Uncovering the Conceptual Models in Ripple Down Rules

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: The need for analysis and modeling of knowledge has been espoused by many researchers as a prerequisite to building knowledge based systems (KBS). This approach has done little to alleviate the knowledge acquisition (KA) bottleneck or the maintenance problems associated with large KBS. For actual KA and maintenance we prefer to use a technique, known as ripple down rules (RDR) that is simple, yet reliable, and later see what models can be produced from the knowledge for the purpose of reuse. Tools based on Formal Concept Analysis have been added to RDR to uncover and explore the underlying conceptual structures.

1 Models and their Role in Knowledge Acquisition

Since Newell's [21] paper on "The Knowledge Level" there has been increasing awareness and acceptance of the need to model knowledge at a level above its symbolic representation. This notion was further explored by Clancey [3] who used task and problem solving methods analysis to divide problems into "heuristic classification" and "heuristic construction". Following Van de Velde [33] approaches which have been built on Newell's knowledge level model include:- Generic Task Framework [2], KADS and CommonKADS [30], Role-Limiting Methods [19], Components of Expertise and the Componential Methodology [32], Protege and Protege II [24], KIF and Ontolingua [22]. All of these approaches impose a particular structure on the knowledge to enable the current problem to be mapped into the appropriate class of problem situation. The structure chosen will depend on the features of the task and the domain of expertise. While each of the approaches mentioned above are different, Van de Velde [33, p.1218] considers three concepts to be generally included as part of a knowledge level model. These are the domain model, the task model and the problem solving method.

While the knowledge level approach is superior to the previous transfer of expertise approach to KA, matching the problem solving method or methods to the problem and adapting them to suit the domain is no mean feat [40]. The use of methodologies such as KADS requires extensive involvement of a knowledge engineer and the modeling process is complex, often doing little to alleviate the KA bottleneck or the problems associated with maintaining large KBS [20].

In the above approaches the purpose of modeling is to allow the capture of expertise in a structured and systematic way. Clancey defines a model as "merely an abstraction, a description and generator of *behaviour patterns over time*, not a mechanism equivalent to human capacity." [4, p. 89]. Since models are at best imperfect representations that vary between users and the same user over time [11] we prefer to use simple, yet reliable techniques for KA. However, we are interested in modeling as an end in itself due to their "explanatory value as psychological descriptions" [4, p.89] and their usefulness in instruction [30].

This paper considers the combination of two approaches, Formal Concept Analysis (FCA) [36] and Ripple Down Rules (RDR) [5], that reduce KA to tasks that can easily be performed by an expert with minimal *a priori* modeling or involvement of a knowledge engineer. In FCA the expert defines a context, which is a Object-Attribute crosstable such as the one shown in Figure 1. From the context, formal concepts are derived, which are ordered to provide a complete lattice of sub and super concepts. These concepts can be represented as a line diagram, also known as a *Hasse* diagram, and can be used to derive implications for use in a knowledge base [37]. This paper investigates starting from the opposite direction by using existing rules in an RDR KBS, acquired manually from an expert, to define contexts and then generate the concepts using FCA to make explicit the models implicit in the rules.

KA using RDR requires the expert to assign a conclusion to a case and then to select the feature/s in the case that were used to make that decision. Each time a new case is seen the expert checks the conclusion assigned by the system and if the expert does not agree a new conclusion is assigned and the features which differentiate the current misclassified case from the case associated with the rule that inappropriately fired are selected by the user. FCA requires some consideration of the whole domain as does Repertory Grids [11] and does not consider incremental maintenance. RDR on the other hand develops the whole system on a case by case basis and automatically structures the KBS in such a way to ensure changes are incremental. The classification of cases and identification of features is very simple and probably less demanding for experts than the development of crosstables.

	Has wings	flys	suckles young	warm-blooded	cold-blooded	breeds in water	breeds on land	has scales
Bird	X	X			X		X	
Reptile					X		X	X
Amphibian					X	X		
Mammal			X	X			X	
Fish					X	X		X

Fig. 1. Context of "Vertebrates of the Animal Kingdom"

FCA and RDR both see that KA should be primarily a task performed by the expert. RDR is founded on the realisation that experts do not offer explanations of why they made a decision, rather they offer a justification and that justification will depend on the situation [5]. Given the inconsistency that exists in users' perceptions, the effort involved in developing the models required by many of the KL-Model approaches mentioned may be pointless. Using FCA and RDR reduces modeling to the tasks of classifying objects (cases) and the identification of the salient features. Interestingly, they approach classification from alternative perspectives. FCA is concerned with finding the similarity between objects, the conjunction of sets of attributes. RDR looks at differences between cases (objects) and is conceptually close to research using repertory grids [11] which is based on Personal Construct Psychology (PCP) [17] and the use of a discernability matrix in Rough Sets [23].

Another view held by FCA and RDR is the importance of context on affecting human behaviour and its impact on the concepts (or rules) formulated and their interpretation [5], [38]. The formalisation of context by FCA, the RDR approach to context and the merging of these approaches are described later in the paper.

While KA and maintenance are easily performed by the expert without an KE [7] it was found that RDR did not easily lend itself to certain modes of use where a model of the knowledge was one of the purposes of building the KBS, such as explanation, *what-if* analysis and critiquing [26]. Other work has looked at deriving causal models from heuristic systems for diagrams and concluded that extra knowledge was needed to provide causal explanation [18]. The explanation provided by the RDR rule pathways has been shown [6] to be superior to a conventional rule trace because the exception structure shows how the knowledge has evolved, why a case has both succeeded and failed and what alternative pathways are possible. However, explanation often required the presentation of a model of the concepts that a rule or collection of rules represents. For the purpose of critiquing it was necessary to determine how close one rule pathway was to another to see if the user's conclusion was within acceptable limits of the system's conclusion or how the two differ.

The issue of the reuse of knowledge and the different ways in which a user may want to use and view the knowledge, prompted the investigation of a technique that would allow the relationships between rules and conclusions to be found. FCA has been shown to be a robust method of finding, ordering and displaying formal concepts [37]. Since the original submission of this paper we have been looking at the extensional definition of the subsumption relationship as adopted by Wille. As Zalta [39] points out intensional subsumption implies extensional subsumption but the converse is possibly but not necessarily true. As a result, in further work we have conducted on comparing concepts (in the form of pathways, new rules, conclusions, etc) we have only been using the intensional definition of concepts to determine the closeness of the concepts to other and to identify sub and superconcepts in the hierarchy. We continue to use Wille's method for finding concepts and drawing line diagrams even though we ignore the extensional definition of concepts for many of our purposes because they were too restrictive. This latter work is still in progress and is not described in this paper. This paper concentrates on what we have done using FCA. We first briefly describe FCA and RDR individually and then describe how FCA tools have been added to RDR. A small case study is included to demonstrate the possible benefits of combining these techniques for comparing the conceptual models of multiple experts.

2 Formal Concept Analysis

Formal Concept Analysis, first developed by Wille [34], 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" (37, p.493). A set of objects and their attributes, known as the extension and intension respectively, constitute a formal context which may be used to derive a set of ordered concepts. The following description of FCA follows Wille [34].

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 1 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 $g \in G$ and then finding all their intersections. Alternatively Y' can be obtained by determining all column-intents $\{m\}'$ with $m \in M$ 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').

Having found the concepts it is necessary to find the subconcept-superconcept relation between concepts so that they may be ordered and represented as a labelled line diagram. 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:

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

From Lattice Theory, we are able to form a complete lattice, called a concept lattice and denoted $\mathfrak{B}(\mathbb{K})$, 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)” [37, p.497]. The line diagram in Figure 6 is our implementation, known as MCRDR/FCA, which involved enhancing a current MCRDR for Windows system using Visual Basic. The concepts are shown as small boxes and the sub/superconcept relations as lines. Each concept has various intents and extents associated with it. In MCRDR\FCA it is possible to display the concept number, attribute/s or object/s belonging to each node or all three dimensions can be displayed concurrently, as in Figure 6. It is also possible to click on an individual node to see the concept number and all of its extents and intents. The labeling provided has been reduced for clarity. All intents of a concept δ are reached by ascending paths from δ and all extents are reached by descending paths from the concept δ.

3 Ripple Down Rules

Ripple down rules were proposed in answer to the problem of maintaining a large clinical pathology KBS [5]. To avoid the problem of side-effects that occur when maintaining a typical production rule KBS, 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. In single classification RDR, we define an RDR as a triple <rule,X,N>, where X are the exception rules and N are the if-not rules [28], see Figure 2(a). 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) [16] 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. Figure 2(b) shows an example of an MCRDR. MCRDR was chosen for this study 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 [27] does not exist.

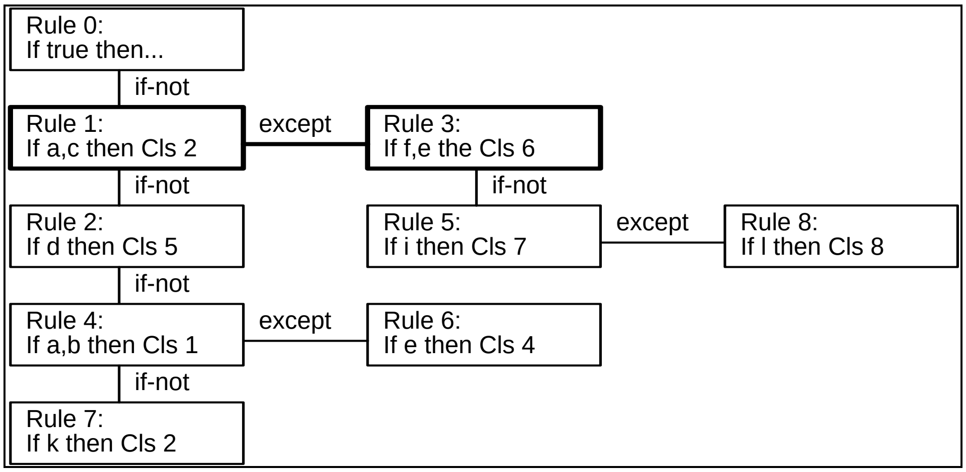

Fig. 2(a). A single classification RDR KBS. The highlighted boxes represent rules that are satisfied for the case {a,c,d,e,f,h,k}

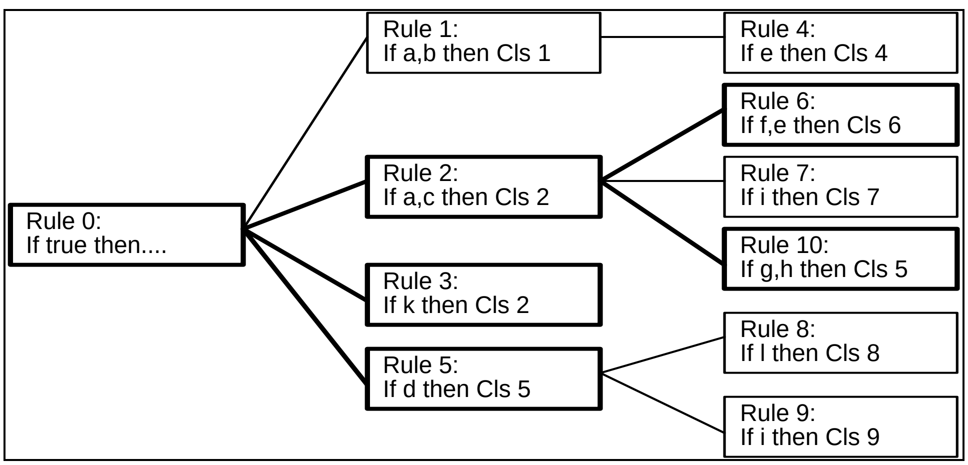

Fig. 2(b). An MCRDR KBS. The highlighted boxes represent rules that are satisfied for the case {a,c,d,e,f,h,k}

Path 1 [(Rule 0,…), (Rule 2, Class 2), (**Rule 6, Class 6**)]
Path 2 [(Rule 0,…), (Rule 2, Class 2), (**Rule 10, Class 5**)]
Path 3 [(Rule 0,…), (**Rule 3, Class 2**)]
Path 4 [(Rule 0,…), (**Rule 5, Class 5**)]

Fig. 2(c). Each of the rules in Fig. 3(a) and 3(b) represent a pathway through the knowledge base. Four pathways from 3(b) are shown above. The rules that fire and produce conclusions from the case {a,c,d,e,f,h,k}are highlighted. (Fig. 2(b) and (c) are taken from [16] .

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. Thus the 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 that this is efficient [16]. The utility of RDR has been demonstrated by Pathology Expert Interpretive Reporting System (PEIRS) [10] which went into routine use in a large Sydney hospital with about 200 rules and over a four year period (1990-1994) grew to over 2000 rules. The compactness and efficiency of both RDR and MCRDR have been demonstrated in simulation studies [16].

4 Combining FCA and RDR

As discussed in Section One, certain modes of use required the ability to understand the underlying relationships and models inherent in the RDR rules. The use of Formal Concept Analysis appeared to be a means of discovering these models. 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. The screen in Figures 3 and 4 use a 60-rule Blood Gases KBS, known as 105, that had been developed from the cornerstone cases associated with the 2000+ PEIRS rules[1].

The first step was to use the rules to generate a context. The RDR KBS was converted to a flat structure by traversing the list structure for each rule 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. Such a global view is only feasible for small, if not very small, KBS. As with any graphical representation, as the number of rules being modeled grew, the line diagram became too cluttered to be comprehensible.

[1] It would have been interesting to use the 2000+ PEIRS rules, but this was not done at this time because they are not in the MCRDR format used by this study.

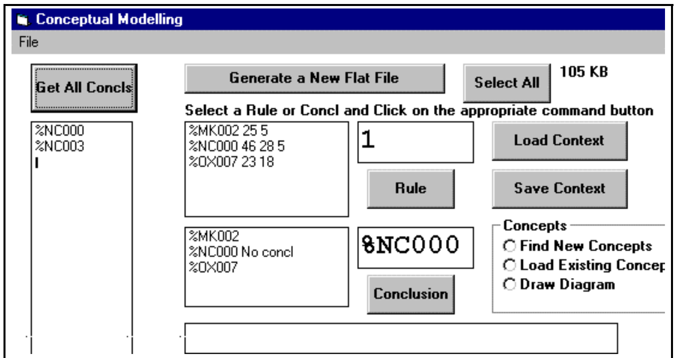

Fig. 3. The Selection Screen in MCRDR/FCA which allows user to choose the focus of attention, either a rule or a conclusion

This was the case with the Blood Gases KBS. Therefore, to limit the concepts to a manageable size that could be understood by looking at a matrix or a line diagram the user was asked to narrow their focus of attention to a particular rule or conclusion. The decomposition of a concept lattice into smaller parts is a strategy that has previously been found useful [35]. Our approach is similar to that proposed by Ganter [13] where the context is shortened to find subcontexts and subrelations.

There are currently 13 different ways a context may be derived. The two main methods are choosing a conclusion or a rule, see Figure 3. If a conclusion was chosen, all rules using that conclusion were selected and added as objects to the set G, forming the extents of the context. As each extent was added the clauses of the rules were added to the set M of attributes to form the intents of the context, first checking to see if any attributes had 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 chose a particular rule then that rule was added as the first object with the rule clauses as the initial intension. Every clause in each rule in the flattened RDR rule base was searched for a match on the initial set of attributes. If a match was found, that rule was added to the extension and all new attributes (clauses) found in the matching rule were also added to the intension.

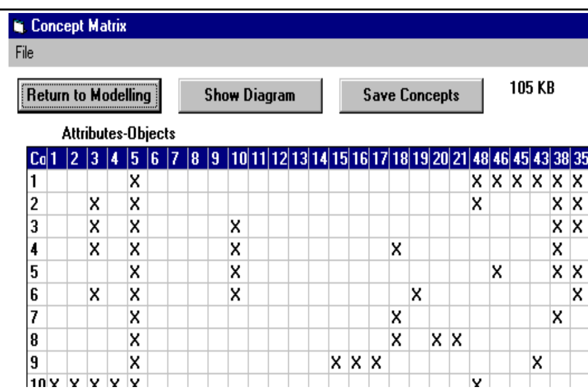

Fig. 4. The Concept Matrix Screen from MCRDR. Here the user is shown the concepts generated from the context. The context used is the NC000 conclusion for the Blood Gases domain

Treating the rule clause, which is an attribute-value pair, as a boolean or condition attribute, is similar to the technique known as *conceptual scaling* [14] 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). Essentially, 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.

```
For each concept
    Find parent concepts from successor list
    Find children concepts from predecessor list
End For
Locate top concept in fixed position (top and centre)
Get x-factor - find how many branchings in the predecessor list in the second column
Get y-factor - find how many branchings in the successor list in the second column
For all children of the top concept
   position one y-factor lower than the parent  and one x-factor to the right of the previous sibling  remove
      any object labelling from the parent that is found in the child
End For
For all children of the top concept
    Add-children(child)
End For
Add the last concept in fixed position
Reduce attribute labelling

Add-children(parent)   (note the child passed to this procedure is now the parent)
Find the children (parent)
For all children
    If      child is the last concept or has already been located
    Then  process next child
    Else   find out how many parents it has
             If      only one parent
          Then  If      child is an only child (its parent has one child only)
                   Then  locate the child directly below the parent
                   Else   x-coord = (Parent x-coord - ((NoChildren of Parent * x-factor)/2)
                                         + ((number of child being processed-1)*x-factor)
                   End if
          Else   locate the child midway between left and right most parents on the x-axis
                   and one y-factor lower than the lowest parent
          End If
          If the parent contains some of the children's objects
          Then  remove
          End If
          If      coordinates used
          Then  until a free position is found change x-coord moving first left and then right by half x-factor
          End If
          Flag as located
          Add_children (child)
    End If
```

Fig. 5. The algorithm used to find the concept coordinates for the Hasse diagram

The crosstable generated in the above process was then used to construct all formal concepts of the formal context, using the process described in Section 2. Appropriate ordering of concepts is difficult as a given concept may be a subconcept of different superconcepts. This can be seen in the concept lattice in Figure 4 where there are a number of sets of concepts. Concept ordering can improve the matrix aesthetically. To allow drawing of the *Hasse* diagram it was necessary to compute the predecessors and successors of each concept. Predecessors were found by finding the largest subconcept of the intents for each concept. Successors were found 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. As Wille [37] points out, there is not one fixed way of drawing line diagrams and often a number of different layouts should be used because concepts can be viewed and examined in different ways depending on their purpose and meaning. An algorithm for the graph layout is described in Figure 5. If the line diagram drawn by the system is not to the liking of the user, they may 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.

5 A Case Study that uses of RDR for KA and FCA for Modeling

A small case study has been performed to demonstrate how FCA can be used for modeling knowledge acquired using RDR. The domain chosen concerns the adaptation and management of the *Lotus Uliginosis cv* Grasslands Maku for pastures in the Australian state of New South Wales. The knowledge was recorded by advisors who were representing local groups of farmers and agribusiness people involved in "Co-learning" about Lotus. Four advisors independently added rules to single classification RDR systems. The domain will be referred to as Lotus and the four KBS will be known as Lotus 1, 2, 3 and 4. Knowledge in the domain is evolving and RDR is being used to accelerate and consolidate knowledge development in the field into a corpus of knowledge about the crop [15]. Such a domain seems eminently suitable for this study since it is small enough to be comprehensible and there is a need to discover the concepts and models that exist but are still emerging.

In order to model each of the KBS it was necessary to reenter the rules into MCRDR. The KBS from the farmers had been developed using XRDR for Windows, a single classification RDR implementation. As stated previously, MCRDR had been chosen as the starting point for adding FCA as there are no false branches which may be difficult to interpret and since many domains, such as Lotus, contain cases that belong to multiple independent classes. It would have been preferable for the individual KBS to have been captured in MCRDR format, since it is always difficult for a non-expert to know what conclusions to assign. While it was easy to assign the classification found in the XRDR version, it was not easy to know which other conclusions may also have been applicable or whether a conclusion should have been stopped. Another problem for a non-expert is the selection of features which justify the conclusion being added or deleted. Given these limitations the knowledge reentered may not be a completely accurate picture of the knowledge entered by the advisor but adequate for the purposes of this case study. Once in MCRDR format, a formal context of each complete KBS was generated and the concepts derived using the processed described in the previous section.

There are two main tools to aid in the comparison of conceptual models. The first is the concept matrix. For space we do not show the four line diagrams associated with each KBS, but we state that we were able to see that each KBS shared a number of concepts and that different concepts could also be identified. By looking at the matrix the farmers are able to see not only what attributes (intents) and conclusions (extents) others consider important but also the relationships between them and how it affects other conclusions.

The second tool for modeling is the line diagram. The diagrams showing the concept labeling only are generally well laid out with minimal line crossing, using the algorithm described in Figure 5. However, in order to understand the models it is necessary to provide complete labeling showing the concept, extents and intents associated with each node. To save space, only the Lotus1 and Lotus4 KBS are shown in Figure 6. The nodes in both diagrams have been moved to fix some overlap of labels and improve readability.

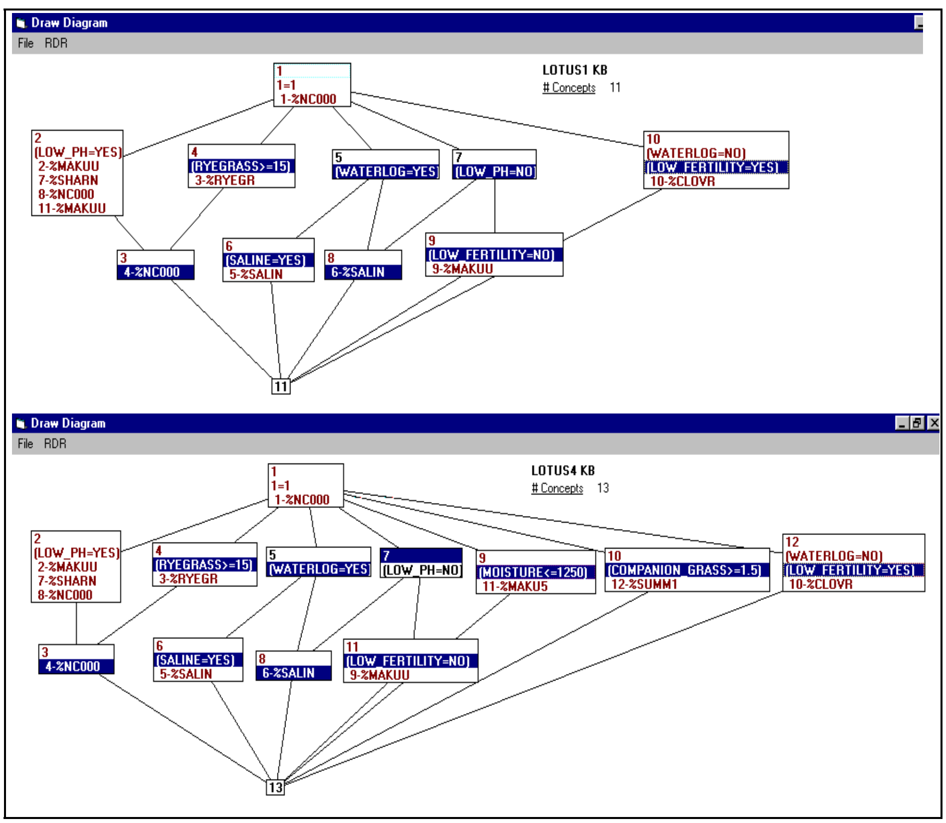

Fig. 6. The line diagrams for Lotus 1 and 4 KBS in MCRDR/FCA

The line diagram provides a more hierarchical understanding of the sub and super relationships in the domain. At a glance we can see that eight concepts are shared by the two KB and that the main difference between the two KBS are the concepts If (MOISTURE ≤ 1250) then the conclusion is %MAKU5 - *Maku OK at rainfalls above 1250mm* and If (COMPANION_GRASS ≥ 1.5) then the conclusion is %SUMM1- *Grass competition is excessive for lotus-keep grass down in summer* which are contained in concepts 9 and 10, respectively, in Lotus4. The diagrams also show that even though the knowledge has been structured slightly differently both farmers consider that when (LOW_PH = YES) and (RYEGRASS ≥ 15) the conclusion should be %NC000 No Conclusion.

To facilitate comparison it was important to ensure the attributes shared by all contexts were in the same order before the concepts were determined. In the case study all rules were used for comparison since the KB size was small enough but it was also possible to choose a particular attribute (rule-clause), rule or conclusion to focus on. It would be useful to view the whole KB to identify variations and then reduce the context by selecting the rules or conclusions that differed.

6 Discussion

The case study has shown 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.

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 this conversion has been relatively straightforward and sufficed for our case study, 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 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. It may be possible to incorporate these techniques in determining which rules should be added to a context.

The usefulness of FCA to support the reuse of knowledge in a wide range of modes, such as explanation or tutoring, is under investigation. We have begun new work that uses the concepts developed to assist the user to understand how a selected concept, which could be a new rule or conclusion, fits in relation to other concepts in the KBS. This is useful for KA, explanation and critiquing. As mentioned in Section 1, for that work we are only using the intensional definition of the concepts derived using

Wille's techniques because of the limitations of the extensional definition [39] and because it is too restrictive in the finding of sub and superconcepts.

The work presented in this paper has simply added to RDR the ideas in FCA that have already been available in other software, such as ConImp and Toscana. The main difference is found in the use of rules to generate contexts. RDR lends itself well to conversion to a formal context because each rule represents a rule pathway. Some work [26] has already been done on the use of rough sets to find relationships in the knowledge base and 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 [12]; the use of *attribute exploration* for acquisition of formal contexts [36] and review of work done on combining the use of repertory grids and FCA [31].

Acknowledgments

RDR research is supported by Australian Research Council grants. The authors would like to thank Zvi Hochman for providing the LOTUS KBS and related information, to Phil Preston for the MCRDR and the Hierarchy software and to Rudolf Wille for the inspiration of this work.

References

[1] 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.

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

[3] Clancey, W.J., (1985) Heuristic Classification *Artificial Intelligence***,** 1985. **27**: p.289-350.

[4] Clancey, W.J., (1993) Situated Action: A Neurological Interpretation Response to Vera and Simon *Cognitive Science*, **17**: pp.87-116.

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

[6] Compton, P., Edwards, G., Kang, B., Lazarus, L., Malor, R., Menzies, T., Preston, P., Srinivasan, A. and Sammut, C. (1991) Ripple Down Rules: Possibilities and Limitations 6th Banff AAAI Knowledge Acquisition for Knowledge Based Systems Workshop, Banff (1991) 6.1 - 6.18.

[7] Compton, P., Edwards, G., Kang, B., Lazarus, L., Malor, R., Preston, P., and Srinivasan, A. (1992*).* Ripple down rules: Turning knowledge acquisition into knowledge maintainance *Artificial knowledge in Medicine 4* pp:463-475.

[8] 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.

[9] Eades, P. (1996) Graph Drawing Methods *Conceptual Structures: Knowledge Representation as Interlingua* (Eds. P.Eklund, G. Ellis and G. Mann) pp:40-49, Springer.

[10] 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.

[11] 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.
[12] Gaines, B. R. and Shaw, M.L.G. (1995) Collaboration through Concept Maps CSCL'95 Proceedings September 1995.
[13] 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.
[14] 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.
[15] 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.
[16] 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.
[17] Kelly, G.A, (1955) *The Psychology of Personal Constructs* New York, Norton.
[18] Lee, M. and Compton, P. (1995) From Heuristic Knowledge to Causal Explanations *Proc. of Eighth Aust. Joint Conf. on Artificial Intelligence AI'95***,** Ed X. Yao, 13-17 November 1995, Canberra, World Scientific, pp:83-90.
[19] 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.
[20] 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,.
[21] Newell, A. (1982) The Knowledge Level *Artificial Intelligence* 18:87-127.
[22] 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.
[23] Pawlak, Zdzislaw (1982) Rough *Sets International Journal of Information and Computer Sciences, 11,* pp:341-356.
[24] 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).
[25] 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.
[26] 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.
[27] 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.
[28] 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.
[29] Schon, D.A. (1987) *Educating the Reflective Practioner* San Francisco: Jossey-Bass.
[30] Schreiber, G., Weilinga, B. and Breuker (eds) (1993) KADS: A Principles Approach to Knowledge-Based System Development *Knowledge-Based Systems* London, England, Academic Press.

[31] 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.

[32] 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.

[33] Van de Velde, W. (1993) Issues in Knowledge Level Modeling In David, J.M., Krivine, J.-P. and Simmons, R., editors *Second Generation Expert Systems* pp: 211-231. Springer, Berlin.

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

[35] 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.

[36] 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.

[37] Wille, R. (1992) Concept Lattices and Conceptual Knowledge Systems *Computers Math. Applic. (23)*6-9:493-515.

[38] 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.

[39] Zalta, E.N. (1988) *Intensional Logic and the Metaphysics of Intentionality*, Cambridge, Massachusetts, MIT Press.

[40] Zdrahal, Z and Motta, E. (1995) An In-Depth Analysis of Propose and Revise Problem Solving Methods *9th Knowledge Acquisition for Knowledge Based Systems Workshop* Banff, Canada, SRDG Publications, Departments of Computer Science, University of Calgary, Calgary, Canada pp:38.1-38.20.