Chapter 5

5 Building Conceptual Hierarchies from MCRDR[1] KBS

Section 1.2 provided a situated view of knowledge which was used to support the argument that models change and are difficult to capture. The preferred solution was to capture simple models based on an experts behaviour that only involved the capture of primitive notions using combinations of attribute-value pairs. This primitive knowledge supported the reflexive modes of KA, maintenance and inferencing in RDR. However as pointed out in Section 3.4.2 the lack of higher-level modelling features in RDR is a limitation due to the benefits that abstraction hierarchies can provide particularly for explanation and instruction. It was desirable to perform reflective reasoning on our reflexive, behavioural knowledge. Shastri notes that:

> *"when semantic networks are realised as massively parallel networks, they may provide an appropriate framework for modelling reflexive reasoning - reasoning that can be performed rapidly, effortlessly, and without conscious effort" (Shastri 1991, p.3)*

The use of a semantic network representation appeared to offer the type of functionality that was necessary. The line diagram or concept lattice developed by FCA is a semantic network that shows terminological subsumption and thus provides the hierarchical structure and the relationships between concepts that were required. Subsumption is defined by Wood as: "a formal relationship between pairs of conceptual descriptions that allow them to be ordered into a taxonomy on the basis of generality" (Woods 1991, p. 67). Woods defines five types of subsumption: extensional, structural, recorded, axiomatic and deduced subsumption. The concept matrix supports the first four types of subsumption. Deduced subsumption involves use of something like a general-purpose deduction component in conjunction with the subsumption hierarchy and is not a feature of MCRDR/FCA. The RDR KBS provided a classification system but not a subsumption system. Woods (1991) describes classification as an operation on a system of concepts whereas subsumption is an operation on a pair of concepts. By combining MCRDR and FCA classification and subsumption can be supported.

[1] Note the approach developed in this chapter is only applicable to MCRDR KBS because its exception structure only contains true branches which may readily be converted into a flat structure of rule pathways that maps directly into the crosstables used by FCA. The false branches in single classification are ambiguous and more difficult to convert into a flat structure.

In addition to abstraction, context is a major factor which will affect the appropriateness of the expert's model, particularly for reuse. The constantly changing nature of contexts (Rosenfield 1988) means that it is necessary to provide "continuity" or applicability of a context to a new situation by consideration of the relations involved (Lave 1988, p.20). Agre (1993) also identifies relationships between categories as a critical issue in the use of categories for locating things. When applied to a KBS these categories can be groups of rules or conclusions. Each rule can be seen as a low level concept that specifies the conditions under which a conclusion applies. It was the goal of this study to use the primitive concepts in the form of individual rules to find the higher level concepts and the relationships between them from which a conceptual hierarchy could be developed. To this end, this chapter describes how rules captured into an assertional MCRDR KBS can be used to develop a terminological KBS. The mechanism for this transformation is formal concept analysis which is discussed next. The screen dumps shown in this chapter, and most of this thesis, are from my implementation, called MCRDR/FCA, which is an enhancement of the current MCRDR for Windows system.

5.1 Formal Concept Analysis[2]

Formal Concept Analysis, first developed by Wille (1982), is a mathematically based method of finding, ordering and displaying formal concepts (Wille 1992). FCA 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" (Wille 1992, p. 493).*

In keeping with this, a concept in FCA is comprised of a set of objects and the set of attributes associated with those objects. The set of objects forms the extension of the concept while the set of attributes forms the intension of the concept. Knowledge is seen as applying in a context and can be formally defined as a crosstable as in Figure 5.1 below. The rows are objects and the columns are attributes. An X indicates that a particular object has the corresponding attribute. This crosstable is used to find formal concepts. The following description of FCA follows Wille (1982).

[2] Much of section 5.1 follows the paper "Knowledge Acquisition First, Modelling Later, Enric Plaza and Richard Benjamins (Eds), *Knowledge Acquisition, Modeling and Management*. 10th European Workshop, EKAW'97, Lecture Notes in Artificial Intelligence 1319, Springer-Verlag, 237-252."

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

Figure 5.1: Formal Context for "Vertebrates of the Animal Kingdom"

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 5.1 we have the context $\mathbb{K}$ of animals with G = {bird, reptile, amphibian, mammal and fish} and M = {has feathers, 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 feathers), (bird, warm-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:

$$X \subseteq G: \; X \mapsto X' := \{m \in M \mid gIm \text{ for all } g \in X\} \qquad (5.1)$$

$$Y \subseteq M: \; Y \mapsto Y' := \{g \in G \mid gIm \text{ for all } m \in Y\} \qquad (5.2)$$

are used to construct all formal concepts of a formal context, by finding the pairs (X′′,X′) for all X ⊆ G or (Y′,Y′′) for all Y ⊆ M. We can obtain all extents X′ by determining all row-intents {g}′ with g ∈ G and then finding all their intersections (formula 5.3). Alternatively Y′ can be obtained by determining all column-extents {m}′ with m ∈ M and then finding all their intersection (formula 5.4) . This is specified as:

$$X' = \bigcap_{g \in X} \{g\}' \qquad (5.3)$$

$$Y' = \bigcap_{m \in Y} \{m\}' \qquad (5.4)$$

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′). Figures 5.2 (a) and (b) demonstrate the steps performed in finding the formal concepts from the formal context in Figure 5.1. Where the intersection of two extents results in a duplicate it is not repeated. Figure 5.2 (a) shows the first pass of picking up the concept's extent. Figure 5.2 (b) is the second pass where the attributes which have the set of objects in the extent are added to the original intent. The end result is shown in Figure 5.3 which shows the eleven formal concepts that were derived for the formal context in Figure 5.1 by finding the intersection of sets of attributes for each object and then finding the sets of objects that have those intents. Note that the concepts have been ordered so that they do not appear in the same order as in Figure 5.2 (b). Note that the 11th concept in figure 5.2 (b) has added the concept that contains all attributes which has resulted in the set of objects being empty. This is done to ensure that we form a complete matrix with concepts 1 and 11 being the topmost and bottommost concepts, respectively.

To present a visualisation of our ordered set of concepts as a line diagram it is necessary to compute the predecessors and successors of each concept. Predecessors are found by finding the largest subconcept of the intents for each concept. Successors are found by finding the smallest superconcept of the intents. A superconcept is a set that has all of the members of another set and additional members. A subconcept is a set that has fewer members than another set but all the members it has are contained in the other set. We only concentrate on finding sub or super concepts of the intents or extents because they are inversely related and using either set will give the same result. It is interesting to note that the larger the intent the smaller the extent and visa versa. This means that the more attributes we use to describe an object reduces the number of objects that can be described in that manner. Conversely, the more objects in the extension the less attributes can be found in the intension which describe those objects. This approach is similar to the process of *realisation* in KL-ONE (Woods and Schmolze 1991) and the predecessor is like Woods (1991) most specific subsumer (MSS) and the successor is like the most general subsumee (MGS) which also require a concept first to be classified and then the MSS and MSG concepts to be determined.

Step	Intent	Extent
1		{1,2,3,4,5)
2	a	{1}
3	b	{4}
4	c	{1,4}
5	d	{2,3,5}
6	e	{3,5}
7	f	{1,2,4}

8	g	{2,5}
		{5}
		{2}

5(a)

Step	Intent	Extent
1	{ }	{1,2,3,4,5)
2	{a}	{1}
3	{b}	{4}
4	{c,f}	{1,4}
5	{d}	{2,3,5}
6	{d,e}	{3,5}

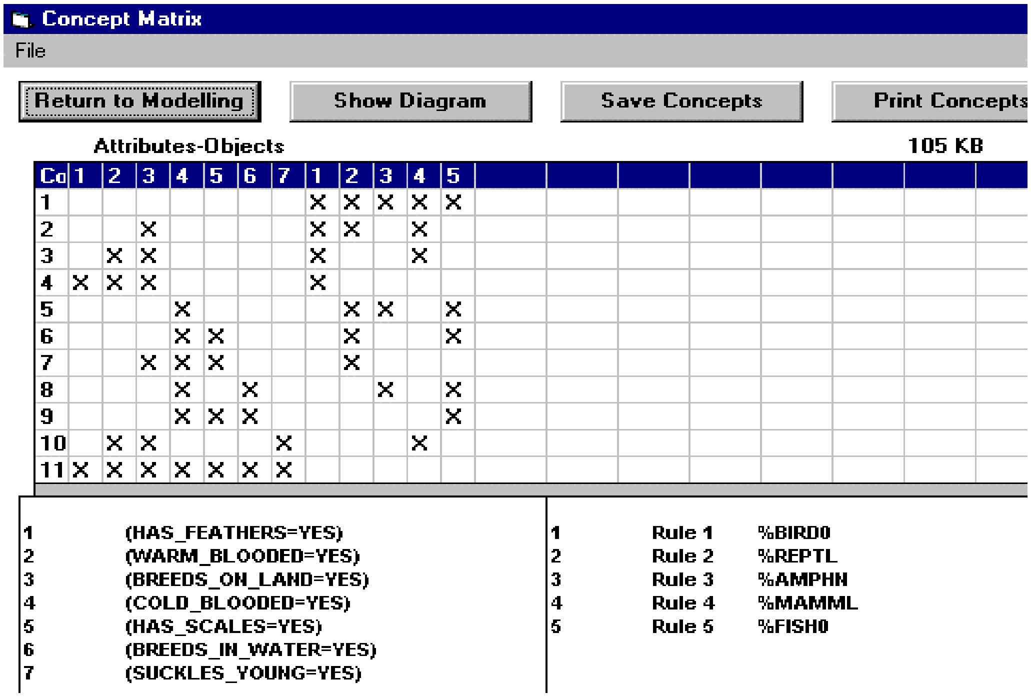
7	{f}
	{d,f,g}
		{1,2,4}
		{2}

8	{d,g}
	{d,e,g}
		{2,5}
		{5}

9	{a,b,c,d,e,f,g}	{ }

5(b)

Figure 5.2 The process of finding formal concepts from the formal context.



Attributes-Objects 105 KB

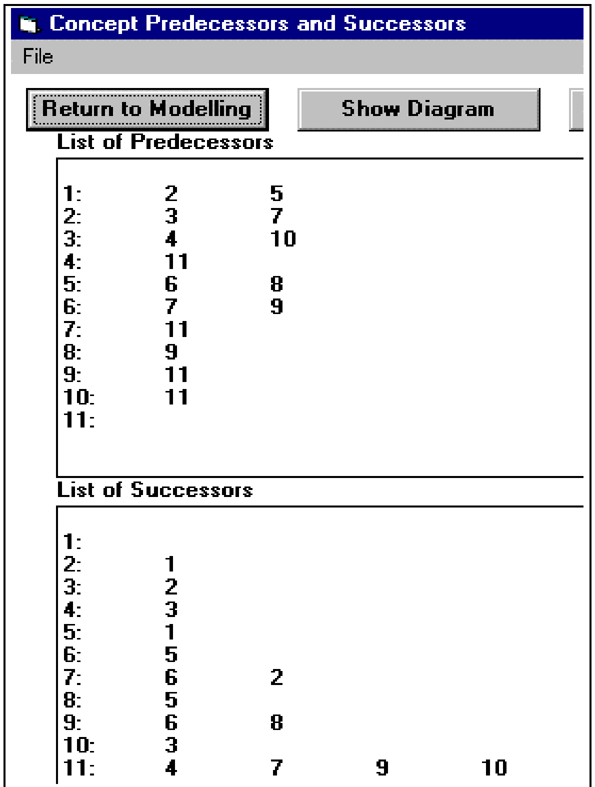
Co	1	2	3	4	5	6	7	1	2	3	4	5
1								X	X	X	X	X
2			X					X	X		X	
3		X	X					X			X	
4	X	X	X					X				
5				X					X	X		X
6				X	X				X			X
7			X	X	X				X			
8				X		X				X		X
9				X	X	X						X
10		X	X				X				X	
11	X	X	X	X	X	X	X					

1	(HAS_FEATHERS=YES)
2	(WARM_BLOODED=YES)
3	(BREEDS_ON_LAND=YES)
4	(COLD_BLOODED=YES)
5	(HAS_SCALES=YES)
6	(BREEDS_IN_WATER=YES)
7	(SUCKLES_YOUNG=YES)

1	Rule 1	%BIRD0
2	Rule 2	%REPTL
3	Rule 3	%AMPHN
4	Rule 4	%MAMML
5	Rule 5	%FISH0

Figure 5.3: The concept matrix screen from MCRDR/FCA.

Eleven (11) concepts have been found. Each row represents a concept. The columns show the attributes, which are listed first, followed by the objects. As was shown in the formal context in Figure 5.1, there are seven attributes and five objects, here labelled 1-7 and 1-5 respectively. The labels have been converted to numbers to allow the relationships between concepts and the possible patterns (particularly useful for comparison of models) to be more readily seen. Full labelling can be obtained by using the pop-up windows as shown in this figure or by clicking on the attribute, object or concept number. The concepts have been ordered to show the subsumption relations that exist. The top and bottom concept, concepts 1 and 11, show the concepts which includes all objects and all attributes, respectively.



List of Predecessors

```
1:     2     5
2:     3     7
3:     4     10
4:     11
5:     6     8
6:     7     9
7:     11
8:     9
9:     11
10:    11
11:
```

List of Successors

```
1:
2:     1
3:     2
4:     3
5:     1
6:     5
7:     6     2
8:     5
9:     6     8
10:    3
11:    4     7     9     10
```

Figure 5.4: The MCRDR/FCA screen which shows the list of predecessors and successors for concepts generated for the vertebrate domain shown in figure 5.1.

Figure 5.4 shows the list of successors and predecessors for the formal context in Figure 5.1. In MCRDR/FCA 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. This notion of parents and children is consistent with the view of an MCRDR KBS being comprised of parent and child nodes. The number of levels of parents and children are used to layout the line diagram and the algorithm from Richards and Compton (1997b) is given in Figure 5.5. However, just as users have different views of their knowledge, there is not one fixed way of drawing line diagrams and often a number of different layouts should be used (Wille 1992). The user may also reposition any of the nodes to their satisfaction providing a node is not moved higher than any of its parents or lower than any of its children. More formally, we use the subsumption relation ≤ on the set of all concepts formed such that $(X_1,Y_1) \le (X_2,Y_2)$ iff $X_1 \subseteq X_2$. For a family (Xi,Yi) of formal concepts of $\mathbb{K}$ the greatest subconcept, the join (5.5), and the smallest superconcept, the meet (5.6), are respectively given by:

$$\bigvee_{i\in I}(X_i,B_i) := ((\bigcup_{i\in I} A_i)'', \bigcap_{i\in I} B_i) \tag{5.5}$$

$$\bigwedge_{i\in I}(X_i,B_i) := (\bigcap_{i\in I} A_i, (\bigcup_{i\in I} B_i)'') \tag{5.6}$$

```
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
```

Figure 5.5: The algorithm used to find the concept coordinates for the Hasse diagram

From Lattice Theory, the ordered concept set can be used to form a complete lattice, called a concept lattice and denoted B(𝕂). The concept lattice is a semantic net which provides "hierarchical conceptual clustering of the objects (via the extents) …. and a representation of all implications between the attributes (via its intents)" (Wille 1992, 497). As noted in section 3.3.1.3 there are limitations to the extensional definition of a concept and only the intensional definition of a concept is used for exploring new concepts in MCRDR/FCA. However, for the development of the concept matrix, see Figure 5.3, and the concept lattice, see Figure 5.6, Wille's definition of a concept is used.

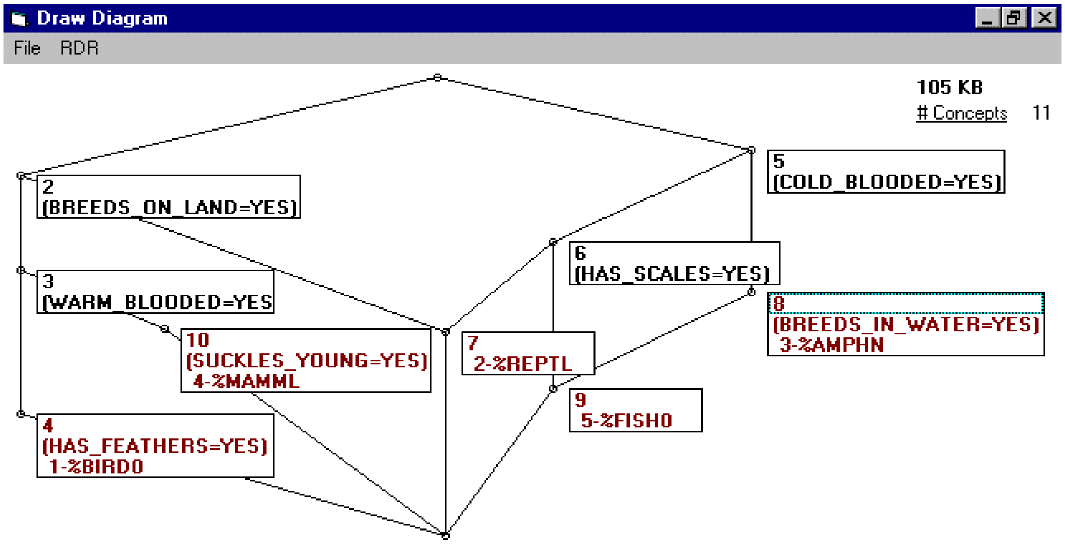


Figure 5.6: The diagram screen in MCRDR/FCA for the formal context "Vertebrates of the animal kingdom" given in Figure 5.3.

Each circle represents a concept. The attributes that belong to a concept are reached by ascending paths and objects are reached by descending paths. The rule conditions are the attributes. The objects are labelled using the rule number and the conclusion code. The conclusion codes have been made as meaningful as possible in the size limitation of five characters but a longer description could be substituted on the line diagram if desired for greater understanding.

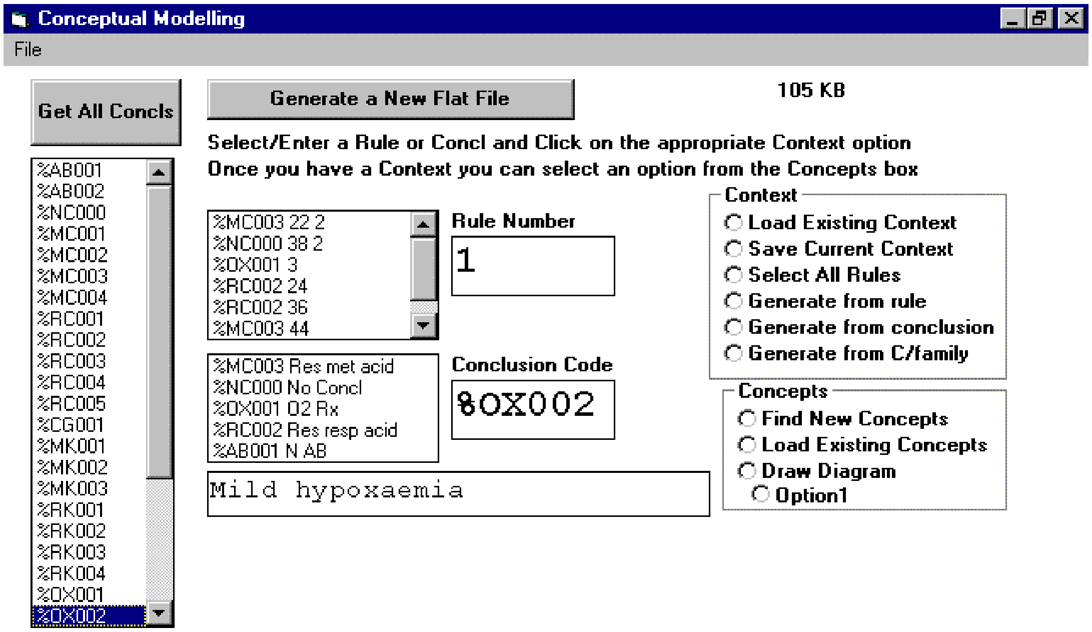
In Figures 5.6 and 5.10 the concepts are shown as small circles and the sub/superconcept relations as lines. Each concept has various intents and extents associated with it. The labelling has been reduced for clarity. All intents of a concept β are reached by ascending paths from β and all extents are reached by descending paths from the concept β. In MCRDR\FCA it is possible to display the concept, attribute/s or object/s belonging to each node or all three dimensions can be displayed concurrently, as in Figures 5.6 and 5.10. Although showing all dimensions increases the amount of information being presented the extra information is important in understanding the diagram presented. Alternatively, the user can click on an individual node to see the concept number and all of its extents and intents. The ability to express all relationships between attributes, such as which attributes occur together, and the ability to describe each object in terms of the concepts it contains and the relationship of those concepts to others is a major strength of the lattice structure (Cole and Eklund 1996).

Clancey points out that semantic networks can embody a cognitive model that exhibits patterns of human behaviour. However, since they are limited to words they constitute a "grammatical model of cognition" (Clancey 1991b, p.251) and do not capture non-verbal conceptualisations or model the perceptual-conceptual learning that occurs when humans attach meanings and interpretations to the words. These limitations are acknowledged together with the fact that to some extent concepts are being treated as "things" rather than "processes of perceiving and processes of behaving" (Clancey 1991b, p.252). Again it is stressed that while RDR systems can exhibit behaviour similar to a human expert there is no claim that the system developed matches the way that the human mind works. A major criticism of programs that use a grammatical model is that because they are bounded by the terminology used they are unable to learn at the knowledge level. In MCRDR/FCA a grammatical model is being used to uncover concepts that may not be so easy for an expert to articulate. This occurs because in many cases an expert will perform tasks at a subconscious level and may have difficulty explaining why they have acted thus. 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, p.261). Next we look at how the FCA theory described has been incorporated into MCRDR.

5.2 Combining MCRDR and FCA

As noted in Section 3.3.1, RDR and FCA share a number of views including the beliefs that knowledge applies in a context and that KA is a task that is best performed directly by experts. In both approaches KA is reduced to the task of classifying objects (cases) and the identification of the salient features. In FCA, KA begins with the elicitation of a crosstable from which the concepts derived can be used to generate implications. The implications generated are shown to the user who is asked to say whether they agree or disagree with the implication. If the user does not agree they are asked to offer a counterexample. This study starts from the opposite direction by using the rules in the MCRDR KBS as the input into a formal context. The reason for this is twofold. Firstly the purpose of using FCA was to uncover higher models in rules that had already been acquired using MCRDR. Secondly, it was felt that the RDR approach to KA was probably less demanding for experts than the development of crosstables, the analysis of the generated implications and the offering of counterexamples which is required by the FCA approach to KA .

The first step was conversion of the MCRDR KBS into a flat structure by sequentially traversing the KB for each rule picking up the conditions from the parent rule until the top node with the default rule was reached. From this flattened pathway of rules the user chose either the whole KB or a more narrow focus of attention from which to derive a formal context. When the whole KB was chosen the rules (which were identified by the rule number and conclusion code) and rule conditions formed the extents and intents, respectively. However, 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. Therefore, to limit the concepts to a manageable size that could be viewed in 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 (Wille 1989a) and is similar to the approach proposed by Ganter (1988) where the context is shortened to find subcontexts and subrelations. The technique described here is a little different in that it is not the formal context that is shortened but the rules which are put into a formal context are restricted according to the user's selection. These options are introduced below and described in more detail in Section 5.2.1.



Select/Enter a Rule or Concl and Click on the appropriate Context option
Once you have a Context you can select an option from the Concepts box

%AB001
%AB002
%NC000
%MC001
%MC002
%MC003
%MC004
%RC001
%RC002
%RC003
%RC004
%RC005
%CG001
%MK001
%MK002
%MK003
%RK001
%RK002
%RK003
%RK004
%OX001
%OX002

%MC003 22 2
%NC000 38 2
%OX001 3
%RC002 24
%RC002 36
%MC003 44

%MC003 Res met acid
%NC000 No Concl
%OX001 O2 Rx
%RC002 Res resp acid
%AB001 N AB

Rule Number: 1

Conclusion Code: %OX002

Mild hypoxaemia

Context
○ Load Existing Context
○ Save Current Context
○ Select All Rules
○ Generate from rule
○ Generate from conclusion
○ Generate from C/family

Concepts
○ Find New Concepts
○ Load Existing Concepts
○ Draw Diagram
○ Option1

Figure 5.7: The first selection screen in MCRDR/FCA which allows the user to select rules based on a specified case, rule, conclusion code or conclusion code family. This selection will be used to build a formal context from which concepts will be generated.

It seemed probable that a user may want to carry out modelling in connection with a particular case. As shown in Figure 5.7, the user could select a case and the rules. The conclusions associated with that case were then presented as the defaults from which to select. The user could also click the GET ALL CONCLS button to see a list of all conclusion codes from which to pick. If the "Generate from conclusion" radiobutton was clicked, all rules using the specified conclusion were selected and added as objects to the set G, forming the extension of the context. As each object was added the conditions of the rules were added to the set M of attributes to form the intension 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 conditions as the initial intension. Every condition 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 (conditions) found in the matching rule were also added to the intension. The result was a formal context $\mathbb{K}$ comprised of a set of objects G and attributes M connected by the binary relation I. An alternative selection screen, described further in Section 5.2.1 has also been developed that provides the user with views based on conditions, attributes, rules or conclusions with numerous options at various levels of abstraction to assist the user in selecting the aspects of the knowledge base of interest to them. An example based on selecting a conclusion follows.

	Normal Blood PH	Low Blood BIC	Low Blood PC02	1=l	High Blood PH	High Blood PC02	Low Blood PH	High Blood BIC	Incr Blood PH	Decr Blood BIC	Curr Blood PH≤7.36
9-%MC002	X	X	X	X							
10-%MC002		X		X	X						
14-%MC002	X			X		X					
15-%MC002				X			X	X			
19-%MC002	X			X		X			X	X	
49-%MC002	X			X		X					X

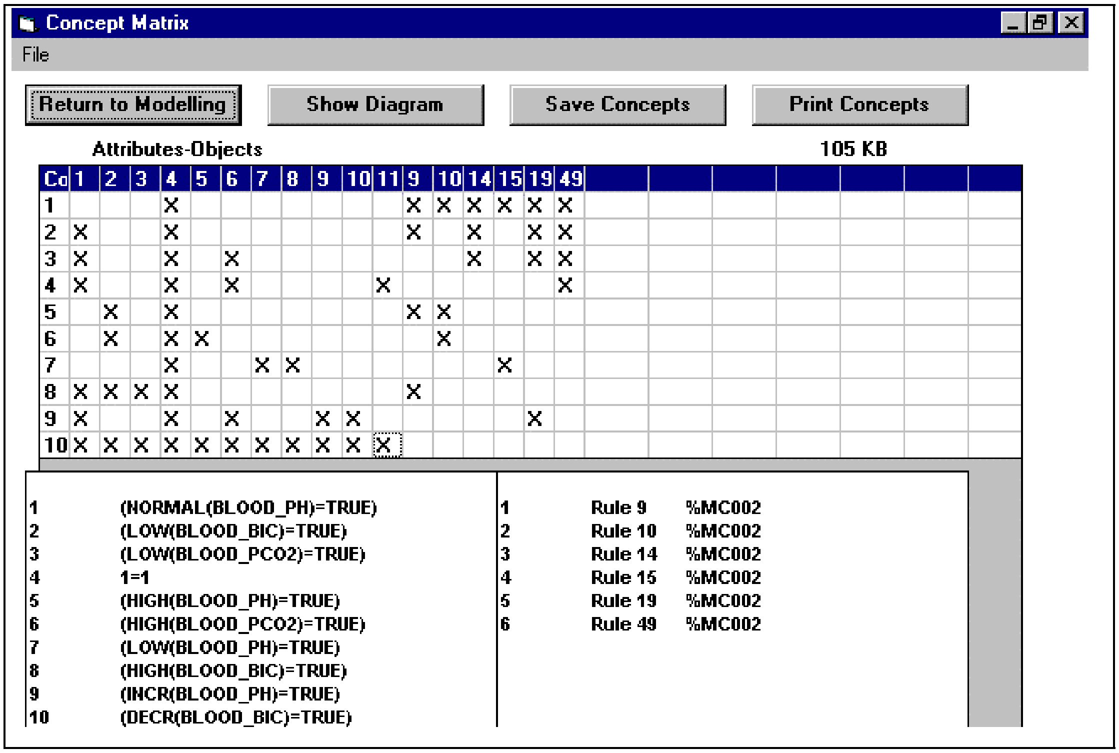
Figure 5.8: A formal context for the MCRDR rules which conclude %MC002- "Metabolic compensation.2" in the blood gases domain.

In figure 5.8 we have selected the conclusion %MC002 - *"Metabolic compensation.2"* from the blood gases domain. In Figure 5.8 the set of objects G = {9-%MC002, 10-%MC002, 14-%MC002, 15-%MC002, 19-%MC002, 49-%MC002}, where the object is referred to by the rule number and the conclusion. The set of attributes M = {Normal(Blood_Ph), Low(Blood_Bic), Low(Blood_PC02), 1=1[3], High(Blood_Ph), High(Blood_PC02), Low(Blood_Ph), High(Blood_Bic), Incr(Blood_Ph), Decr(Blood_Bic), Curr(Blood_Ph) ≤ 7.36}. The set of relations between G and M is I = {(9-%MC002, Normal(Blood_Ph), (9-%MC002, Low(Blood_Bic), (9-%MC002, Low(Blood_PC02),...,(49-%MC002, Curr(Blood_Ph) ≤ 7.36) }.

Our treatment of each rule condition, which is actually an attribute-value pair, as an attribute 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 MCRDR 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.

[3] 1=1 is the default condition for the default rule which is inherited by all rules.

Once the formal context has been generated from the users selection, concepts are found, ordered using the subsumption relation ≤, predecessors and successors are computed and a line diagram drawn according the process described in section 5.1. 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 5.9 where we can see five groupings of concepts in concepts 1-4, 5-6, 7, 8 and 9-10. In the ordering shown in Figure 5.8, concept 5 is a superconcept of concept 6. An alternate ordering could have shown concept 5 as a superconcept of concept 8. If the context is ordered before derivation of the concepts, a more aesthetic ordering of concepts may be produced but does not affect the calculation of predecessor and successors or the graph layout. The concept lattice for the conclusion %MC002 - “*Metabolic compensation.2*” is shown in Figure 5.10.



Co	1	2	3	4	5	6	7	8	9	10	11	9	10	14	15	19	49
1				X								X	X	X	X	X	X
2	X			X								X		X		X	X
3	X			X		X								X		X	X
4	X			X		X					X						X
5		X		X								X	X				
6		X		X	X								X				
7				X			X	X							X		
8	X	X	X	X								X					
9	X			X		X			X	X						X	
10	X	X	X	X	X	X	X	X	X	X	X						

1	(NORMAL(BLOOD_PH)=TRUE)
2	(LOW(BLOOD_BIC)=TRUE)
3	(LOW(BLOOD_PCO2)=TRUE)
4	1=1
5	(HIGH(BLOOD_PH)=TRUE)
6	(HIGH(BLOOD_PCO2)=TRUE)
7	(LOW(BLOOD_PH)=TRUE)
8	(HIGH(BLOOD_BIC)=TRUE)
9	(INCR(BLOOD_PH)=TRUE)
10	(DECR(BLOOD_BIC)=TRUE)

1	Rule 9	%MC002
2	Rule 10	%MC002
3	Rule 14	%MC002
4	Rule 15	%MC002
5	Rule 19	%MC002
6	Rule 49	%MC002

Figure 5.9: The concept matrix in MCRDR/FCA for Windows for the conclusion %MC002- “Metabolic compensation.2” in the Blood Gases domain. Ten (10) concepts have been found. Each row represents a concept. The columns show the eleven attributes, which are listed first, followed by the six objects as shown in the formal context in Figure 5.6. The attribute labels have been converted to sequential numbers and the object labels correspond to the rule number to allow the relationships between concepts and the possible patterns to be more readily seen. Full labelling can be obtained by using the pop-up windows as shown in this figure or by clicking on the attribute, object or concept number. The concepts have been ordered to show the subsumption relations that exist. The extent of the top concept, No 1, includes all objects. The intent of the bottom concept, No 10, includes all attributes.

The example given in this section showed the results when the %MC002- "Metabolic compensation.2" conclusion was chosen. As stated, this is only one of a number of ways of selecting what rules should be included in a formal context. We look at the range of possible views in the next section.

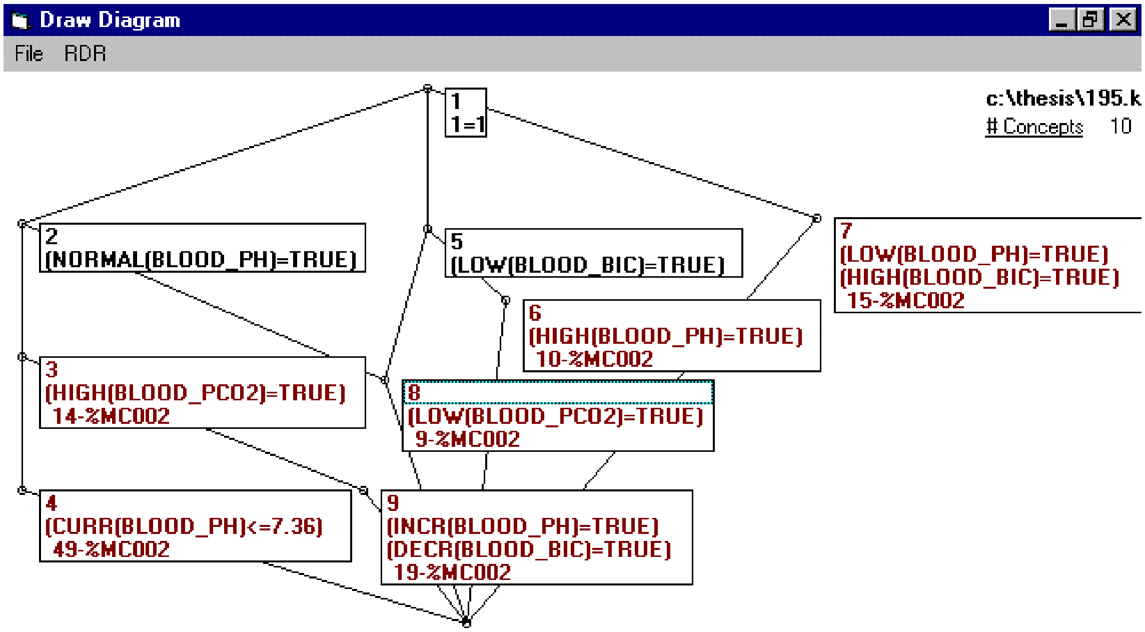
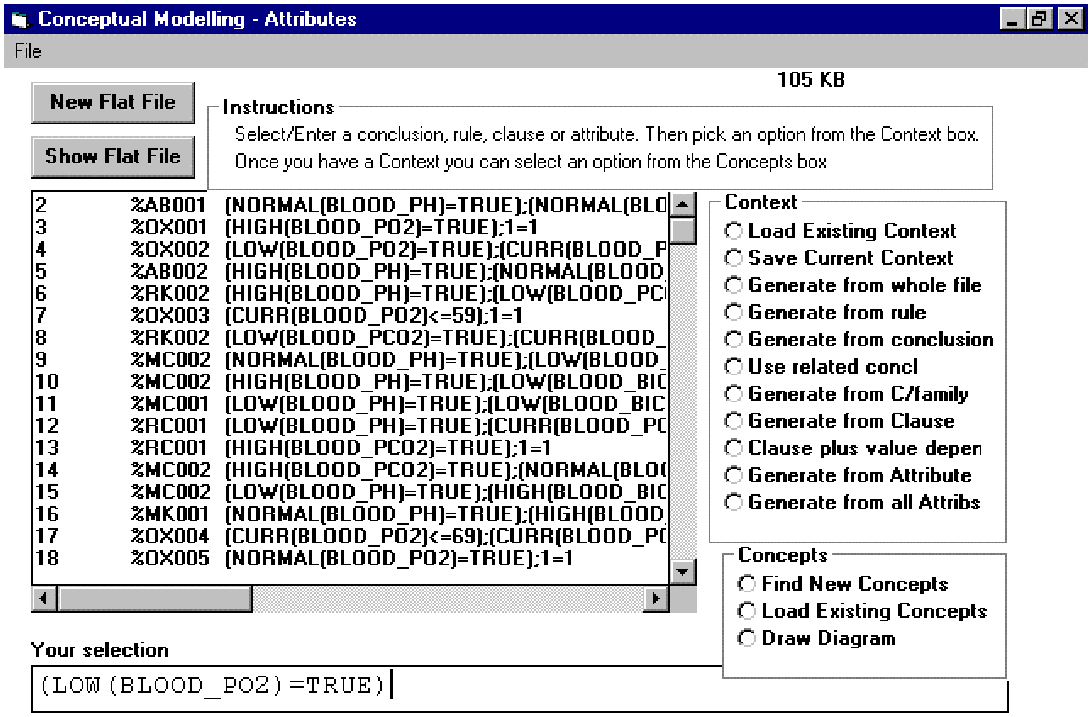


Figure 5.10: The Concept Lattice in MCRDR/FCA for the conclusion %MC002- "Metabolic compensation.2".

5.2.1 The Possible Views using MCRDR/FCA

The selection screen described above and shown in Figure 5.7 was the first selection screen developed in MCRDR/FCA and only offered limited options from which the user could define a focus of attention. Restricting the aspects of the KBS to be viewed is important for keeping the information presented to the user to a manageable size. Below a total of nine different options, as shown in Figure 5.11, are described. Within some of these options there were further choices that the user could make. The purpose of each view is given. In keeping with the situated view, the idea was to present the user with the range of views that may be useful but the choice of which one to use was to be left to the user who would best know their current situation. The major problem with such an approach is that the more choices a person is given the harder it is to understand and remember the differences between choices and when they are applicable. As can be seen below in Figure 5.11 there are 11 radio buttons in the Context frame. The first two options involve file handling with the "Load Existing Context" button for loading a previously created context file and the "Save Current Context" button for writing the last context formed into the context file. The nine options that follow concern the different selection criteria that can be used in deciding which rules should be included in a formal context. We look at these options now.



Select/Enter a conclusion, rule, clause or attribute. Then pick an option from the Context box.
Once you have a Context you can select an option from the Concepts box

```
2          %AB001  (NORMAL(BLOOD_PH)=TRUE);(NORMAL(BLO
3          %OX001  (HIGH(BLOOD_PO2)=TRUE);1=1
4          %OX002  (LOW(BLOOD_PO2)=TRUE);(CURR(BLOOD_P
5          %AB002  (HIGH(BLOOD_PH)=TRUE);(NORMAL(BLOOD
6          %RK002  (HIGH(BLOOD_PH)=TRUE);(LOW(BLOOD_PC
7          %OX003  (CURR(BLOOD_PO2)<=59);1=1
8          %RK002  (LOW(BLOOD_PCO2)=TRUE);(CURR(BLOOD_
9          %MC002  (NORMAL(BLOOD_PH)=TRUE);(LOW(BLOOD_
10         %MC002  (HIGH(BLOOD_PH)=TRUE);(LOW(BLOOD_BIC
11         %MC001  (LOW(BLOOD_PH)=TRUE);(LOW(BLOOD_BIC
12         %RC001  (LOW(BLOOD_PH)=TRUE);(CURR(BLOOD_PC
13         %RC001  (HIGH(BLOOD_PCO2)=TRUE);1=1
14         %MC002  (HIGH(BLOOD_PCO2)=TRUE);(NORMAL(BLO
15         %MC002  (LOW(BLOOD_PH)=TRUE);(HIGH(BLOOD_BIC
16         %MK001  (NORMAL(BLOOD_PH)=TRUE);(HIGH(BLOOD
17         %OX004  (CURR(BLOOD_PO2)<=69);(CURR(BLOOD_PC
18         %OX005  (NORMAL(BLOOD_PO2)=TRUE);1=1
```

(LOW(BLOOD_PO2)=TRUE)

Figure 5.11: The Second Selection Screen in MCRDR/FCA.

5.2.1.1. Generate from whole file

This option uses all the rules in the flat file to generate a context. This is the simplest technique. Each rule corresponds to a row in the crosstable. The object name is a combination of the rule number and the rule's conclusion. The rule conditions form the attributes/columns.

5.2.1.2. Generate from rule

This option uses the specified rule to form the first object in the crosstable, with the conclusion being the object and the rule conditions forming the attributes. Then each record in the flat file, which is a pathway in the KB, is searched for a match on any of the specified rule's conditions. If a match is found the whole record is added. This means that any new attributes introduced by the matching rule are added as a new column in the context. An 'X' is marked for all attributes found in this rule. Only exact matches are picked up. Neighbouring values are not picked up. Neigbouring or borderline values are discussed further under option 5.2.1.7.

5.2.1.3. Generate from conclusion

Each record in the flat file is read and any rules with the specified conclusions are added to the formal context. Each condition is checked against existing conditions (attributes) in the formal context and added if it does not already exist. An 'X' is marked in the corresponding column.

5.2.1.4. Use related conclusion.

This option allows the user to generate concepts for all conclusions related to and including the specified conclusion. The notion of *related to a conclusion* is simply that if a rule shares any conditions with rules that give the specified conclusion there is some sort of relationship between them. This option is a combination of the rule and conclusion options. Thus the range of possible indicators that the two conclusions are related has been broadened. Each rule with the specified conclusion is used to generate a list of conditions. Then all rules which use any of these conditions are added to the formal context. So instead of using one rule as the basis for comparison the set of rules with the specified conclusion are used in the same manner as reported in 5.2.1.2. This will typically result in a much larger number of selected rules as each condition in every rule in the set is used as the selection criteria of other rules. This takes a different approach to using conclusion families, option 5.2.1.5, because it does not rely on the expert to identify a relationship but uses the content of the rule as the basis for a possible relationship.

This approach will include some rules whose conclusions are not really related to the selected conclusion. This overgeneralisation is a drawback but is not seen as a major problem. The aim of these selection options is to narrow the focus and remove at least some of the irrelevant concepts. It should be apparent from the diagram if a rule/conclusion is not really related. An extension to the current diagram screen would be to allow the user to drop concepts that are not of interest. Further suggested improvements are described in Section 5.2.2.

5.2.1.5. Generate from conclusion family

A conclusion family refers to a naming convention used to identify that certain conclusions are related to others. The PEIRS conventions for defining a conclusion has been used. The conclusion codes are in the form %XXNNN where X is an alphabetical character and N is numeric, such as %CL001 may stand for wear jumper and %CL002 may mean wear a t-shirt. Both conclusions are related to the which clothes (CL) should be worn. Other rules that concern which clothes to wear should use the same %CL prefix with the next sequential number allocated at the end. Generating a context from a conclusion family is very similar to option 5.2.1.3 that uses the whole conclusion, except only the first three characters are compared. If another conclusion naming convention was used then the current code would need to be modified, but as the system is a prototype a better way that lets the user specify the convention is not currently planned.

5.2.1.6. Generate from clause

In this option the user is able to specify/select a condition which is used for comparison with the attributes used in the rule conditons. If the condition is found then the rule is added as a new object with the rule conditions as the attributes. All conditions are checked if they are already a part of the formal context before being added, so there are no duplicates. An ‘X’ is marked in the corresponding column for that attribute/condition. The user does not need to enter the full condition but can use any string for matching. For example the attribute name (BLOOD_PC02) or even partial name (PC02) can be entered.

The user may combine conditions using AND (&) so that all joined conditions must be found to be a match. OR (|) has not been implemented because it can be achieved by rerunning the option and taking ADD to add the new context to the existing context which will pick up OR conditions.

When the user takes this option they are asked the question "Do you wish to pick up attributes associated with your selection ?". If the reply is NO only the conditions that match those in the selection criteria are added as attributes to the formal context. By replying NO and using the “&” operator only the attribute information for specified conditions is added to the formal context. This avoids adding other conditions that are not of primary interest and which may obscure the picture. Such an approach would be used when the expert knows which attributes/conditions they are interested in.

5.2.1.7. Clause plus value dependency

This is similar to the option above but instead of only picking up exact matches, conditions that use the same attribute as the one in the specified condition are evaluated against an attribute file to see if the specifed value has neighbouring values. For example, the condition specified may be TEMP=HOT. If the possible values for TEMP are HOT, WARM and COLD then WARM can be seen to be a bordering value. All rules that used TEMP=HOT or TEMP=WARM are added to the formal context. This is a kind of fuzzy treatment of values and is aimed at overcoming differences in perception between individuals and over time.

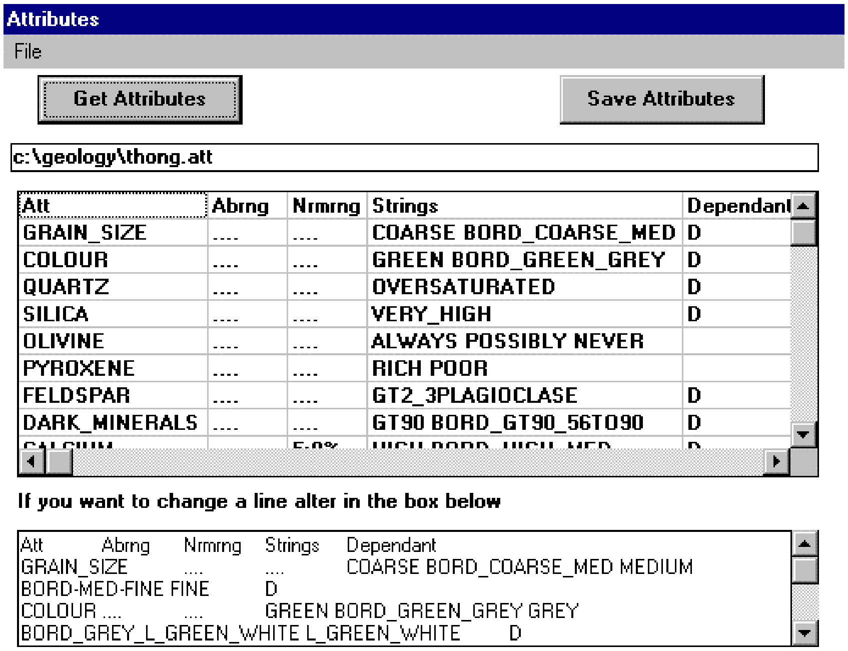
Two possible approaches to handling borderline values when developing formal contexts have been considered. The first is to use a border category, such as BORDER-HIGH-MEDIUM, so that if the specified value is HIGH and a rule is found with this value, an ‘X’ is marked in the HIGH and BORDER-HIGH-MEDIUM attribute column for that rule. If another rule uses the value MEDIUM it would be picked up as a borderline value and an ‘X’ marked in the MEDIUM and BORDER-HIGH-MEDIUM attribute column for the second rule. This will result in both rules intersecting on the BORDER-HIGH-MEDIUM attribute showing a relationship between the two rules that may otherwise have not been revealed in the line diagram. The other approach is to find the value/s next to the value in question, which could be higher and/or lower values, and mark an ‘X’ in the value specified and those values adjacent to it. This would result in the two rules above having an ‘X’ marked in the both the attribute-value HIGH and MEDIUM columns. But this approach does not show the actual value assigned to the rule condition and does not show which value is the borderline value.

```
Attribute       Strings                                 Dependency Code
GRAIN_SIZE.     COARSE BORD_COARSE_MED MEDIUM BORD-MED-FINE FINE   D
COLOUR          GREEN  BORD_GREEN_GREY  GREY  BORD_GREY_L_GREEN_WHITE  L_GREEN_WHITE
                   D
QUARTZ          OVERSATURATED   BORD_OVERSAT_SAT   SATURATED   BORD_SAT_UNDERSAT
                UNDERSATURATED       D
SILICA          VERY_HIGH  BORD_VERY_HIGH_INT  INTERMEDIATE  BORD_INT_LOWISH  LOWISH
                BORD_LOWISH_V_LITTLE V_LITTLE    D
OLIVINE         ALWAYS POSSIBLY NEVER
PYROXENE        RICH POOR
FELDSPAR        GT2_3PLAGIOCLASE  BORD_GT2_3PLAG_1_3TO2_3PLAG  1_3TO2_3PLAGIOCLASE
                GT2_3ORTHOCLASE ABSENT     D
DARK_MINERALS   GT90   BORD_GT90_56TO90   56TO90   BORD   56TO90_30TO56   30TO56
                BORD_30TO56_LT30 LT30      D
CALCIUM         HIGH BORD_HIGH_MED MEDIUM BORD_MED_LOW LOW   D
POTASSIUM       HIGH BORD_HIGH_MED MEDIUM BORD_MED_LOW LOW  NO     D
IRON            RICH BORD_RICH_INT INTERMEDIATE BORD_INT_LOW LOW   D
```

Figure 5.12: The Attribute file for the SISYPHUS III Geology Domain. Bordering value have been included in attributes identified as containing values that may overlap.

The first option using boundary categories has been implemented. In Figure 5.12 the attribute file for the SISYPHUS III geology domain is shown. This file has been updated to include bordering value categories and the last field holds the code “D” which indicates the attribute has dependant values. Currently, to maintain the attribute file the user goes into the attribute screen shown in Figure 5.13 and specifies which values are related to others. It is envisaged that this task would be done at system setup time when the user is also specifying the relevant functions and appropriate preprocessing of the data.

This option also lets the user just focus on the selected attributes but does not cater for the & or | operators as in option 5.2.1.6. To handle an OR situation the user can repeatedly use this option for as many conditions as they are interested in, taking the ADD option when asked if they want to create a new context or add to an existing context. The ability to select using conjunctions of attribute using bordering values is left to future work. Further discussion on bordering values and their treatment in the nearest neighbour algorithm was given in Section 3.3.2.1.



c:\geology\thong.att

Att	Abrng	Nrmrng	Strings	Dependant
GRAIN_SIZE	....	....	COARSE BORD_COARSE_MED	D
COLOUR	....	....	GREEN BORD_GREEN_GREY	D
QUARTZ	....	....	OVERSATURATED	D
SILICA	....	....	VERY_HIGH	D
OLIVINE	....	....	ALWAYS POSSIBLY NEVER	
PYROXENE	....	....	RICH POOR	
FELDSPAR	....	....	GT2_3PLAGIOCLASE	D
DARK_MINERALS	....	....	GT90 BORD_GT90_56TO90	D
CALCIUM		5-0%	HIGH BORD_HIGH_MED	D

If you want to change a line alter in the box below

```
Att         Abrng      Nrmrng     Strings      Dependant
GRAIN_SIZE             ....       ....         COARSE BORD_COARSE_MED MEDIUM
BORD-MED-FINE FINE                D
COLOUR ....            ....       GREEN BORD_GREEN_GREY GREY
BORD_GREY_L_GREEN_WHITE L_GREEN_WHITE        D
```

Figure 5.13: The Attributes screen in MCRDR/FCA where the user can define bordering value categories for any attributes.

5.2.1.8. Generate from attribute

This option is appropriate when the user wants to generate concepts for a specified attribute without regard for the value and also display the attribute without the value. Option 5.2.1.6 allows the user to select by attribute alone but when the concept is displayed the whole rule condition is shown in the diagram. Disregarding the value in the condition both in finding other similar objects as well as in the labelling results in abstraction of the concepts to a higher level.

5.2.1.9. Generate from all attributes

This option is similar to the previous option 5.2.1.8 but the user does not have to specify which attribute to abstract as all attributes are abstracted. This option saves repeating every attribute and substantially reduces the number of columns in the crosstable to cover a domain. This option is seen as a good first insight into a KBS at a higher level to determine what other areas may be of interest. Figure 5.14 shows the line diagram produced when this option was taken for the blood gases KBS. The diagram shows which rules use which attributes and gives some idea of relationships between attributes and which ones are likely to occur together.

5.2.1.10 Discussion of the possible views in MCRDR/FCA

The options offered above for selecting which rules to include in a formal context may be combined. It is envisaged that different users and applications will have particular requirements which may only require a subset of the options offered here and may require the addition of options not considered in this list. The purpose of these options was to:

- minimise the number of concepts generated to improve the comprehensibility of the line diagram,
- let the user focus on what is of greatest interest without extraneous concepts to cloud the result,
- provide sufficient choices to give the user control and flexibility to suit a wide range of situations.

While, the views of the knowledge provided through this selection screen meets these criteria it is not claimed that these options are optimal or comprehensive. A key issue with the options for selecting views described above is how to present them to the user so that they know when and how they can use them. As more options are made available the decision when to use them and how to combine them becomes difficult. It is suggested that the particular options made available, which may include options not explored in this thesis, should be tailored to the needs of the users for each particular application. The options that allow selection by rule, conclusion or conclusion family should be standard inclusions but others will depend on the nature of the domain and the needs of the user. We go on to look at the shortcomings of MCRDR/FCA and possible enhancements.

5.2.2 Limitations of MCRDR/FCA and Proposed Improvements

The system described in this chapter is a prototype that has been used as a proof-of-concept that knowledge in an MCRDR KBS could be reused as the basis of developing an abstraction hierarchy to support activities that require greater understanding of the concepts and their structure than was provided in the RDR assertional KBS. The system thus does not offer the full range of features that may be wanted in a deployed system. There is no apology for this as it is strongly believed that any system that solves a significant problem will need tailoring to the environment in which it will be used. This is seen as one of the benefits of RDR since the KBS is user-developed and tailoring of the system by the KE to different domains and applications is relatively simple using the Visual Basic or Hypercard software. There are, however, a number of enhancements that would improve the usability and usefulness of MCRDR/FCA regardless of the application or domain. Of course, the particular application or domain will affect how important it is to include these enhancements. We look now at some features that would improve the MCRDR/FCA system.

There are many changes which can be made to the user interface. Many of these changes relate to the limitations of the amount of information that can be displayed comprehensibly at one time. Important enhancements include the ability to drop nodes from the lattice or to zoom in and out using selections made via the concept lattice to determine what new contexts and concept lattices should be developed.

The limitations of visual representations and the need to capture and show related values were major driving factors in the development of TOSCANA (Vogt and Wille 1995). In TOSCANA a scale for each attribute is developed which is a simple line diagram for that attribute showing the relationships between the possible values. When more than one attribute is considered at a time then the line diagrams become nested according to the attribute sequence selected so that at a node for one attribute-value combination a smaller line diagram for the next level of attribute is shown within that node. TOSCANA is a commercial product and a number of large organisations are making use of, or are at least interested in, this tool. However, the number of levels of nesting possible are obviously once again limited by the amount of space on the screen and there is also the problem that the order of attributes will affect the interpretation of the line diagram.

The major problems with extension of the MCRDR/FCA tool to support zooming and the handling of multiple contexts concurrently are the limitations of Visual Basic and the amount of processing required. Visual Basic has been satisfactory for the development of a prototype but many restrictions on the size of such things as strings and grids (tables) has meant that KBS over a few hundred rules can not be handled without most of the code being rewritten in another language such as C. Essentially all processing needs to be performed externally and Visual Basic used simply for the interface, just as the RDR Engine part of MCRDR/FCA was written in C by Phil Preston.

An additional problem which goes beyond the limitations of Visual Basic is the fan out problem that occurs as the size of the knowledge base grows. As described in Section 5.1, concepts are formed by taking the intersection of sets of attributes and the set of objects which share those attributes. Each time a new row in the formal context is processed the new concept is intersected with every existing concept and results in the exponential growth of the set of concepts. In the worst case this is an NP-Hard problem that can not be remedied with the addition of more computer resources as computer resources can only assist in linear time problems. As is common with NP-hards problems in AI some sort of pruning of the solution space is offered in this thesis by restricting the size of the contexts through the various selections offered in Section 5.2.1.

In Section 7.3.1 it is shown how a proposed rule can be critiqued against existing pathways in the KBS. A nice and not too hard extension to this would be the ability to view this proposed rule as part of a concept lattice. This would also be a form of 'what-if' analysis that allows the user to explore what the outcome of the rule will be on the existing model before they make the commitment to add the rule.

Section 5.3.2 and the next chapter describe the use of MCRDR/FCA for the comparison of different KBS. Comparison of the concept matrices can be assisted by ordering the rule conditions (attributes in the formal context). Then some form of visual pattern recognition, manual or automatic, could be performed to produce subsets of the concept matrices showing matches and differences.

Extensions have been made to MCRDR/FCA which allow the user to develop a number (currently up to four) concept lattices and then to be shown a concept lattice of the similarities and differences between the four models. The user may also load the four KBS so that they can view the case associated with a particular rule to be used as a counterexample in possible discussions between owners of the different KBS. This feature was preliminary work to the RE work described in Chapter Six which combines the different KBS at the beginning into one flat file and then allows different views of the combined KBS to be taken.

In Section 5.2.1.7 a description was given on how boundary-related values could be handled. This issue is important because relevant rules may not be included in the formal context developed if the right selection criteria is not given. It is also important because without consideration of boundary-related values the actual relationships between rule conditions and conclusions will not be evident in the diagram. The line diagram should show that two rule conditions that refer to the same attribute one with the value HIGH and the other with the value MEDIUM are closer than if the value for that attribute in the second rule condition was LOW. The solution described in Section 5.2.1.7 has been implemented but not explored or evaluated in great depth and could possibly be improved by the technique known as conceptual scaling where different operators are used for comparison. In MCRDR/FCA the = operator is used when determining the intersection of sets but it should be possible to implement a range of operators such as >= or <= which can be combined. This would allow more precision in the specification of what type of boundary-values are to be included. For example, the user may be interested in all rules where BLOOD-PH = HIGH and BLOOD_BIC = NORMAL. To ensure cases with borderline values for these attributes are selected they could specify the borderline value for BLOOD_PH which would pick up in addition any conditions where BLOOD_PH=NORMAL and only choose to pick up borderline values greater than the specified value for BLOOD_BIC so that conditions with BLOOD_BIC=HIGH would be included but not BLOOD_BIC=LOW. When we consider the nature of the Blood Gases domain where each rule condition is made up of a function, attribute, operator and value and the large number of possible variations for each, particularly the value, it may be essential to include conditions that satisfy a number of criteria.

Despite the limitations and need for further development beyond a prototype, MCRDR/FCA has been applied in a number of domains where it has been used to gain some understanding of the KBS that was not as easy through other methods such as rule traces or browsing the KBS. Four Domains are described. The first three involve classification problems and the fourth is a configuration problem. As is noted in the description of the case studies which follow the classification domains provided a much richer source of information than the configuration domain where there was virtually no abstractions to be found. Since it is early days for MCRDR for configuration it is not clear whether this result is due to the nature of the domain or the way in which it has been developed. Some conjectures have been made in Section 8.2.2.4 regarding the results in this domain.

5.3 Case Studies using MCRDR/FCA

The MCRDR/FCA system has been used in a number of domains including pathology, agriculture, geology and chemistry. Each of these are discussed briefly below with a demonstration of a different aspect of how the tool can be used.

5.3.1 Blood Gases Domain

The first domain that was looked at with MCRDR/FCA 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 line diagram in Figure 5.14 has been generated from a context based on attributes only without regard for the values, option 11 on the second selection screen in Figure 5.11 and described in Section 5.2.1.9. This diagram is interesting as an abstraction of the knowledge base. It can be seen that conclusions from the same conclusion family are generally grouped together. For example, the attribute BLOOD_BIC is used by all conclusions for the %MC conclusion family and BLOOD_P02 is only used by the %OX conclusion family. These groupings of conclusions into families are to be expected since the expert has already identified that there is a relationship between them. There were eight different conclusion families: AB, CG, MC, MK, NC, OX, RC and RK. As pointed out by the expert in the evaluation study performed in Section 8.2.2.1 but not known at the time by myself, the use of the letter “K” as the second letter in the code meant that BLOOD_PH=HIGH. In Figure 5:14 it can be seen that all the conclusions using “K” use the attribute BLOOD-PH but this abstracted view does not give us the information regarding the value. To find this information it is necessary to zoom further into the diagram taking into consideration all concepts that use the attributes BLOOD_PH, BLOOD_BIC and BLOOD_PC02. The view shown in Figure 5:14 was one of the first drawn and was used as a proof of concept that the models developed using FCA were consistent with the model the expert was expressing in the KBS assertions.

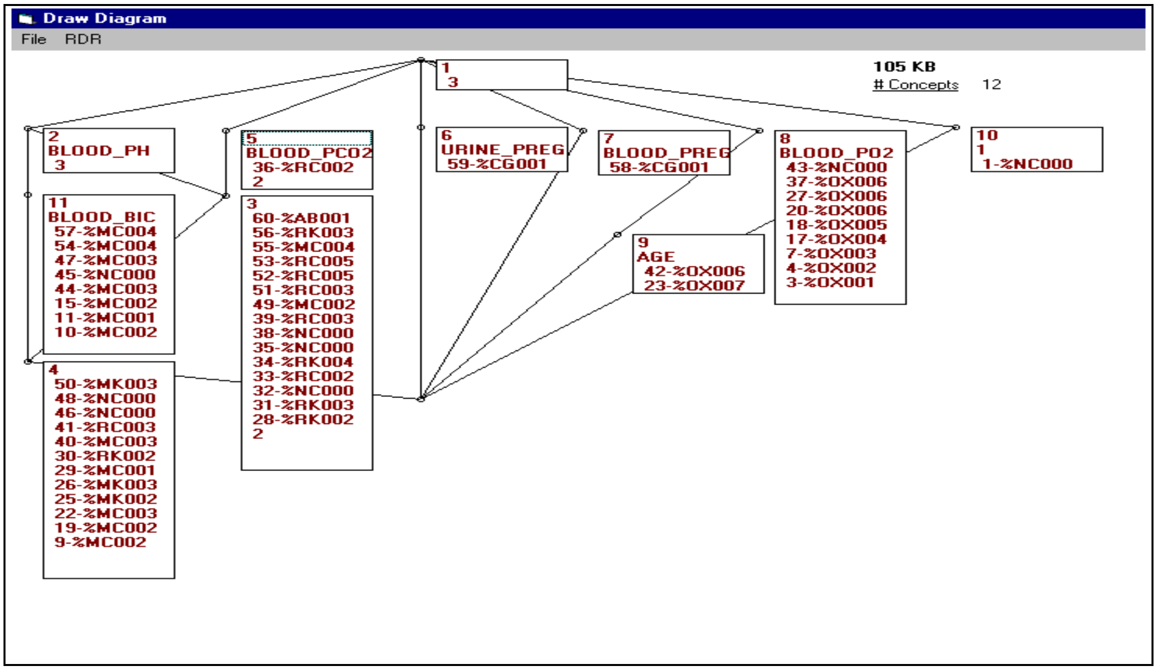


Figure 5.14: The line diagram in MCRDR/FCA for Windows using a formal context based on the attributes in the Blood Gases KBS without regard for the values of those attributes.

The blood gases KBS was used as a test domain to see how useful MCRDR/FCA could be for learning the key concepts in the domain. The results of these evaluations which involved the expert that developed the 105 KBS are described in Section 8.2.2.1.

5.3.2 Lotus Domain

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 et al 1996). The knowledge was recorded into four KBS by four independent agricultural advisors. These four advisors were not experts regarding the Lotus crop as knowledge in this domain was emerging and the main purpose of developing multiple KBS was to build up a corpus of knowledge for this new domain. The concept matrices and line diagrams in Figures 5.15 and 5.16 were used to compare the conceptual models of the advisors. This technique was seen as a useful way to identify the main concepts and reconcile any differences within this domain.

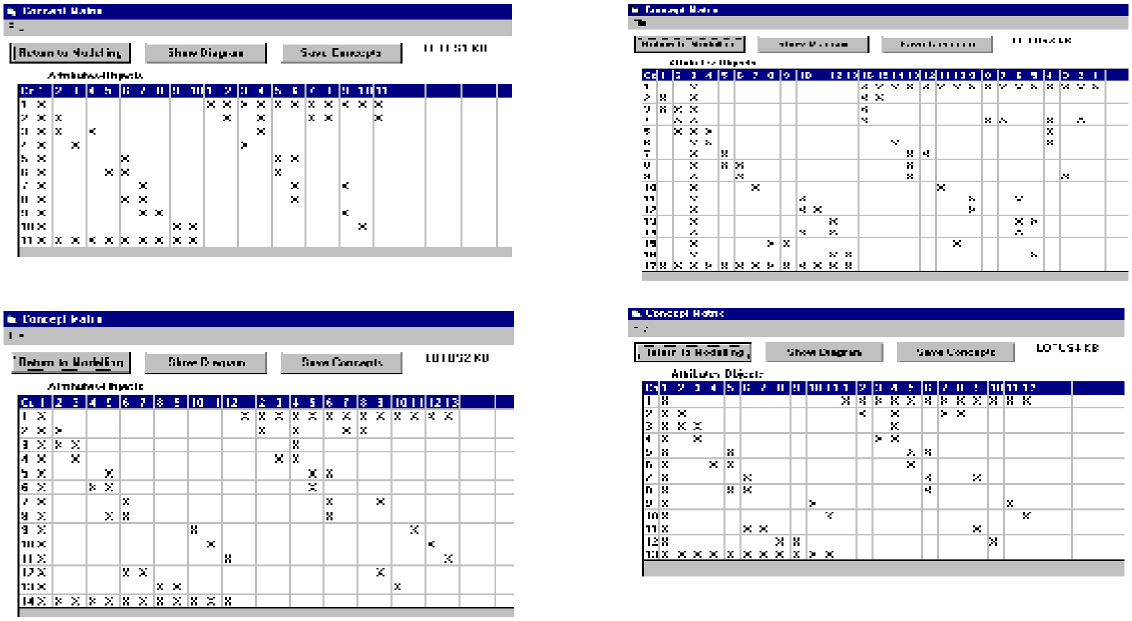


Figure 5.15: The concept matrices for the four LOTUS KBS in MCRDR/FCA

The concept matrices for the four LOTUS KBS are shown in Figure 5.15. It is not expected that you can properly read these four matrices since they have all been reduced to fit into the one figure but the purpose of Figure 5.15 is to show the visual similarity between the four KBS. Through the use of labelling as shown in Figures 5.3 and 5.9 it is possible to make a detailed comparison between the four models using the concept matrix. From such comparisons it can be seen that all KBS share a number of concepts, (the first nine attributes and ten objects are the same in each KBS). It can also be seen that the fifth concept in the Lotus3 KB is not shared by any of the others and that the advisor considers that when the LOTUS_RATE >=3 the conclusion of *Ryegrass* no longer holds and it should be replaced by *No Conclusion*. Concepts 12 and 13 in the Lotus3 KB represent new concepts that are not shared by any of the other KBs. The concept intent (SCARABS=YES) and the concept extent (%SCARA), which represent a rule condition and rule conclusion respectively, have been introduced by this advisor. The conclusion %SCARA is an abbreviation for “Maku OK, but persistence limited by scarabs”. By looking at the matrix the experts are able to see not only what attributes (intents) and conclusions (extents) others consider important but also the relationship between them and how it affects other conclusions.

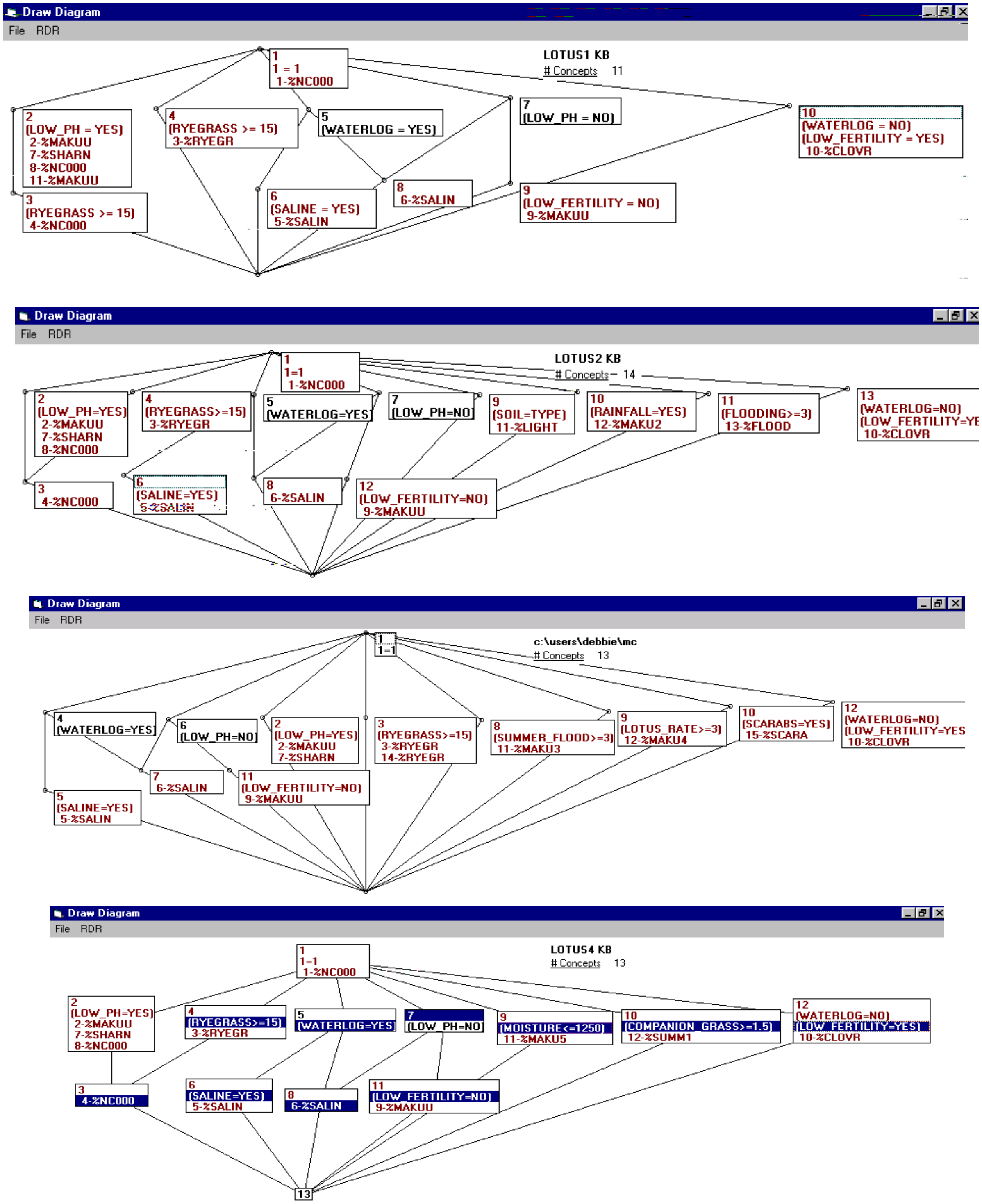


Figure 5.16: The line diagrams for Lotus 1, 2, 3 and 4 KBS in MCRDR/FCA

The line diagrams for Lotus 1, 2, 3 and 4 are shown in figure 5.16. The line diagram provides a more hierarchical understanding of the sub and super relationships in the domain than the concept matrix. At a glance we can see that Lotus1 and 2 have 11 and 14 concepts, respectively, and that the three concepts that are different are concepts number 9, 10 and 11 in the Lotus2 KBS. These concepts have introduced new attributes and conclusions (objects) not used by the Lotus1 KBS. The structure of the knowledge in both KBS is very similar with four levels of concepts in both. Even though concepts 2, 3 and 4 in both KBS appear to be slightly different structurally they embody the same ideas. Due to inheritance of attributes on higher paths both advisors consider that when (LOW_PH=YES) and (RYEGRASS>=15) the conclusion should be %NC000 No Conclusion. Without enumerating all similarities and differences between all KBS, we can see that the concept matrix, and even more so the line diagram, provide succinct but powerful tools for analysing conceptual models. To facilitate comparison it was important to ensure the attributes shared by all contexts were in the same order before the concepts were determined. The examples shown used all the rules in the LOTUS KBS since the number of rules ranged from 11-18. As described earlier the focus of attention could have been narrowed by using the various options on the selection screen to reduce the number of objects to be included in the formal context. The recommended approach is to view the whole KB to identify variations and then reduce the context by selecting the rules or conclusions that differed.

An extension to the approach is given in Chapter Six where the reconciliation of expertise in multiple KBS from multiple sources is treated as a requirements engineering task and was spawned from the introductory work described in this section. In the next section we look at the third and final classification domain considered.

5.3.3 SISYPHUS III Geology Domain

The third domain was the SISYPHUS III (Shadbolt 1996) geology domain which also involved knowledge from multiple experts. Initially the tool has been used to understand the key concepts of the domain. We can see in Figure 5.17 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.

In addition to using the concept lattice and matrix for analysis of the rule base, the concept hierarchy derived using FCA is being used to assist the user in deciding whether and where a new concept/proposed-rule fits in with the existing concepts. This can be done by entering a new rule as usual in the MAKE screen in Figure 3.3 and then taking the EVALUATE RULE command button or by entering the pathway on the Test Pathways screen in Figure 3.13. This feature is described under critiquing in Section 7.3.1.

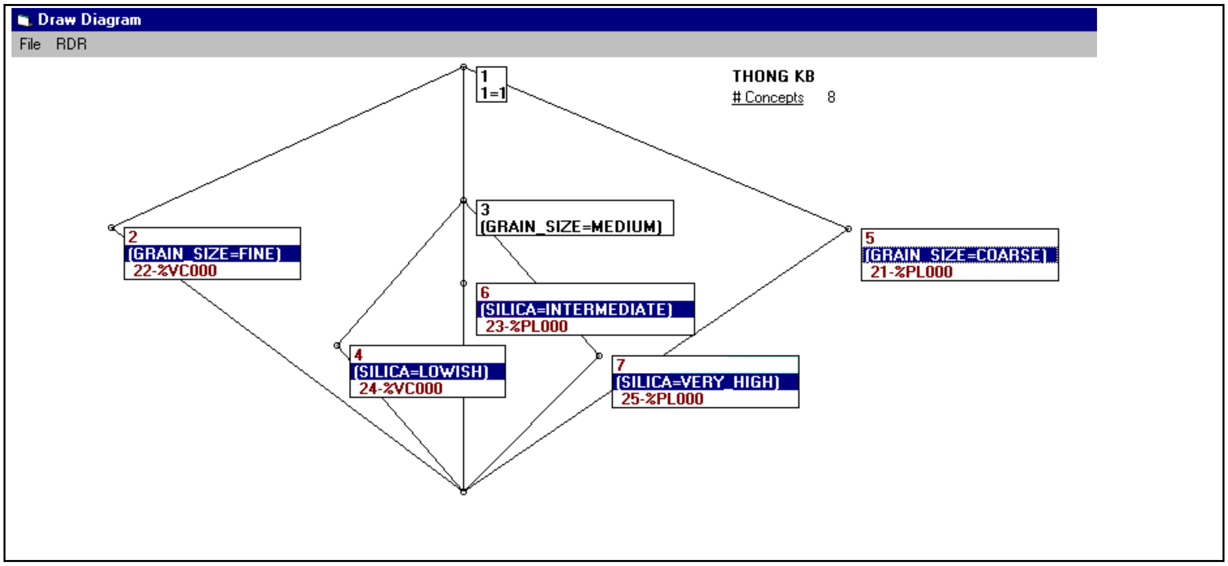


Figure 5.17: The line diagram for the SISPHYUS III domain from the context for the conclusion %PL000 - Plutonic and %VC000 - Volcanic.

This section has made use of the final combined MCRDR KBS developed as part of the RDR contribution to the Sisyphus III experiment. Interestingly one of the other submissions to Sisyphus III used FCA to build line diagrams from the card sorts (Erdmann 1998) but in the approach there was no strategy for handling conflict (which was referred to as *noise*), the next step after developing diagrams was not clear and no KBS were developed. This domain is discussed in more detail in Chapter 6 where the various sources of expertise provided in the KA material were used to test out the RE framework developed. We look next at a configuration task.

5.3.4 The Ion Chromatography Domain - a configuration task

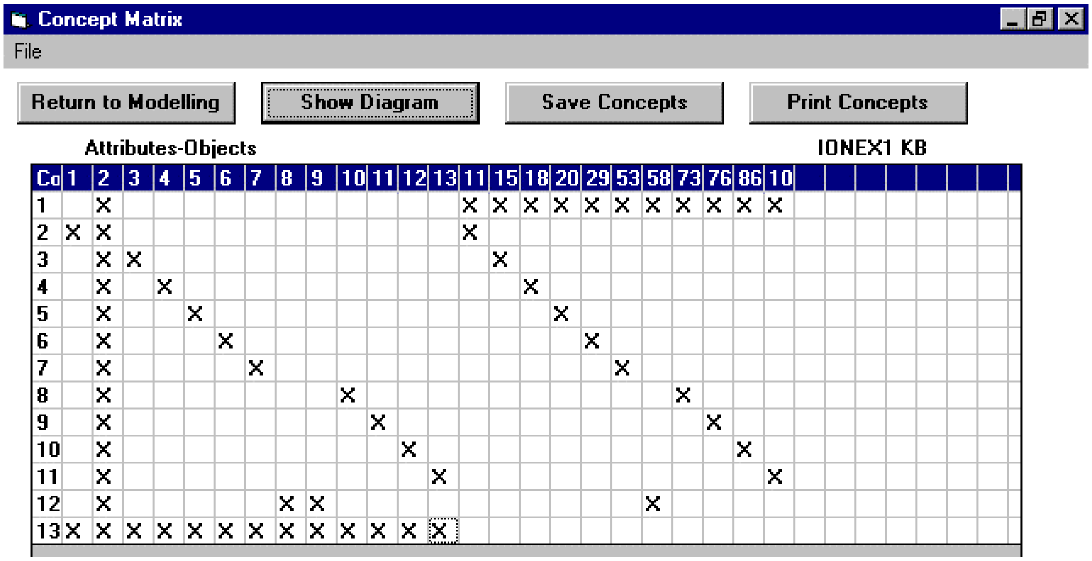
A description of the domain is taken from Ramadan et al (1997b)

> *"Ion exclusion is a well known ion-exchange chromatographic technique for separating strong acids as a class from weak acids using a high-capacity sulfonated ion exchange resin. The ICE mechanism of solute retention is based on the phenomenon that neutral molecules penetrate the resin while the counter ions with respect to the exchange ion are repulsed or, in other words, excluded from it. Therefore by this mechanism acidic compounds can be separated on a cation exchange resin and basic compounds on an anion exchange resin. The aim of the system is to define appropriate conditions for the separation of desired groups of acids or bases".*

The ion chromatography domain is complex with 23 attributes with 2 to 10 values for each attribute. The configuration task itself is also more complex than the task of classification because in addition to parameters (attributes) and values we also need to take into consideration constraints, requirements, preferences and global cost function. For a configuration to be a solution each parameter must have a value, no constraints can be violated and all appropriate requirements must to be satisfied. Inferencing with a configuration tasks is not as simple as with classification as the inference cycle must be repeated a number of times until a solution can be found. This is because the outputs of one cycle will be the inputs of the next cycle until a solution is found (Ramadan et al 1997a).

```
1     1=1~   2-%DEA08~  3-%POS01~  4-%MOA03~  5-%COL01~  6-%MOB01~  7-%DEA08~  8-
      %MOA05~ 9-%MOA01~ 10-%COLP1~ 11-%SUP01~ 12-%DEA01~ 13-%POS02~ 14-%DEA05~
      15-%SUP02~  16-%DEA03~  17-%POS02~  18-%SUP02~  19-%DEA02~  20-%SUP02~  21-
      %PRE02~ 22-%POS02~ 23-%COL01~ 24-%COL03~ 25-%COL05
2     (FORMCOM=MOLYBD)~1=1~    2-%DEA08~ 3-%POS01~ 4-%MOA03~
3     (FORMCOM=MOLYBD)~1=1~(DETECTOR=DIR_SPEC)~  3-%POS01~
4     1=1~(DETECTOR=DIR_SPEC)~       3-%POS01~  100-%SUP02~   101-%ELB04~   106-
      %MOA02~ 108-%POS01~ 109-%COL01~ 113-%MOA02~
5     1=1~(DETECTOR=DIR_SPEC)~(MOBILEPHASE=H2SO4)~     109-%COL01~
6     1=1~(MOBILEPHASE=HNO3)~  6-%MOB01~
7     1=1~(ANY(FORMCOM)=MANNIT)~      7-%DEA08~
8     1=1~(MANNIT(FORMCOM)=TRUE)~     8-%MOA05~ 110-%DEA08~
9     1=1~(MANNIT(FORMCOM)=TRUE)~(YES(CONDCTANCE)=TRUE)~(MOLYBD(FORMCOM)=TRUE)~
             110-%DEA08~
10    1=1~(YES(CONDCTANCE)=TRUE)~     9-%MOA01~ 57-%DEA01~ 80-%DEA06~ 110-%DEA08~
```

Figure 5.18: Partial listing of the concepts found for the Ion Chromatography KBS. The concept number is shown first, then the set of attributes (rule conditions) separated by "~". The set of objects (rule number and conclusion) separated by "~" are shown last.



Co	1	2	3	4	5	6	7	8	9	10	11	12	13	11	15	18	20	29	53	58	73	76	86	10
1		X												X	X	X	X	X	X	X	X	X	X	X
2	X	X												X										
3		X	X												X									
4		X		X												X								
5		X			X												X							
6		X				X												X						
7		X					X												X					
8		X								X											X			
9		X									X											X		
10		X										X											X	
11		X											X											X
12		X						X	X											X				
13	X	X	X	X	X	X	X	X	X	X	X	X	X											

Figure 5:19: The concept matrix for the conclusion family %SUP in the Ion Chromatography KBS.

No changes were necessary to MCRDR/FCA to allow the 154 rule Ion Chromatography KBS to be modelled using the line diagrams since the structure of the knowledge is no different to any other MCRDR KBS. The only difference is the addition of an inference cycle. However, it appears that theories such as rough sets and FCA are more aimed at classification type tasks and the taxonomy FCA develops in the form of a line diagram is in fact a classification tool. This may account for why building explanations of the ion chromatography domain using the line diagrams were not easy. Another more obvious reason for the poor results is that many of the rules only had one condition and thus the number of objects that share an intersection of attributes is often one as shown in the partial listing of the concepts determined for the whole domain in Figure 5.18. We can see this in the example in Figure 5.19 where I chose to look at all conclusions that start with %SUP. Note that attribute number 2 is the default condition 1=1 and is shared by all rules.

To try to find some interesting concepts or higher level abstractions that could be explored further, I had developed a general technique for analysing a KBS that involved generating all concepts for the whole KBS. This full set of concepts was generally too large to look at on the screen so I had the list printed, like the partial list in Figure 5.18. From the printout I would try to find concepts which had a large number of attributes shared by a large number of concepts. Unfortunately in this domain there were virtually no concepts that met that criteria. An exception is shown in Figure 5.20 where some intersecting concepts could be found which could represent abstractions.

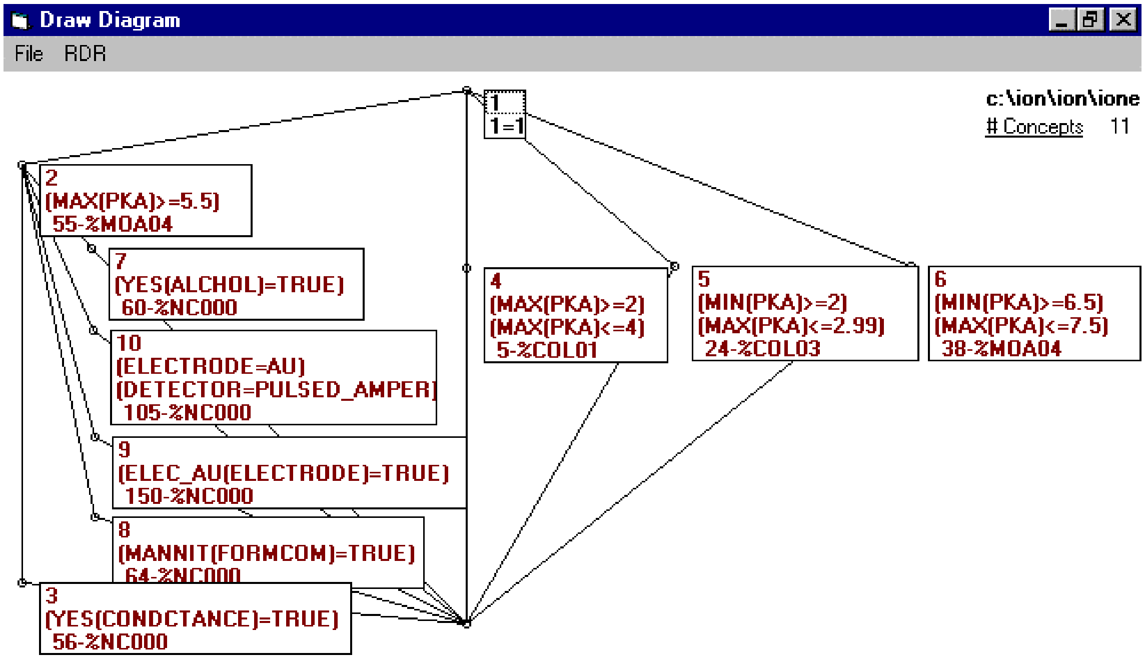


Figure 5:20: The line diagram for the formal context using the rules which have the partial condition MAX(PKA) in the ion chromatography domain.

In Figure 5.20 we can see that the conclusion %MOA04-"MOBILEPHASE = H_2O" should be given if MIN(PKA) >= 6.5 and MAX(PKA) <= 7.5 (concept 6) or MAX(PKA) >=5.5 (concept 2) except if ALCOHOL is TRUE (concept 7) or

ELECTRODE = AU and DETECTOR = PULSED_AMPER (concept 10) or

ELECT_AU(ELECTRODE) is TRUE (concept 9) or

MANNIT(FORMCOM) is TRUE (concept 8) or

CONDUCTANCE is TRUE (concept 3).

As an aside it appears that the condition ELECTRODE = AU is the same as ELECT_AU(ELECTRODE) is TRUE.

MAX(PKA) is also used to determine if column 1 - %COL01 or column 3 - %COL03 should be selected.

On the whole the results from the ion chromatography domain were disappointing and it was difficult to learn anything interesting about the domain.

5.3.5 Summary of the Analyses from Each Domain

The line diagrams from each of the four domains show that MCRDR/FCA can be used on a range of domains. The blood gases KBS was the first domain considered and a number of interesting views of the knowledge were possible. The agricultural and geological KBS also provided sources of multiple expertise and prompted further investigation of the tool for reconciliation of different viewpoints. The combined SISYPHUS III KBS has also been used in the verification studies considered in Section 4.7.2. Generally speaking the MCRDR/FCA tool seems to be a useful method of gaining greater understanding of a domain. Based on the use of MCRDR/FCA in these domains, it appears that classification tasks use more structured knowledge than configuration tasks. At this stage it is hard to know whether the tendency to build rules with one condition was a feature of the expert used in the ion chromatography domain or whether this was a feature of knowledge and KA in that domain. It is also not known whether this domain is representative of configuration tasks in general. The answer to these questions will involve more configuration tasks to be solved using MCRDR and further comparison. However, some reasons why this may have happened are given in Section 8.2.2.4. To assist the reader in determining the usefulness, particularly for instruction, of the various models built in each of these domains using MCRDR/FCA each domain has been analysed using the line diagrams and that analysis is evaluated by domain experts in Section 8.2.2.

5.4 Chapter Summary

This chapter has described how an assertional KBS in the form of an RDR KBS may be interpreted as a formal context which is used by FCA to derive a terminological KBS in the form of an abstraction hierarchy. This finding is significant as it makes possible a whole new way of viewing the development life cycle of KBS and addresses some of the problems associated with developing complex models and domain ontologies.

RDR has been shown to be a successful way of capturing assertions but has not offered a higher level view of a domain. Early investigations of the type of knowledge needed to support multiple uses showed that assertional knowledge alone was not enough but that the abstractions, relationships and structure provided by terminological knowledge was also needed. The concept lattice of FCA provides these features and in a way that allows the existing RDR knowledge to be used without alteration. By retrospectively uncovering the higher models we avoid the bottleneck problems associated with mainstream approaches to building such models plus we have a more reliable means of validating the knowledge we are capturing since the assertional knowledge is based on real cases and observed expert behaviour.

The different selection criteria that could be used to specify which primitive concepts to include in a formal context also provide a powerful knowledge base browsing tool and the ability to see different views of the knowledge. These views can involve a particular area of the domain knowledge which includes all related concepts or can be a view of the knowledge at various levels of abstraction.

The ability to show different views is extended in the next chapter to cover viewpoints from different parties which has enabled the activity of requirements engineering to be added to the range of activities that RDR can support. In Chapter Seven we review how the incorporation of FCA into RDR has facilitated a number of other activities such as critiquing and tutoring. In Chapter Eight the FCA line diagram is evaluated along a number of dimensions. The work reported in the current chapter is the major contribution of this thesis.