particular interest to us, although not discussed in this paper, is the usefulness of these techniques to support the reuse of knowledge in a wide range of modes, such as explanation, tutoring or 'what-if' analysis, and this is currently under investigation.

Acknowledgements

RDR research is supported by Australian Research Council grants. The authors would like to thank Phil Preston for being a sounding board for different implementation issues and for the MCRDR software on which this is based and thanks to Rudolf Wille who inspired the search and provided the foundations for finding concepts.

References

Burmeister, P. (1996) Formal Concept Analysis with ConImp: Introduction to the Basic Features A translation of ConImp -Ein Programm zur Formalen Begriffanalyse In G.Stumme, R.Wille (eds.) *Begriffliche Wissenerarbeitung:Methoden und Anwendungen*, Springer Verlag.

Chandrasekaran, B. (1986) Generic Tasks in Knowledge-Based Reasoning: High Level Building Blocks for Expert System Design IEEE Expert pp: 23-30. Fall 1986.

Clancey, W.J., (1991b) Book Review - Israel Rosenfield, The Invention of Memory: A New View of the Brain *Artificial Intelligence*, 1991. **50**: p.241-284.

Compton, P. and Jansen, R., (1990) A Philosophical Basis for Knowledge Acquisition. *Knowledge Acquisition* 2:2

Compton, P., Preston, P. and Kang, B. (1995) The Use of Simulated Experts in Evaluating Knowledge Acquisition, *Proceedings 9th Banff Knowledge Acquisition for Knowledge Based Systems Workshop* Banff. Feb 26 - March 3 1995.

Cragun, B.J. and Steudel, H.J. (1987) A Decision-Table-Based Processor for Checking Completeness and Consistency in Rule-Based Expert Systems Int. Journal of Man-Machine Studies **26**(5):633-648.

Edwards, G., Compton, P., Malor, R, Srinivasan, A. and Lazarus, L. (1993) PEIRS: a Pathologist Maintained Expert System for the Interpretation of Chemical Pathology Reports *Pathology 25*: 27-34.

Gaines, B. R. and Shaw, M.L.G. (1989) Comparing the Conceptual Systems of Experts *The 11th International Joint Conference on Artificial Intelligence* :633-638.

Gaines, B. R. and Shaw, M.L.G. (1995) Collaboration through Concept Maps CSCL'95 Proceedings September 1995.

Ganter, B. (1988) Composition and Decomposition of Data *In Classification and Related Methods of Data Analysis* (Ed. H. Bock) pp:561-566, North-Holland, Amsterdam.

Ganter, B. and Wille, R. (1989) Conceptual Scaling In *Applications of Combinatorics and Graph Theory to the*

Biological Sciences (Ed. F. Roberts) pp:139-167, Springer, New York.

Grossner, C., Preece, A.D., Chandler, P.G., Radhakrishnan, T. and Suen, C.Y. (1993) Exploring the Structure of Rule Based Systems *Proceedings of the Eleventh National Conference on Artificial Intelligence,* pp704-709, Washington DC, MIT Press.

Guha, T.V., and Lenat, D.B. (1990) CYC:A Mid-Term Report *AI Magazine* **11**(3):32-59.

Hochman, Z., Compton, P., Blumenthal, M. and Preston, P. (1996) Ripple-down rules: a potential tool for documenting agricultural knowledge as it emerges. p313-316. In *Proc. 8th Aust. Agronomy Conf.*, Toowoomba 1996.

Kang, B., Compton, P. and Preston, P (1995) Multiple Classification Ripple Down Rules: Evaluation and Possibilities *Proceedings 9th Banff Knowledge Acquisition for Knowledge Based Systems Workshop* Banff. Feb 26 - March 3 1995, Vol 1: 17.1-17.20.

Kang, B., Gambetta, W. and Compton, P. (1996) Verification and Validation with Ripple-Down Rules *Int. J. Human-Computer Studies* **44** :257-269

McDermott, J. (1988) Preliminary Steps Toward a Taxonomy of Problem-Solving Methods *Automating Knowledge Acquisition for Expert Systems* Marcus, S (ed.) Kluwer Academic Publishers, pp: 225-256.

Menzies, T.J. and Compton, P. (1995) The (Extensive) Implications of Evaluation on theDevelopment of Knowledge-Based Systems in *Proceedings 9th Banff Knowledge Acquisition for Knowledge Based Systems Workshop* Banff. Feb 26 - March 3 1995,.

Newell, A. (1982) The Knowledge Level *Artificial Intelligence* 18:87-127.

O'Keefe, R.M. and O'Leary, E.E. (1993) Expert System Verification and Validation: A Survey and Tutorial *Artificial Intelligence Review* **66** 273-309.

Patil, R. S., Fikes, R. E., Patel-Schneider, P. F., McKay, D., Finin, T., Gruber, T. R. and Neches, R., (1992) The DARPA Knowledge Sharing Effort: Progress Report In C. Rich, B. Nebel and Swartout, W., *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference* Cambridge, MA, Morgan Kaufman.

Pirlein, T and Struder, R., (1994) KARO: An Integrated Environment for Reusing Ontologies *A Future for Knowledge Acquisition EKAW '94*, pp: 200-225, Springer Verlag

Preece, A.D., Shinghal, R and Batarekh, A. (1992) Principles and Practice in Verifying Rule-Based Systems *The Knowledge Engineering Review* 7:115-141.

Puerta, A. R, Egar, J.W., Tu, S.W. and Musen, M.A. (1992) A Mulitple Method Knowledge Acquisition Shell for Automatic Generation of Knowledge Acquisition Tools *Knowledge Acquisition 4*(2).

Richards, D ., Gambetta, W. and Compton, P (1996) Using Rough Set Theory to Verify Production Rules and Support Reuse *Proceedings of the Verification, Validation and Refinement of KBS Workshop, PRICAI'96* 26-30 August 1996, Cairns, Australia, Griffith University.

Richards, D and Compton, P (1996) Building Knowledge Based Systems that Match the Decision Situation Using Ripple Down Rules, *Intelligent Decision Support '96* 9th Sept, 1996, Monash University.

Richards, D., Chellen, V. and Compton, P (1996) The Reuse of Ripple Down Rule Knowledge Bases: Using Machine Learning to Remove Repetition Proceedings of Pacific Knowledge Acquisition Workshop PKAW'96, October 23-25 1996, Coogee, Australia.

Scheffer, T. (1996) Algebraic Foundation and Improved Methods of Induction of Ripple Down Rules *Proceedings of Pacific Knowledge Acquisition Workshop PKAW'96*, October 23-25 1996, Coogee, Australia.

Schreiber, G., Weilinga, B. and Breuker (eds) (1993) KADS: A Principles Approach to Knowledge-Based System Development *Knowledge-Based Systems* London, England, Academic Press.

Shadbolt, N., (1996) URL:http://www.psyc.nott.ac.uk/aigr/research/ka/SisIII

Solloway, E., Bachant, J. And Jensen, K. (1987) Assessing the Maintainability of XCON-in-RIME: Coping with Problems of a very Large Rule Base *Proceedings of the Sixth International Conference on Artificial Intelligence* Vol2:824-829, Seattle, WA: Morgan-Kauffman.

Spangenberg, N and Wolff, K.E. (1988) Conceptual Grid Evaluation In H.H. Bock ed. *Classification and Related Methods of Data Analysis* Elsevier Science Publishers B.V. North Holland.

Steels, L. (1993) The Componential Framework and Its Role in Reusability In David, J.M., Krivine, J.-P. and Simmons, R., editors *Second Generation Expert Systems* pp: 273-298. Springer, Berlin.

Suwa, M., Scott, A. and Shortliffe, E. (1982) An Approach to Verifying Completeness and Consistency in a Rule-Based Expert System *AI Magazine* 3(4):16-21.

Vogt, G. and Wille, R. (1995)TOSCANA- A Graphical Tool for Analysing and Exploring Data *Knowledge Org22(2):7881.*

Wille, R. (1982) Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts In *Ordered Sets* (Ed. Rival) pp:445-470, Reidel, Dordrecht, Boston.

Wille, R. (1989a) Lattices in Data Analysis: How to Draw them with a Computer In *Algorithms and Order* (Ed. I. Rival) pp:33-58, Kluwer, Dordrecht, Boston.
Wille, R. (1989b) Knowledge Acquisition by Methods of Formal Concept Analysis *In Data Analysis, Learning Symbolic and Numeric Knowledge* (Ed. E. Diday) pp:365-380, Nova Science Pub., New York.
Wille, R. (1992) Concept Lattices and Conceptual Knowledge Systems *Computers Math. Applic. (23)*6-9:493-515.
Wille, R. (1996) Conceptual Structures of Multicontexts *Conceptual Structures: Knowledge Representation as Interlingua* (Eds. P.Eklund, G. Ellis and G. Mann) pp:23-39, Springer.
Zalta, E.N. (1988) *Intensional Logic and the Metaphysics of Intentionality*, Cambridge, Massachusetts, MIT Press.