Chapter 6

6 Requirements Engineering[1]

The work reported in the previous chapter opened up a new range of activities beyond those that were originally envisaged in this thesis. The concept matrices and lattices developed by MCRDR/FCA can be viewed as conceptual models that were implicitly represented in the simple RDR KBS comprised of conclusions and A-V pairs. By comparing conceptual models we are able to handle multiple sources of expertise and the types of activities that this involves. The work described in Section 5.3.2, which compared the models developed for the four Lotus MCRDR KBS, was the starting point for the work described in this chapter. However, this chapter is concerned with more than the development of multiple models that can then be manually compared. Capturing and using knowledge from multiple sources changes the nature of the questions users may ask and requires a different type of *activity-reuse* compared with the type of activities needed in an individual KBS. A major factor when handling multiple sources of expertise is how to handle the inevitable differences between the individual viewpoints that each KBS represents.

This chapter is concerned with the detection, identification and management of conflict between viewpoints to provide a combined KBS that reflects each individual viewpoint. The management of conflict is essentially a requirements engineering task and thus the framework and enhancement of the MCRDR/FCA tool described here extends the knowledge engineering task to an RE task.

This chapter is an example of increased interest in synergy between KA and RE research. RE is becoming recognised as an important concern in KA research. This chapter provides a comprehensive approach to RE that is implemented using FCA and MCRDR but the framework offered is applicable beyond MCRDR and is applicable to any KR that can be converted to a decision/cross table.

There is much similarity between the needs of knowledge engineering using multiple sources of expertise and those of requirements engineering (RE) from multiple perspectives. To a large extent in KE, the knowledge engineer decides which viewpoint

---

[1] With the exception of section 6.3.3, this chapter follows closely the paper "Richards, D. and Menzies, T. (1998) Extending the SISYPHUS III Experiment from a Knowledge Engineering to a Requirements Engineering Task *11th Workshop on Knowledge Acquisition, Modeling and Management,* Banff, Canada, SRDG Publications, Departments of Computer Science, University of Calgary, Calgary, Canada, Vol 1:SIS-6".

to adopt. The RE approach, by way of contrast, is to negotiate a resolution between the owners of the different perspectives. However, is requirements engineering (RE) a complicated addition to current knowledge engineering? Will the process of rationalising multiple conflicting viewpoints slow down the production of an expert system? If the costs of RE are prohibitive, then RE will rarely be applied.

The RE strategy described in this chapter is a simple extension to the knowledge engineering techniques described in Chapter Five. Given assertions in rulebases (the A-boxes) from different stakeholders, a concept hierarchy (the T-box) is generated and critiqued. Conflicts recognised in the T-box can be used to drive negotiation strategies amongst the different stakeholders. A general framework for this approach, that applies beyond MCRDR KBS, is described together with an implementation using FCA and RDR. This work addresses the situation where we are able to reduce requirements to a set of A-V pairs and thus represents a subset of the total available RE approaches.

This chapter is organised as follows. Section 6.1 introduces RE. Section 6.2 describes the RE framework and an implementation is found in Section 6.3. A case study is presented in Section 6.4 using seven knowledge bases built from the data for the SISYPHUS III (Shadbolt 1996) project. The evaluation technique is described in Section 6.3.6 and is used in Section 6.4.6 to show how the resolution operators have halved the initial degree of conflict. Related work, future work, further discussion and chapter summary are presented in Sections 6.5, 6.6, 6.7 and 6.8 respectively.

6.1 Requirements Engineering: A Review

Requirements engineering (RE) can be defined as "the elicitation and formulation of requirements to produce a specification " Easterbrook (1991, p.8). Current requirements engineering focuses on the maintenance of multiple concurrent viewpoints from different stakeholders (e.g. Easterbrook or Finklestein et al 1994). Here, a subtle re-expression of RE is offered: "RE is the application of negotiation operators to reduce the differences between stake holder viewpoints". The advantage of this re-expression is that it can be mapped into an evaluation program for RE, which are discussed more in 6.3.6 and demonstrated in 6.4.6. While RE also involves other issues such as how to gather initial specifications and formalisation of the end specification the RE issue addressed in this chapter is one of the most difficult issues.

There are a number of reasons why it is important to capture these multiple viewpoints rather than taking the approach that the different viewpoints must be captured into one specification (Easterbrook 1991, Ramesh and Dhar 1992). Tracking multiple perspectives is needed because:

1. Specification errors are often the cause of a poor choice between alternatives during the specification phase. By tracking multiple perspectives a history of the design rationale is provided so that when modifications are necessary they can be made more quickly and based on the background that formulated them in the first place.
2. Many people are involved in projects requiring information to be passed between groups and phases. Individuals will forget and may change over time, subgroups have different roles and viewpoints.
3. Design changes can be replayed and retraced. This has implications for reuse of specifications because if it is understood in what circumstances a certain path should be taken then these steps can be reapplied.
4. A more representative specification can be developed and a better framework for conflict resolution can be provided. The specification acts as both a contract and a communication channel.
5. Ownership is an important issue and by allowing multiple perspectives owned by the originator of that perspective we are more likely to motivate the user to participate in the resolution process.

The concept of a viewpoint corresponds to Finkelstein et al's (1989) formalisation of a viewpoint which includes: a style, area of concern, a specification, a work plan and a work record. This allows for an individual to hold a number of viewpoints and removes some of the problems of equating a particular viewpoint with an individual. Each owner of a viewpoint is called a stakeholder. When managing different viewpoints, conflict between different stakeholders must be handled.

6.2. The RE Framework

The viewpoint management framework has the following five steps that are repeated until all stakeholders are satisfied.

1. Requirements acquisition - Capture each viewpoint in a working knowledge based system (KBS). The KBS is an assertional knowledge base (A-box), which is called the performance system, and a terminological knowledge base (T-box), which is called the explanation system, plus a set of cases. These cases can be divided into historical cases representing true observations from the domain; and hypothetical cases representing some desired functionality. Note that historical cases cannot be doubted while hypothetical cases can possibly be ignored since an imagined case may never occur in the real world.
2. Requirements integration - convert all KBS into a common format.

3. Concept generation - In this phase we add to the T-box for each individual KBS.
4. Concept comparison and conflict detection - Compare the T-boxes of each KBS and detect conflicts.
5. Negotiation - Employ a resolution strategy based on the type of conflict detected in Phase Four. Output of this phase is fed back into Phase One where modifications are performed.

6.3. An Implementation

The general framework in Section 6.2 has not committed to any particular implementation choices. I continue now looking at this framework but within the context of an instantiation. As a result a number of restrictions on the generality of the implementation are imposed and are described in the following subsections.

6.3.1 Phase One: Requirements Acquisition using a Knowledge-Based Approach

Easterbrook (1991) argues that specification can be viewed as a knowledge acquisition process and that many KA tools can be adapted for requirements engineering. One such system, Requirements Apprentice (RA), has been developed by Reubenstein and Waters (1989) which uses a KBS to provide analysis of conclusions and inconsistencies and can also be used for documentation. The knowledge developed can be used further by other tools. The main criticism Easterbrook makes of systems developed to support requirements engineering is that they tend to support single viewpoints only. If we assume that inconsistency and conflict is the norm then this limitation is great. Nii's (1986) solution was to keep the knowledge from each expert separate and treat each KBS as a separate reasoning system.

Like the studies mentioned above, this study takes a KBS approach to requirements acquisition. This section focuses on the situation where the inputs are multiple A-boxes provided from multiple experts, the T-boxes are empty and the set of cases for each A-box is not empty, that is $A \neq \emptyset$, $T = \emptyset$ and $X \neq \emptyset$, where A, T and X denote an assertional KBS, terminological KBS and set of cases, respectively. A further restriction requires that the A-box must be convertible into a decision table. The work in Chapter Four and Five has shown how MCRDR systems can easily be converted into decision tables that do not require mapping of intermediate conclusions into primitive rules (objects). Conversion to a decision table is also suitable for production rule-based systems and has been applied to a number of CLIPS KBS as described in Chapter Eight. In this chapter RDR systems are distinguished from propositional rulebases. The latter are referred to as "standard rules".

This phase is also the maintenance phase for once one cycle is completed it is vital to ensure that the changes made to the explanation system (T-box) output from Phase Five are reflected in the appropriate individual and shared performance systems (A-boxes). The ease of KA, maintenance and mapping to a decision table with MCRDR makes it the preferred KBS approach. Also, as Herlea (1996) points out user involvement is a critical issue in making RE work and system development using RDR has been designed to be performed by the user.

As noted in the general framework, the input to Phase One also includes a set of cases. The importance of the set of cases is twofold. Firstly, on a general level, the cases are used in the negotiations as counterexamples for discussion. Secondly, using the MCRDR approach as described in Chapter Three, or other case-based technique, the cases are also used for initial KA and for modification of other views. For example, as one of the modification strategies outlined below (see Figure 6.4), when a concept is found to be in conflict, the case or cases associated with that concept are passed to the other stakeholders for KA. This should either resolve the conflict or at least ensure that all parties have given their views given the same set of criteria. In Section 6.5 on Related Work it is suggested how cases can be obtained.

6.3.2 Phase Two: Requirements integration

Requirements integration is the process of ensuring that all viewpoints are in formats that can be compared. Adopting the general framework, it may be that viewpoints have been captured using different KR. To avoid the requirement of mapping from all KR's used into all other KR's, that is N^2 mapping schemes, all KRs are converted into one format so that only 2N mapping schemes are needed. As mentioned in Phase One, in the current implementation any representation that maps into a decision table can be used so that a common approach to subsequent phases can be taken. It will become apparent in the next section why the decision table format restriction exists.

6.3.3 Phase Three: Concept Generation

In the general framework, an explanation system could already exist (that is, $T \neq \emptyset$ in Phase One). Alternatively, it could be supplemented or built in this phase. In the current work, we restrict ourselves to the case where $T = \emptyset$. The approach chosen is to begin with a performance system and later derive the explanation system. We start with a set of privately owned and defended A-boxes ($A_1..A_i$) written by some experts ($V_1..V_i$). The knowledge base also includes some data structures generated from previous cycles of the five phases. These structures are:

- One circumvent table for each A-box. This table identifies which rules not to include in future RE sessions.
- One subsumes table for the entire system. This table stores mappings of different terms to a common term.
- One CircumDelayIgnore table for the entire system. This table tags which T-box conflicts have been marked as “circumvented”, “ignored” or “delayed” in the previous cycle.

For more on these tables, see Section 6.3.5.

Easterbrook (1991) points out that one of the reasons why systems fail to meet the user’s needs is that the original mental model of the user has not been captured in the final design model. By letting the user have direct involvement with system development and using a simple KA approach based on the user’s behaviour, the capture and validation of assertions seems more feasible. Using FCA on the KBS rules as described in Chapter Five the leap from A-box to T-box is achieved which is demonstrated in an example below based on the Library World (Finkelstein et al 1994).

In the Library World there is a hierarchy of agents made up of borrowers, staff (which is further subdivided into librarians and clerks), library and catalogue. In this example, only the viewpoints of the borrower, librarian and clerk agents are considered as they can be considered “direct” agents which not only supply or receive information but they also process information (Finkelstein et al 1994). The action table for the “borrower” agent in Finkelstein et al 1994, which is reproduced in Figure 6.1 below, is interpreted as a formal context as shown in Figure 6.2.

SOURCE	INPUT	ACTION	OUTPUT	DESTINATION
Borrower	book	check-in	book	Clerk
	card		card	
Library	book	check-out	book	Borrower

Figure 6.1: An action table elaborating the agent “Borrower” From Finkelstein et al 1994, p. 7.

In Figure 6.2 we have the formal context “Library from the Borrower Viewpoint” with the set of objects G = {Borrower check-in, Borrower check-out, Library check-out} and set of attributes M = {source borrower, source library, input book, input card, action check-in, action check-out, output book, output card, destination borrower, destination clerk}. The crosses show where a relation between the object and attribute exists, thus the set of relations I = {(Borrower check-in, source borrower), (Borrower check-in, input book),…,(Library check-out, destination borrower)}. As described in the previous chapter, each row in the crosstable represents a concept. Using FCA we can form a complete lattice as shown in Figure 6.3.

	source borrower	source library	input book	input card	action check-in	action check-out	output book	output card	dest. borrower	dest. clerk
Borrower-check-in	X		X		X					X
Borrower-check-out	X			X		X	X	X	X	
Library-check-out		X	X			X	X	X	X	

Figure 6.2: Context of "Library from Borrower Viewpoint"

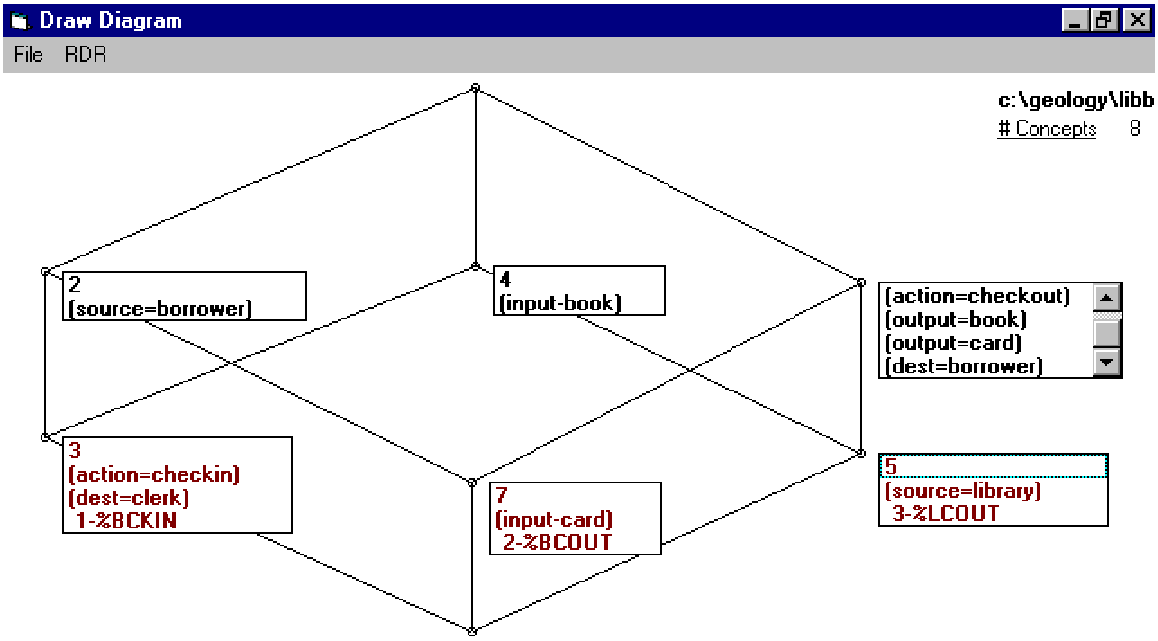

Figure 6.3:The Line Diagram for the Formal Context "Library from Borrower Viewpoint".

A formal context and line diagram may be developed for each of the viewpoints in the Library World. The line diagrams can then be compared to detect differences between viewpoints. This process is considered next.

6.3.4 Phase Four: Concept Comparison and Conflict Detection

A number of researchers offer different sets of conflict types (e.g. Easterbrook 1991 and Schwanke and Kaiser 1988). This study has chosen to use the four quadrant model of comparison between experts developed by Gaines and Shaw (1989) which is used to classify two conceptual models as being in one of four states:

- Consensus is the situation where experts describe the same concepts using the same terminology.

- Correspondence occurs where experts describe the same concepts but use different terminology.
- Conflict is where different concepts are being described but the same terms are used.
- Contrast is where there is no similarity between concepts or the terminology used.

In other sections a broader view of conflict is taken and encompasses inconsistencies that include the states of contrast, correspondence and conflict. Gaines and Shaw's model, however, does offer greater precision in describing the nature of the conflict which is important in deciding how it can be handled. More formally the states of consensus and contrast are defined according to the FCA notion of a concept as a related set of attributes and objects. V denotes a View, P denotes a concept, B denotes a set of attributes and O denotes a set of objects. For an example of each of the four states as they may appear on a concept lattice see section 6.4.4 and Figure 6.9.

Consensus $\{V_1.P_i.B_j\} = \{V_2.P_k.B_L\}$ where $B_j = B_L$ and $\{V_1.P_i.O_j\} = \{V_2.P_k.O_L\}$ where $O_j = O_L$ (6.1)

Contrast $\{V_1.P_i.B_j \cap V_2.P_k.B_L\} = \varnothing$ and $\{V_1.P_i.O_j \cap V_2.P_k.O_L\} = \varnothing$ (6.2)

As shown in (6.1), consensus exists where *all* of the attributes and objects of a concept in one viewpoint match with a concept in another viewpoint. The search for a match is sequential and terminates when a match is found or when all the concepts in the other viewpoint have been compared. Thus in the worst case we need N x N-1 comparisons. Discovering the consensus between conceptual models is important for establishing common grounds from which differences can be viewed. Contrast (6.2) can be viewed as the opposite to consensus and exists when *none* of the attributes or objects in a concept in one viewpoint can be found in any of the concepts in another viewpoint. The search for a contrasting concept terminates when a match on any attribute or object is found or when all concepts in the other viewpoint have been processed.

A concept not in a state of consensus (match found in another viewpoint) or contrast (completely different to all concepts in another viewpoint) is then either in a state of correspondence or conflict. The key to deciding which state it belongs to depends on the terminology. In the approach presented it would be up to the stakeholders to decide whether the terminology used for an attribute or object was the cause for two concepts not appearing at the same node. If more assistance for the user is desired, Gaines and Shaw (1989) have shown that the repertory grid technique, which the crosstable could possibly be mapped into, can be used to identify where terminology is the cause of inconsistency. Having detected various conflicts let us consider how to resolve them.

6.3.5 Phase Five: Conflict Negotiation

Before it can be decided how to fix a detected inconsistency it is necessary to provide a conflict resolution strategy. There are a number of resolution methods which include negotiation, arbitration, coercion and education (Strauss 1978). Negotiation is the most appropriate within the assumed context of parties of equal status and ability. As Easterbrook (1991) points out, a good solution will require creativity and creativity is not something that can be automated. However, since automation is a fundamental goal of requirements engineering research the approach is extended beyond a general, genial chat by offering as much automated assistance for this step as possible.

Each RE researcher appears to use a different set of resolution strategies (e.g. Easterbrook 1991, Thomas 1976). Easterbrook and Nuseibeh (1996) offer five categories that cover the actions that have been found necessary. These are:

- Resolve, correct any errors;
- Ignore, no action is performed;
- Delay, identify the existence of the inconsistency but defer action until a later date;
- Circumvent, identify the existence of the inconsistency so it can be avoided;
- Ameliorate, reduce the degree of inconsistency. This action requires analysis and reasoning.

Resolving conflict will involve performing modifications. If the cause of disagreement is differences in terminology (correspondence in the Gaines and Shaw four state model) then one technique is to up-date all views to conform to an agreed upon set of terminology. This option is probably not satisfactory to the various stakeholders and also means that the history of changes is lost or altered. A simple and more appropriate solution is to use a subsumes table which maps terms from individual views into a shared terminology which is then used for comparison. The use of individual, rather than shared, terminology is a key feature of this work and as is shown in the SISYPHUS III case study later it is important that the terminology used in each source is not altered or reinterpreted. Naturally terminology will need to be reconciled to achieve a reduction in conflict but the choice of what can be treated as the same or similar is left to the stakeholders and is updated in the shared view only. Where a stakeholder feels that their own terminology is superior or more precise but still related to a term in another viewpoint/s it is possible to introduce an intermediate term, which may represent an abstracted concept from the original, to all viewpoints sharing the higher level concept so that the combined viewpoint reflects consensus at the higher but not at the lower level. The original terms in the individual viewpoints are not changed or lost.

Another way in which conflict may be resolved is through the addition or deletion of attributes or objects. The conflict may be due to the set of attributes or objects being only partially shared by another concept. To bring these concepts into a state of consensus it may be decided to drop or add attributes or objects. As mentioned in Phase One, part of the automated support for negotiation is the ability to produce a case associated with the object (rule) that is in question. The cases associated with all objects that can be reached by downward paths are also relevant to the discussion. The closer, distance measured in number of objects separating the two nodes, the object is to the concept in question the more relevant the case should be considered. New attributes or objects could also be added by showing the associated case to the other party and using that case for KA. Alternatively, if a hypothetical case is shown to be impossible, then the rules based on this case should be dropped. The decision of what action to take is made in this last phase and performed in Phase One. In Figures 6.4(a) and (b) a summary of the strategies applicable to standard rules and MCRDR is provided. Note that:

- In each strategy for handling attributes or objects from standard rules, the requirement has been added that some sort of checking should be done after each change to ensure that there are no unknown side effects elsewhere in the rulebase. With MCRDR this checking is given for free because no previously correctly classified cases can become misclassified with the RDR approach to KA and the exception structure.

- RDR has a very tight link between rules and cases. This link is not defined for standard rule bases. Hence, handling case addition/deletion is not defined in standard rule bases (see Figure 6.4(b)).

-

	Add Attribute	Delete Attribute	Add Object	Delete Object
Standard Rules	Add attribute to rule. Check effect on other rules.	Remove attribute from rule. Check effect on other rules.	Add new rule. Check effect on other rules.	Remove rule. Check effect on other rules.
MCRDR	Defined - use existing KA approach	Add stopping rule to rule in error. Add new rule at top level with the old rule minus the attribute to be removed.	Perform KA using the case associated with the concept which has the desired object.	Add a stopping rule.

Figure 6.4(a) The resolution strategy for handling attribute and objects.

	Adding Real Cases	Deleting Real Cases	Adding Hypothetical Cases	Deleting Hypothetical Cases
Standard Rules	N/A	N/A	N/A	N/A
MCRDR	Show to all	ILLEGAL	Show to all	Drop related rules - check refinement rules if they should be dropped or stopped and a new rule, without the dropped rule conditions, added.

Figure 6.4(b) The resolution strategy for handling cases.

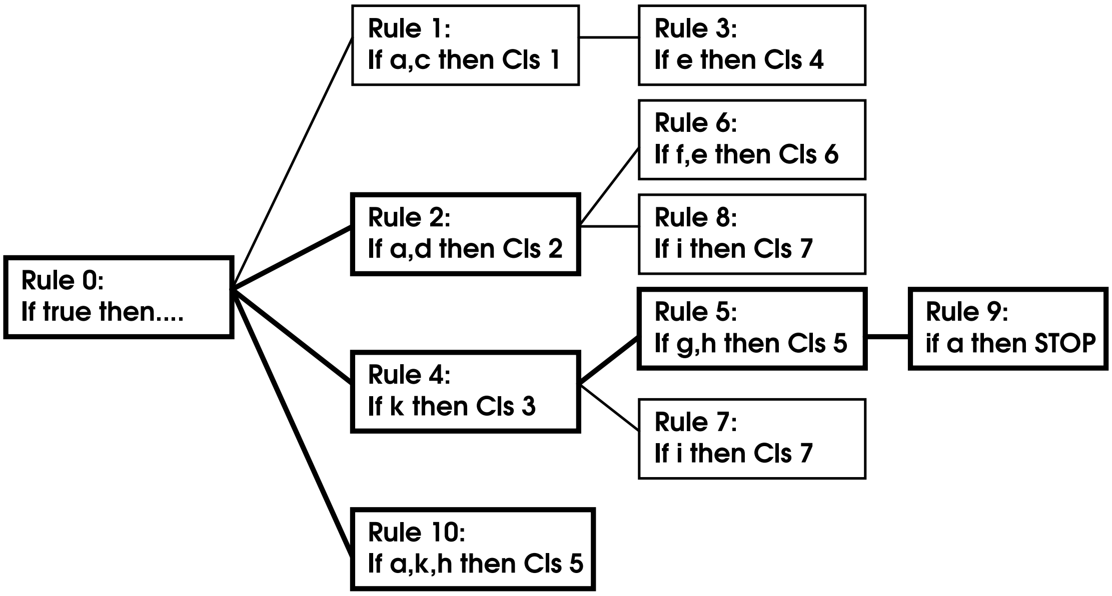

Figure 6.5. An MCRDR KBS with stopping rules.
This KBS is a modified version of Figure 3.2 which demonstrates the use of the stopping rule. The highlighted boxes represent rules that are satisfied for the case {a,d,g,h,k}. We can see that there are two independent conclusions for this case, Class 2 (Rule 2) and Class 5 (Rule 10). Rule 5 had been the cause of a conflict between viewpoints. To resolve this conflict it was decided that attribute g should be dropped. As described in Figure 6.4, the STOPPING RULE is used to say that this pathway should not fire, so even though the case satisfies Rule 5 that rule is stopped from being reported. We can see that Rule 10 now replaces the rule pathway for Rule 5 dropping the attribute g.

An example of how one attribute may be dropped and another added using MCRDR and the resolution operators is shown in Figure 6.5 which shows the important role of stopping rules to facilitate these changes. In Figure 6.5 the user has decided to add the attribute “a” and delete the attribute “g” to the concept associated with rule 5. The original rule 5 has been stopped and is replaced by rule 10.

The negotiation strategies can be strengthened by offering filtering rules which guide the dialogue between the stakeholder and the system. One filter is the use of preference criteria to guide the stakeholders in deciding what part of the view should be considered first as a candidate for change. Taking the concept lattice structure into consideration (where objects belonging to a concept are reached by descending paths and attributes are reached by ascending paths) it can be said that if an attribute at the bottom of a pathway or an object at the top of a pathway is to be removed then no other concepts will be affected and this act can be performed without further investigation. This strategy can be useful for example, if it had been agreed upon that only one of two attributes were necessary and one was at the bottom of a pathway and the other higher up, then it would be advisable to remove the lower attribute. Other templates or KA scripts (Gil and Tallis 1997) can be used to guide the user with revising their KBS and ensuring that each of the rules related to the change are modified and tested.

The last four resolution strategies are relevant for situations in which a complete resolution can not be negotiated and each one has its appropriate usage. For example, ignoring is a useful strategy where the issue is not that important or pursuing it is not worth the effort or harm it may cause to the end solution. These approaches can be termed as living with inconsistency or 'lazy' consistency (Narayanasway and Goldman 1992) and can be compared to fault-tolerant systems that continue to function after non-critical failures occur. It has been argued that enforcing removal of all inconsistency "constrains the specification unnecessarily" and "tends to restrict the development process and stifle novelty and invention" (Finkelstein et al 1994, p.2 & 4). They see that consistency is necessary within a viewpoint but partial consistency between viewpoints is allowable.

This study also accepts that living with inconsistency will be necessary and uses tags to identify the status of the conflict. The use of tags is similar to the use of "pollution markers" (Balzer 1991) that act as a warning that code may be unstable or that the users should carefully check the output. Pollution markers can also be used to screen inconsistent data from critical paths that must have completely consistent input. If it is the concept that is being circumvented, ignored or delayed, the concept is marked in the shared T-box since there is not necessarily a one-to-one correspondence between rules and concepts. This updated T-box is used as input in the next T-box generation. When it comes to rules (the A-box) the only tagging strategy offered is circumvention. Tagging a rule as "circumvented" in the individual A-boxes is used to avoid certain unstable parts of the requirements. When the new T-boxes are generated these rules will not be included.

The resolution strategies shown in Figures 6.4(a) and (b) are also applicable to the strategy of amelioration. However, the result is not consensus but a reduction in the degree of the conflict. Amelioration results in bringing concepts closer together. If we think in terms of the concept lattice the result will be a shortening in the distance between the two concept nodes.

6.3.6 Evaluation

To evaluate that our RE operators are reducing the amount of conflict between viewpoint we need two sub-routines:

- A "distance metric" which can assess how far apart are two viewpoints.
- A set of "negotiation operators" which extract features from the viewpoints that could lead to reducing the distance metric.

These RE sub-routines can now be evaluated as follows. Different negotiation operators can be assessed via their observed effects on the distance metric: good RE operators have a big impact on this distance. Operators also have a construction cost: good operators should be easy to build. An ideal RE operator is cheap to build, cheap to use, and reduces the distance metric considerably (though, in practice, some trade-offs may be required between operator complexity and operator effectiveness).

The negotiation operators already defined are used to compare the degree of conflict before and after. By computing a score for each concept in each viewpoint compared to each other viewpoint and taking the total of these scores it can be checked that the degree of conflict after the RE process is less than at the start. A score of 0 is assigned to a concept found to be in a state of consensus with a concept in another viewpoint, since the distance between them is zero. For concepts in a state of contrast (no partial or complete match in the other viewpoint) a score of 1 is assigned, which is the same result as if the conflict measure was used since the number of attributes not shared divided by the number of attributes is equal to one. These measures assume that the two concepts being compared share the same object (conclusion). If they do not then it appears that they are not meant to represent the same concept so that comparison is not meaningful. Concepts in a state of correspondence are treated the same as concepts in a state of conflict since the reason for conflict depends on user input and the focus here is on the size of the difference not the cause. Once terminology differences are reconciled such concepts will move into one of the other states and be handled accordingly.

These measures have been used in Section 6.4.6 to determine how well the RE strategy is working on the SISYHPUS III viewpoints. Since in the next section we restrict ourselves to looking at one conclusion at a time in all viewpoints, the measures described are appropriate. Of greater difficulty is determining the degree of conflict if it is not known which concept in the other view represents a similar concept. It is necessary to either get the experts to label their concepts so that, for example, two concepts labelled “cat” would be compared or to compare the concept in one viewpoint with all others in the other viewpoint finding the one that shares the largest set of attributes on the assumption that it is the concept which most likely represents the same concept. For the purposes of the example in the next section it was not necessary to resolve this problem and further work on distance measures is part of the future RE work.

6.4 Requirements Engineering using SISYPHUS III: A Case Study

It is argued in this chapter that the needs of knowledge engineering using multiple sources of expertise and those of requirements engineering (RE) from multiple perspectives are very similar and that an RE approach to management of conflict is superior to the knowledge engineering approach that relies on the KE to resolve differences. The SISYPHUS III rock domain, in particular, is a good example of a domain where there is little consensus of opinion in the actual classification of individual rocks. Many rocks are classified differently, with some experts arguing that certain rock classifications do not exist but are simply different examples of another rock classification whose appearance has been altered due to the formation process they have undergone. The disagreement amongst experts and problems that this causes is evident in a number of approaches to solving the SISYPHUS III problem. For example, the KADS (Jansen, Schreiber and Wielinga 1998) solution was to abandon the data provided and to start again using other sources such as books and web-based tutorials. However, there are also differences between these sources which they have endeavoured to reconcile by becoming experts in the domain and making their own decision as to which viewpoint to adopt. As an example, Jansen, Schreiber and Wielinga (1998) point out that the rock Adamellite was missing from three key sources but that eventually they managed to find “an exact description”. So they have decided to accept the viewpoint of the source containing a description but it appears that if such an exact description were well accepted it would not have been so hard to find. It is interesting to note that this happened to be the rock chosen for many of the examples that follow. With such inconsistency within the domain, the data provided by the SISYPHUS III appeared to provide a very suitable testbed for the RE approach presented in Sections 6.1-6.3. We look first at the system requirements of the SISYPHUS III experiment.

The problem statement and scope of SISYPHUS III is given as (Shadbolt 1996):

> *During the Apollo moon program it came to be recognised that many of the primary scientific objectives of the missions could not be achieved without the astronauts acquiring a certain level of geological competence. This included the ability to undertake the collection and documentation of rock samples from the lunar surface. All of the moonwalkers except one - Harrison Schmidt - were taught their geology in snatched excursions with field geologists to selected sites in the USA, and via a limited number of class based lectures. We can expect that future manned missions to the moon and other planets will also contain non-specialists who will be expected to carry out similar tasks.*
>
> *In Sisyphus III we are building a geological expert system for rock sample characterisation. This is intended to act as a tutorial aid and diagnostic decision support system for the trainee astronauts. The final run-time system will be expected to run on a high end colour laptop computer. The initial requirement is that a system be ready for preliminary trials within four months. It must be capable of identifying major types of igneous rocks. Igneous rocks are materials that have solidified from*

molten or partially molten material. These were probably the first formed portions of the earth's surface. They have also provided most of the components of the other two rock types - sedimentary and metamorphic rocks. Igneous rocks are normally first studied in the field and then in the laboratory. However, in this case it has been stipulated that the users should be able to use the system in conjunction only with a hand specimen of the rock, a hand lens and a prepared thin section of the rock that can be viewed using a geological microscope. The aim is to support the description and discrimination of the major types of rock along their most salient and pertinent characteristics.

The problem description is extended a little further for the purposes of this case study. The KA material supplied with the first phase includes: 5 card sorts, 4 laddered grids, 5 structured interviews, 4 self reports and 4 repertory grids. The card sorts and repertory grids provide useful frameworks that require the least amount of data extraction compared to the other techniques that use natural language. The problem description is extended beyond the development of a system that classifies a set of igneous rocks correctly and offers an explanation of the conclusion to the question of how to use data from multiple and conflicting sources of expertise to build this system. These multiple sources of expertise impose a further requirement not explicitly stated in the problem statement above, which is to develop a strategy for handling any differences.

The RDR SISYPHUS III submission produced a number of KBS from a combination of the above sources. The systems were built by a geological and KE novice in the same manner that has been used in this study which is described in section 6.4.1. Differences in terminology, measurement and inconsistencies between the sources were reconciled using the rule of thumb “choose the option that most experts agreed on”, where each KA source was treated as an expert. This is a reasonable heuristic but one that can not always be applied since in some cases it requires interpretation to decide that two concepts match. Thus, to some extent the main final system is a product of what the novice KE thought was correct. The approach outlined in this chapter for reconciling conflict in multiple viewpoints appeared to provide a better way of arriving at a final though not complete model that took into account each perspective without the bias of the KE as the method of resolution. It is proposed that even without the involvement of the stakeholders in the resolution process the RE framework offered is a more methodical approach to conflict resolution than trying to handle conflict when and if it is detected.

In the next subsections the five phases already presented are used to show how the proposed framework can be utilised. Note that the resolution process is conducted on a small scale, that is, the conclusions are compared one at a time rather than comparing all the complete models in one go. The main reason for this is the more information being dealt with, in particular represented visually, the more incomprehensible the diagram becomes. While this approach may be seen as time consuming it is not necessarily the most inefficient because problems can often be better identified and resolved when broken down into smaller problems. An expected benefit of this incremental approach is that as conflicts are resolved, for example differences in terminology are reconciled through the subsumes table or concepts are tagged to be ignored, there will be progressively less errors or differences.

6.4.1 Phase One - The Data and Knowledge Acquisition

```
#1 10-03-1997 09:16:57
GRAIN_SIZE
  COARSELY_CRYSTALLINE
SILICA          VERY_HIGH
COLOUR LIGHT
OLIVINE UNLIKELY
MICA            LIKELY
#2 10-03-1997 09:16:57
GRAIN_SIZE      FINE
SILICA
  INTERMEDIATE
COLOUR FAIRLY_LIGHT
OLIVINE         UNLIKELY
MICA            UNLIKELY
#3 10-03-1997 09:16:57
GRAIN_SIZE      FINE
SILICA          BASIC
COLOUR DARK
OLIVINE LIKELY
MICA            UNLIKELY
```

Figure 6.6: Extract from Card Sort 5 Case File.

To reconcile the viewpoints contained in each of the data sources mentioned above, The KE novice developed six individual KBS using MCRDR for each of the card sorts and laddered grids. For the purposes of this case study the other material was not used even though supporting or extra information had originally been gleaned from them and added to the final KBS. It was important that the models should only represent the data directly specified as related to them and avoid inclusion of the KE's own interpretation[2]. The card sorts contained differences in the number of attributes used, the attributes chosen, the values assigned to attributes and the terminology of attributes and objects. It was these types of differences I wished to reconcile. One alternative is to force a common set of terminology and scales to be used. However, it is important to stakeholders/experts that they be allowed to express their models in their own words. This is supported by the work on repertory grids, where subjects preferred to use their own constructs over ones supplied to them (Shaw 1980).

The data in the card sorts were used directly to develop cases. In Figure 6.6 the first three records in the Card5 case file can be seen. For the restricted study performed in this chapter, a case file was developed only for Card Sorts 1, 2, 3 and 5 and Laddered

[2] As stated here it was the aim of this study to avoid interpretation by the KE. Unfortunately, it was found later that the KE had made certain decisions regarding the data. For example, in Card Sort 5 Adamellite, Dacite, Granodiorite and Granite had been described by the expert using identical features. The KE had decided to drop Dacite and Granite and to change a feature of Granodiorite. This was the type of situation I had wanted to avoid. If all four rocks had been classified using the same features they would have appeared at the same node and have shown that either an error had been made and/or additional condition/s were needed to differentiate between the four rocks. In this way our modelling technique would have been useful for identifying not only conflict between but also conflict within a viewpoint.

Grids 1 and 3[3], and will be referred to as C1, C2, C3, C5, L1 and L2, respectively. Using the appropriate case file, KBS were developed by running an inference on each case. If the conclusion given by the system was incorrect, a new conclusion was assigned and the attribute value pairs which justified that conclusion were selected as the rule conditions. The conditions were selected based on intuition and the *difference list* that shows the differences between the current case and the case that gave the misclassification.

```
0
11
12
1 0 11 0  %NC000 : 1 = 1
2 1 0 0   %AD000 : (GRAIN_SIZE = COARSELY_CRYSTALLINE) & (SILICA = VERY_HIGH) &
                   (COLOUR = LIGHT)
3 1 0 2   %AN000 : (GRAIN_SIZE = FINE) & (SILICA = INTERMEDIATE)
4 1 0 3   %BA000 : (GRAIN_SIZE = FINE) & (SILICA = BASIC)
5 1 0 4   %DI000 :(GRAIN_SIZE=COARSELY_CRYSTALLINE) &(SILICA=INTERMEDIATE) &
                   (COLOUR=FAIRLY_LIGHT
6 1 0 5   %DO000 : (GRAIN_SIZE = NOT_COARSE) & (SILICA = BASIC)
7 1 0 6   %DU000 : (GRAIN_SIZE = COARSELY_CRYSTALLINE) & (SILICA = ULTRABASIC) &
                   (COLOUR = DARK)
8 1 0 7   %GA000 : (GRAIN_SIZE = COARSELY_CRYSTALLINE) & (SILICA = BASIC) &
                   (COLOUR = DARK)
9 1 0 8   %KE000 : (GRAIN_SIZE = ?)
10 1 0 9  %GR002 : (GRAIN_SIZE = NOT_COARSE) & (SILICA = VERY_HIGH)
11 1 0 10 %RH000 : (GRAIN_SIZE = FINE) & (COLOUR = LIGHT)
```

Figure 6.7: The rule file for Card Sort 5.

The first rule number, the total number of rules and the number of cases associated with the rules are given in the first three lines. The default rule is on line four. The structure of each rule is rule number, parent rule number, child rule number, sibling rule number, conclusion code and the conjunction of conditions appear after the colon.

The MCRDR rule file developed from the Card5 case file is shown in Figure 6.7. For the rest of this case study, each KBS is treated as a different expert viewpoint which is in fact the origin of each card sort and laddered grid (Shadbolt, personal communication).

4.2 Phase Two: Requirements Integration

As described in the framework, each KBS is converted into a common format, specifically a crosstable or decision table. Figure 6.8 shows the crosstable, which is a formal context, for the first four MCRDR rules for Card Sort 5.

[3] To make the examples using the conclusion %AD000- Adamellite in Figures 6.9, 6.11 and 6.13 more complete, I developed a partial KBS for Card Sort 4 to cover that rock so that seven viewpoints are actually compared in the example.

	1=1	grain_size= coarsely_crystalline	silica= very_high	colour= light	grain_size= fine	silica= intermediate	silica= basic
1-%NC000	X						
2-%AD000	X	X	X	X			
3-%AN000	X				X	X	
4-%BA000	X				X		X

Figure 6.8: Crosstable of the first four MCRDR rules for Card Sort 5

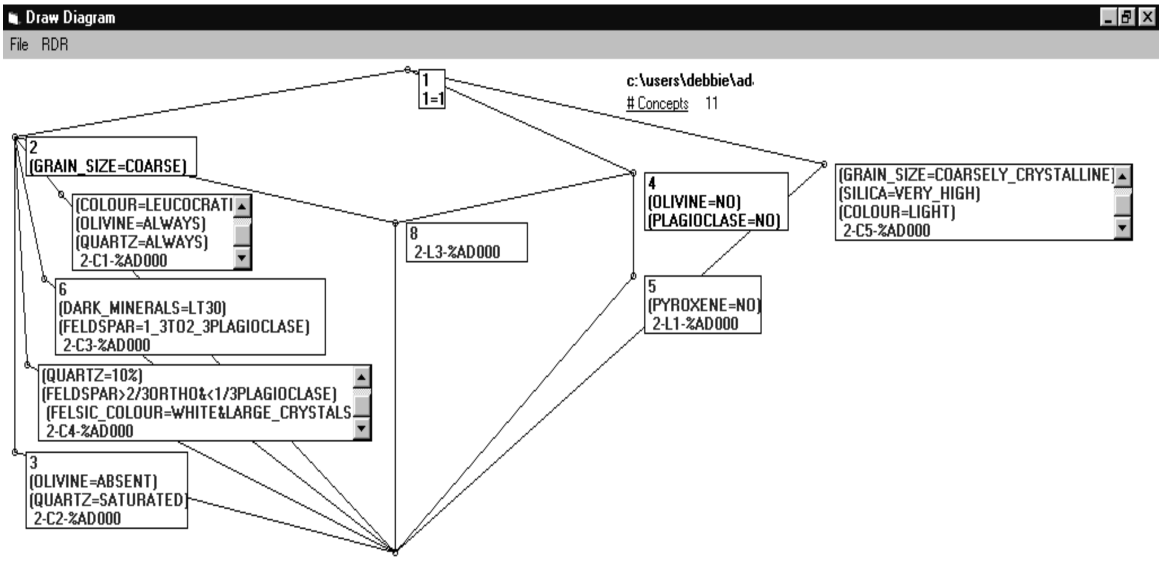

Figure 6.9: The Concept Lattice for the Conclusion %AD000- Adamellite based on seven KBS.

6.4.3 Phase Three: Concept Generation

In this phase the crosstable is used to derive the ordered set of formal concepts which can be shown as a concept lattice. When dealing with multiple KBS there are two main alternative approaches to implementation of this phase. The crosstables or formal contexts output from Phase Two can be combined into one formal context with the source of each row annotated[4] for identification OR the concepts for each formal context can be derived and then compared. The simplest method with the existing MCRDR/FCA tool is to combine all the individual formal contexts into one formal context and perform queries or comparisons on them as a whole. However, handling such a large context has its problems. As noted before: “the diagrams for activities of any reasonable complexity become very difficult to visualise and understand” (Gaines and Shaw 1993a, p. 59). The MCRDR/FCA options described in Section 5.2.1 offered

[4] The approach to annotation was simply to add the source identification to the beginning of the conclusion code in column one of the crosstable so that in Figure 6.10 the first conclusion would read 2-C5-%AD000 which stood for rule 2, source Card Sort 5 and conclusion code %AD000.

13 (this number includes the choices within the options) combinable options that allow the user to narrow their focus of attention to selected part/s of interest in the knowledge base. In order to ensure that some parts of the KBS are not missed which may result in some conflict being undetected, it is important to select views that will cover all the knowledge in all the viewpoints. Since the major goal of the SISYPHUS III system is to correctly classify a particular rock it seems reasonable to explore each conclusion from the combined viewpoints to see what conflict exists. Figure 6.9 shows the concept lattice for the conclusion %AD000- Adamellite.

6.4.4 Concept Comparison and Conflict Detection

The concept lattice in figure 6.9 may be analysed and the nature of the differences between concepts determined. The case study described here involved the manual detection of conflicts from the line diagram. However, as described briefly in Section 5.2.2, a tool was also developed which allowed up to four viewpoints to be loaded, which were used to build two different line diagrams. One diagram showed all the concepts shared by the combined viewpoints and the other diagram showed the complement of the shared concepts. The idea behind this was that it was an easy and quick way of identifying common ground and what concepts/terminology needed to be negotiated. These diagrams are not used here as the aim is display the combined viewpoints with shared and unshared concepts and to show how the distance between viewpoints can be reduced.

Looking at Figure 6.9, we can see that the C5 viewpoint for Adamellite is in a state of contrast since none of its attributes are shared with any of the other viewpoints. However, some of these differences appear to be terminology related. There is consensus that GRAIN_SIZE=COARSE between C1, C2, C3 and L2 but in C5 the GRAIN_SIZE=COARSELY_CRYSTALLINE. This appears to be a correspondence type of conflict because of differences in terminology. There appear to be other correspondence errors. The attribute QUARTZ is used in C1, C2 and C4 with the values ALWAYS, SATURATED and 10%, respectively. The value of OLIVINE in L1 and L2 is NO and in C2 the value is ABSENT. In C5 the COLOUR=LIGHT and in C1 the COLOUR=LEUCRATIC. The dictionary meaning of “leuco” is white (Macquarie Dictionary), so it appears that the terms in these two concepts are compatible. It also appears that the feature DARK_MINERAL=LT30 indicates a lightness of colour. The differences in the terminology used for the values assigned to GRAIN_SIZE, QUARTZ, OLIVINE and COLOUR can be reconciled by using the subsumes table to map to a common term as shown in Figure 6.10.

The value assigned to OLIVINE in C1 is ALWAYS and represents a conflict where the terms are compatible but the concept is obviously the opposite to the concepts in L1, L2 and C2. There is consensus between L1 and L2 that PLAGIOCLASE=NO but these concepts conflict with the concepts FELDSPAR=1_3TO2_3PLAGIOCLASE for C3 and FELDSPAR>2/3ORTHO&<1/3 PLAGIOCLASE for C4. These various conflicts need to be resolved which takes us to the next stage.

6.4.5 Phase 5: Negotiation

Original Term	Synonym
GRAIN_SIZE=COARSLY_CRYSTALLINE	GRAIN_SIZE=COARSE
QUARTZ=SATURATED	QUARTZ=ALWAYS
QUARTZ=10%	QUARTZ=ALWAYS
OLIVINE=ABSENT	OLIVINE=NO
COLOUR=LEUCRATIC	COLOUR=LIGHT
FELSIC_COLOUR=WHITE&LARGE_CRYTALS	COLOUR=LIGHT
DARK_MINERAL=LT30	COLOUR=LIGHT

Figure 6.10: The Subsumes Table

Reconciliation of conflicts in terminology is a good place to start in the negotiation process. Those terms identified to be similar are put in the Subumes Table, see Figure 6.10. As noted in Section 6.3.5 individual terms may be kept and a new shared term is introduced in the situation unless there is agreement to change. With the updated Subsumes Table phases 1-5 are repeated. In Figure 6.11 the effect of the subsumes table on the original conflict in Figure 6.9 can be seen.

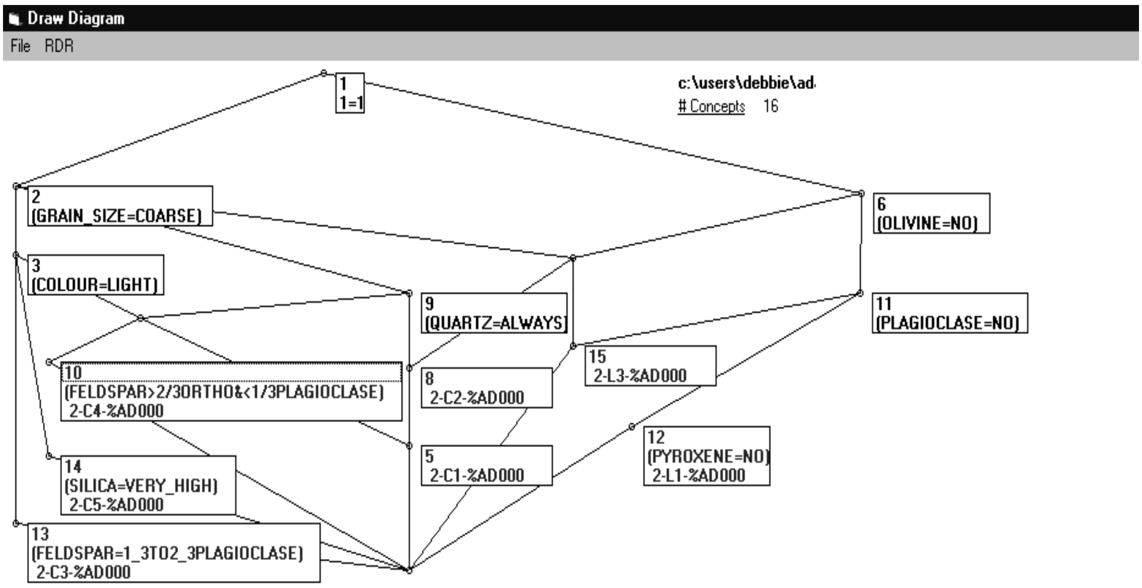

Figure 6.11: The updated Line Diagram after applying the Subsumes Table in Figure 6.10.

In addition to terminological differences there are some conceptual differences. The following scenario simulate how some of these differences can be resolved:

- Since sources C2, L1 and L2 agree that OLIVINE=NO is relevant, the expert in C1 decides that they have made an error and changes OLIVINE=ALWAYS to OLIVINE=NO. As described in Figure 6.4(a) this is done by adding a stopping rule to the incorrect rule and adding a new rule with all of the previous conditions but with the value of OLIVINE changed. This also requires amending the value of OLIVINE in the associated hypothetical case or passing a case from another viewpoint which covers this situation.
- To ignore or delay a concept the Ignore Tag Code "I" or Delay Tag Code "D", respectively, is used as shown in Figure 6.12. This Tag Code is used to update the CircumDelayIgnore table and shows the concept in the concept lattice annotated with the word "IGNORE" or "DELAY". Fig 6:13 shows two delayed concepts.
- The L1 expert agrees that GRAINSIZE=COARSE should be included so they amend their KBS by adding an exception rule to Rule 2, as outlined in Figure 6.4(a), where the conclusion is given as %AD000 and the condition as GRAINSIZE=COARSE.
- A feature of the approach is the ability to offer counterexamples that can be used in negotiations. In Figure 6.13 the case associated with Concept No. 8 (Rule 2 in Card Sort 5) is shown to the group to assist with reconciliation of this conflict. The user takes the Show_Case option from the RDR menu and specifies the source (which KBS) and the rule in which they are interested. The C5 and L1 experts can not be persuaded by the other experts to drop the attributes SILICA=VERY_HIGH and PYROXENE=NO, respectively, so it is decided to delay resolution of these conflicts until a later date. This is achieved by using the Ignore Tag which drops this concept and updates the CircumDelayIgnore table, see Sections 6.3.3 and 6.3.5. The user may also defer action on a conflicting concept by using the Delay Tag which updates the CircumDelayIgnore table as above but rather than dropping the concept, the control background and foreground colour is reversed and the word DELAYED is displayed as in Figure 6.13.
- To circumvent an object (rule) the related rule is flagged before a context is generated. In these examples all rules that conclude %NC000 (the no conclusion rule) have been circumvented.
- A final strategy concerns the handling of the controversy over the importance of FELDSPAR in determining if a rock is Adamellite. Expert C3 believes that the FELDSPAR content is one to two-thirds PLAGIO-CLASE. Expert C4 believes FELDSPAR is less than one-third PLAGIO-CLASE, experts L1 and L2 believe that PLAGIOCLASE = NO and experts C1 and C2 do not consider FELDSPAR or the PLAGIOCLASE content. It is thus decided to circumvent the concepts with these

attributes. This is achieved by selecting Concepts No 10, 11 and 13 in the CMATRIX screen, see Figure 6.12, and adding a "C" tag for circumvent. The concept and the tag are written to the CircumDelayIgnore Table. Once given this tag, a concept is not included in determination of the list of predecessors (parents) and list of successors (children) which are used to layout the line diagram. It can be seen in Fig 6.13 that these attributes are no longer shown. If desired, these concepts can be reinstated and shown.

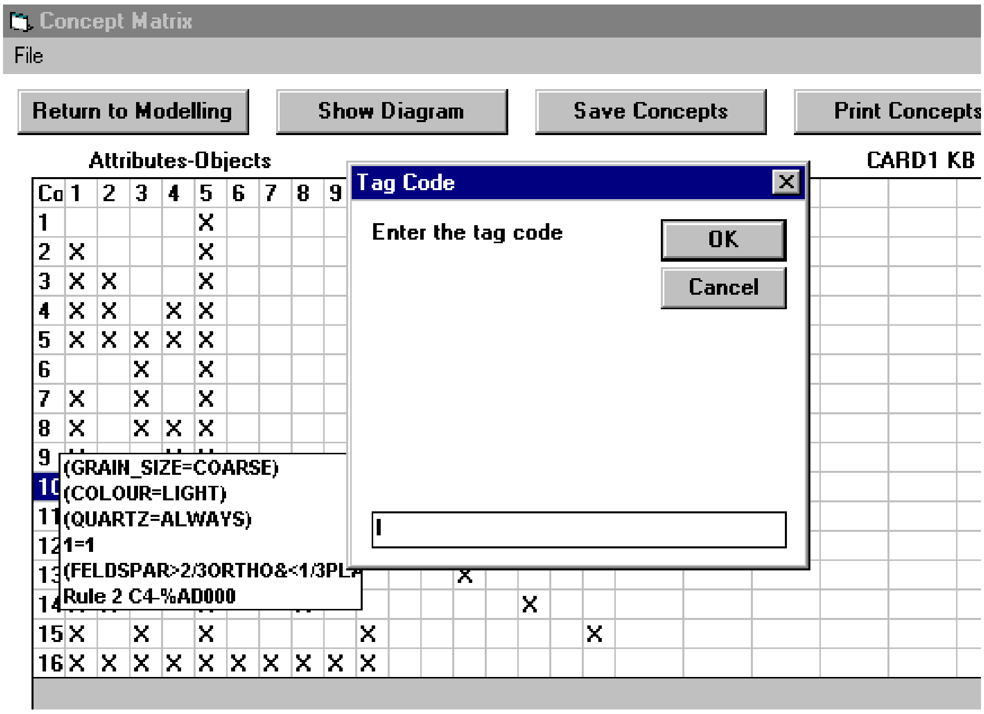

Figure 6.12: Tagging concepts to be "ignored" on the CMatrix Screen. This is achieved by clicking on the Concept No and entering the Tag Code into the pop-up InputBox.

All of the changes mentioned above are reflected in the final diagram in Figure 6.13. Note that even though the number of concepts has only reduced by 1, the concepts are much less complex. In figure 6.9 GRAIN_SIZE= COARSE offered the most, but not total, point of agreement. Now all views agree with this and there are more attributes shared by viewpoints which previously only appeared in one viewpoint. As shown visually in Figure 6.13, the viewpoints in Card Sorts 1,2,3,4 and 5 are more similar to each other than the viewpoints in Laddered Grids 1 and 3 which are similar to each other.

As shown in the diagram in Figure 6.13 each of the viewpoints can be summarised as:

C1- GRAIN_SIZE=COARSE;COLOUR=LIGHT;OLIVINE=NO;QUARTZ=ALWAYS
C2- GRAIN_SIZE=COARSE; OLIVINE=NO;QUARTZ=ALWAYS
C3- GRAIN_SIZE=COARSE;COLOUR=LIGHT
C4 - GRAIN_SIZE=COARSE;COLOUR=LIGHT;QUARTZ=ALWAYS
C5- GRAIN_SIZE=COARSE;COLOUR=LIGHT;SILICA=VERY_HIGH(DELAYED)
L1- GRAIN_SIZE=COARSE;OLIVINE=NO;PLAGIOCLASE=NO;PYROXENE=NO(DELAYED)
L3- GRAIN_SIZE=COARSE;OLIVINE=NO;PLAGIOCLASE=NO

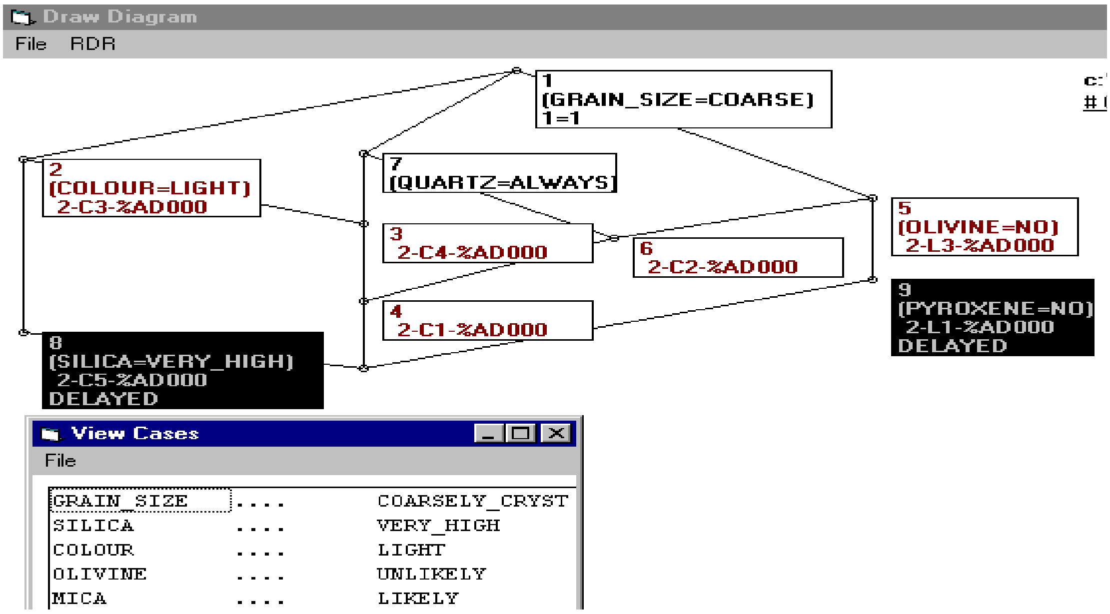

Figure 6.13: The final Line Diagram screen from this round of negotiations. This diagram includes all seven viewpoints (C1, C2, C3, C4, C5, L1 and L3) for the classification of the rock Adamellite. Some attributes have been dropped or added to views, concepts have been tagged to be circumvented (not shown) or delayed (shown). The case for concept 8 is shown to assist with negotiation. There is considerably less conflict now than in Figure 6.9.

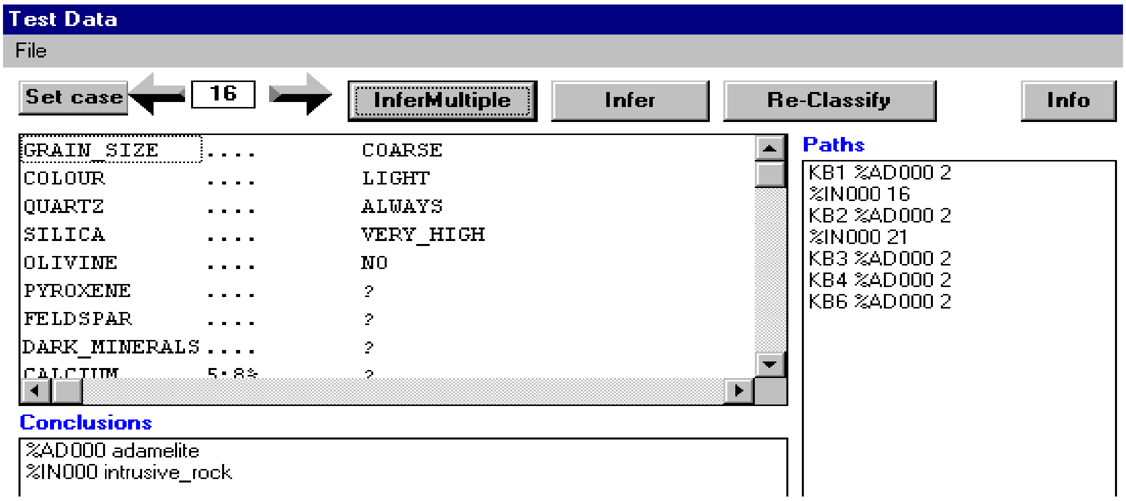

Fig 6.14: Running a Case against Multiple KBS as a Test on the Degree of Agreement

As a further test on the amount of agreement in the T-boxes, all six A-boxes can be loaded (recall only six were complete) and cases run to see the amount of agreement. In Figure 6.14 the case for Adamellite was selected and it was found that C1(KB1), C2(KB2), C3(KB3), C5(KB4) and L3(KB6) all concluded %AD000. The rule for Adamellite in L1 did not fire because it had the condition (PYROXENE=NO) which is not present in the case. If the value for PYROXENE in the case were modified to NO instead of missing then all KBS would correctly classify this rock specimen. Note that when this case was originally run before the modifications none of the KBS gave the conclusion %AD000. The reader is invited to verify this by using the case in Figure 6.14

against the concept lattice in Figure 6.9 treating the attributes that can be reached by an ascending pathway from a conclusion as the rule conditions.

Choosing a conclusion is only one of the many ways that viewpoints can be compared. It can be seen in figure 6.15, that comparisons made by attribute, in this case GRAIN_SIZE, can give us information on differences in terminology and measurement. This diagram can be used to detect possible synonyms (eg COARSE=COARSELY_CRYSTALLINE) and to compare the closeness of viewpoints. It can be seen in concepts 6, 7, and 8 that C5 uses terminology not used by the other views. If COARSELY_CRYSTALLINE were changed to COARSE and NOT_COARSE to MEDIUM then these objects would agree and move to the same concepts as in the other views. It can also be seen that experts C2 and C5 use "?" and UNSURE for the rock %KE000, Kentallenite. These two values are obviously equivalent. Since the other experts do not use the attribute GRAIN_SIZE for classifying Kentallenite, these experts may decide that it is not a useful classifier and drop these attributes (conditions) from the relevant rule. Each conclusion can also be compared to see which have been given the same value. For example, conclusions %AD000 has GRAIN_SIZE = COARSE in all views that use this attribute for that rock, but GRAIN_SIZE = FINE for %DA000-Dacite in C1 and C3, but C2 uses GRAIN_SIZE=MEDIUM for this rock. This may be due to a difference in perception or measurement and the rock may be borderline FINE_MEDIUM.

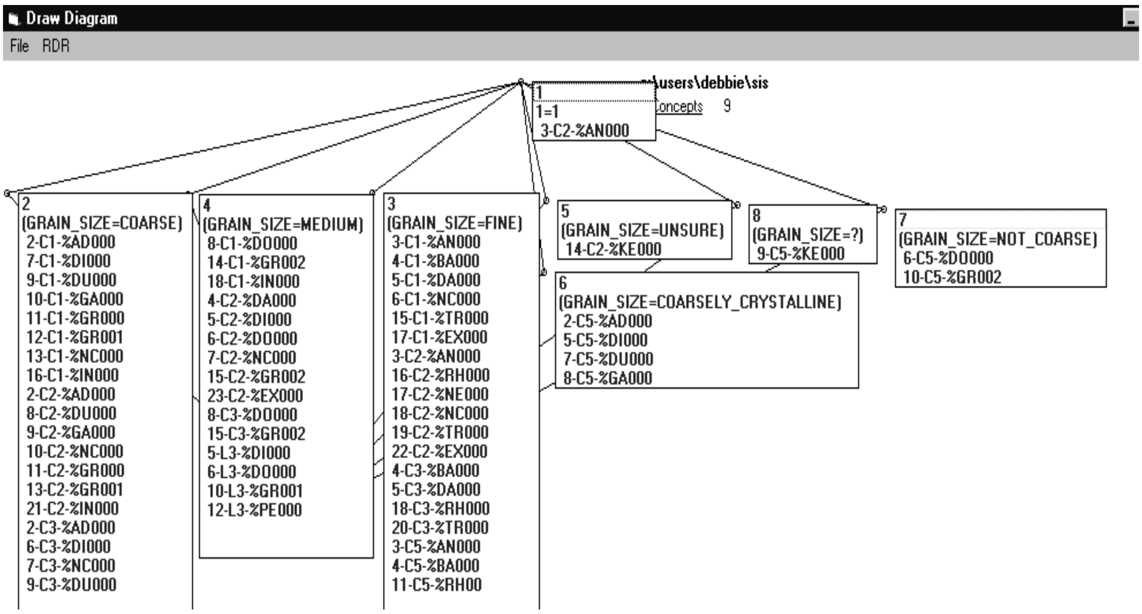

Figure 6.15: Line Diagram for the selection -"attribute = grain_size without related attributes".

6.4.6. Evaluation

Using the measures described in 6.3.6 the degree of conflict at three points was computed: at the beginning; after updating the subsumes table; and at the end. The only viewpoints/concepts in consensus are viewpoints/concepts matched against themselves. There are a number of contrasting concepts shown by the cell entries which equal 1, such as C1-C5 and C1-L1. Concepts shown as a fraction are in conflict where the numerator is the number of attributes not shared and the denominator is the total number of attributes in both concepts. The total degree of conflict for a viewpoint is shown as a decimal for better comparison.

	C1	C2	C3	C4	C5	L1	L3	TOTAL
C1	0	5/7	5/7	6/8	1	1	5/7	4.89
C2	5/7	0	4/6	5/7	1	1	4/6	4.76
C3	5/7	4/6	0	5/7	1	1	4/6	4.76
C4	6/8	5/7	5/7	0	1	1	5/7	4.89
C5	1	1	1	1	0	1	1	6.00
L1	1	1	1	1	1	0	2/6	5.33
L3	5/7	4/6	4/6	5/7	1	2/6	0	4.10
								34.74

Table 6.1: The degree of conflict between each viewpoint before RE.

In Table 6.1 it can be seen that all viewpoints are in conflict with others, with views C1, C2, C3, C4 and L3 having similar degrees of conflict. From this table it can be seen that viewpoint C5 is in complete contrast with all other views, thus it has the highest degree of conflict, followed by L1 that only shares some attributes with L3.

	C1	C2	C3	C4	C5	L1	L3	TOTAL
C1	0	3/7	3/7	2/8	1	1	5/7	3.25
C2	3/7	0	4/6	3/7	3/6	4/6	2/6	3.02
C3	3/7	4/6	0	3/7	2/6	1	4/6	3.52
C4	2/8	3/7	3/7	0	3/7	1	5/7	3.25
C5	1	3/6	2/6	3/7	0	1	4/6	3.35
L1	1	4/6	1	1	1	0	2/6	5.00
L3	5/7	2/6	4/6	5/7	4/61	2/6	0	3.48
								24.88

Table 6.2: The degree of conflict after the synonym table.

In Table 6.2, it can be seen that the total degree of conflict for each viewpoint is lower after similar terms have been reconciled using the subsumes table, even though it has not reduced the conflict in all cells. The total degree of conflict for all viewpoints has reduced from 34.74 before the resolution strategies were applied to 24.88 after the first strategy of reconciling terms was used. It is interesting to note that all views except L1 now have similar, though lower than before, degrees of conflict. This shows that much of the conflict originally in C5 was due to differences in terminology. This is not the case with L1 which is the most in conflict after this round of negotiations (Table 6.3).

	C1	C2	C3	C4	C5	L1	L3	TOTAL
C1	0	1/7	2/6	1/7	3/7	4/8	3/7	1.98
C2	1/7	0	3/5	2/6	4/6	3/7	2/6	2.50
C3	2/6	3/5	0	1/5	1/5	4/6	3/5	2.60
C4	1/7	2/6	1/5	0	2/6	5/7	4/6	2.39
C5	3/7	4/6	1/5	2/6	0	5/7	4/6	3.01
L1	4/8	3/7	4/6	5/7	5/7	0	1/7	3.17
L3	3/7	2/6	3/5	4/6	4/6	1/7	0	2.84
								18.49

Table 6.3: The degree of conflict between each viewpoint after RE.

Table 6.3 shows the degree of conflict remaining after the resolution strategies in Figure 6.4 and the various resolution Tag Codes were applied to produce the final concept lattice in Figure 6.13. As noted before not all conflict has been removed but it has reduced by 53% from 34.74 to 18.49.

6.5 *Related Work*

In the framework, cases play a critical role and it has been assumed that cases are available. In the use of the SISYPHUS III data, a source of cases has been available but this may not always be so, particularly when dealing with software specification requirements. One viable option for the purposes of RE is *use cases* since they satisfy the need for sets of attributes and outcomes and are "primarily an approach to discovering requirements from a user-centred viewpoint" (Rumbaugh 1994, p.23). They could be used in conjunction with RDR or as direct input into the decision table and maintenance would consist of modification to the use cases. So that parts of the system are not forgotten, the actors, which are external agents that require services from the system, are enumerated followed by the use cases. Specific values, not generalisations should be plugged into the cases so that thinking is grounded in precise examples. Rumbaugh suggests first building a system which contains the domain model and then the application model using use cases. The domain model and application model can be equated to the T-box and A-box, respectively. The process described in section 6.3.3 advocates the use of cases to build the A-box, the rules, from which the T-box, the concept hierarchy, is derived. The problem with use cases is that it may be hard to extract the relevant details to go into the case. There may also be too much information extracted to be included in a formal context which would make comparisons using the line diagrams difficult (Shaw, personal communication). Another possibility is the use of the repertory grid technique to acquire specifications from different viewpoints which could be used as cases for RDR. Alternatively repertory grids could be used directly as input into Phase Two and easily converted to crosstables. The process described in Phases Three to Five can then proceed as outlined in this chapter. Phase One would consist of development and maintenance of individual repertory grids.

Prior work in this area has been presented as verification or maintenance research (Compton et al 1992, Richards and Compton 1997a). Richards and Menzies (1997) claim that an RE technique could be viewed as a technique for knowledge maintenance (KM) and V&V. Menzies (1997b) argues that knowledge management techniques amount to a small number of activities processing less than a dozen types of knowledge. At different points of the software lifecycle, the emphasis is on certain activities processing some of the knowledge types:

- At later stages of the life cycle, the process of fixing inconsistencies is called "maintenance" or "validation". The essential difference between requirements engineering and maintenance or validation is that more behavioural knowledge is available (e.g. libraries of previously successful runs of the system). That is, $|X|$ is larger for (V&V and KM) than RE and $|X_{RE}| << |X_{KM}|$. With behavioural knowledge, it is possible to assess proposed KB revisions via running again all old cases.
- At the initial stages of the life cycle, the "inconsistency detection" and "fix" activities process the knowledge types of domain assertions and problem solving methods. Typically, at this early stage, there is little "behavioural knowledge" describing known or desired behaviour of the system. Requirements engineering, then, is the process of fixing inconsistencies in a system which has yet to execute. Without behavioural knowledge, it is necessary to rely on inspection tools that allow the expert to reflect over their knowledge. This chapter has assumed limited behavioural knowledge and has hence focused on inspection tools. If actual cases are available, which contain the behavioural knowledge, then we are able to avoid much of the conflict and develop a specification from these cases possibly using the RDR KA technique.

The only difference between MCRDR/FCA for requirements engineering and MCRDR/FCA for maintenance is that one more revision assessment operator can be used in the maintenance case (regression analysis over behavioural knowledge). Richards and Menzies (1998) speculate that this work is a possible first step towards a succinct toolkit to support knowledge engineering and software engineering activities right across the life cycle.

6.6. Future RE Work

A preliminary exploration of the RE framework has been defined and parts of the toolkit in operation have been demonstrated. Clearly, the next stage is full implementation of this approach. Such an implementation would have to handle a variety of interface details (e.g. appropriate displays of large concept lattices via such techniques as scrolling or zooming) and could be an extension of the existing MCDRD/FCA tool.

The approach aims to minimise the conceptual distance between A-boxes from different experts. The decision to accept that some conflict will always remain means that we need an evaluation and termination strategy that lets us test whether the framework is reducing the conflict and when enough conflict has been resolved to allow the specification to be used. As demonstrated in Sections 6.3.6 and 6.4.6 it is desirable to track over-time whether with each subsequent cycle there is a move towards consensus and these statistics should be computed when the resolution strategies for the current cycle have been applied. However, the statistics used were easily computed because it was known which concepts to use for comparison. As pointed out in Section 6.3.6, it we want to compute an overall degree of conflict within each KBS it may be harder to determine what concepts should be compared and how these scores should be used. A graph-theoretic distance measure between T-box lattices is proposed.

6.7. Further Discussion

This chapter has argued for a novel view of RE. Standard RE is an early-software lifecycle issue. The viewpoint resolution technique discussed here can be performed whenever there are some A-box (rules) and cases. Initially, cases will be hypothetical and the rules sets small (snippets of known business processes). Finally, cases will be "live" data and rule sets will be large. Large KBS can be handled in this framework by narrowing the focus of attention and reducing the context. In either case the technique can be applied. That is, the "RE Tool" can be applied right throughout the system development life cycle.

The RE framework described is a small extension to the KE approach described in Chapter Five. RDR and FCA were used as subroutines within the RE system. For the example given, the only additional code needed for RE was about 100 extra lines of Visual Basic code added to a system that uses 2,500 - 3,000 lines of C code and 1,500 - 2,000 lines of Visual Basic code for the interface. I envisage that full implementation would require no more than a total of 500 extra lines. Thus, the RE extension to a KA tool will be a mere 10% extension (about 500 lines on top of 5000 existing lines).

FCA was used to build explanatory T-boxes from performance A-boxes. This approach is applicable to any representation which can be mapped into a decision table. However, during the discussion on resolution strategies (Figure 6.4), it was noted that certain representations offered advantages. For example, when adding an attribute in a standard propositional rule base, the effects of this addition had to be checked all over the KBS. Such a check comes for free in RDR.

6.8 Chapter Summary

In this chapter it has been shown that by using the MCRDR/FCA tool it is possible to develop, compare and combine viewpoints from multiple stakeholders or sources. The RE framework provided offers more than just a comparison tool but included a number of operators which were extensions to MCRDR/FCA to handle the resolution of conflict. This has involved a change in some of the modification strategies offered by MCRDR as it describes a technique for maintaining the assertional KBS that is driven by the changes needed to the terminological KBS and also supports the removal of terminological knowledge, which is not possible in the assertional KBS.

To instantiate the RE framework the SISYPHUS III data has been used. It was not the goal of this chapter to evaluate the accuracy of the KBS developed or their usefulness for classification or instruction which was the aim of the SISYPHUS III project. However, it has been shown that the MCRDR KBSs, which include the individual as well as the combined viewpoints, are executable and it is claimed that the concept lattices could act as a tutoring tool in helping the user learn about the domain. The goal of this chapter has been to address an issue not stated explicitly in the problem statement for SISYPHUS III and that is how to manage multiple and conflicting sources of expertise. Measures have been provided that show the resolution strategies can reduce the degree of conflict between multiple stakeholders. In tackling this problem, this approach has also addressed a drawback of standard RDR. RDR systems have been shown to be useful for single expert knowledge acquisition. In such a situation, RDR offers a good performance module, but a poor explanation module. However, in the case of multiple experts, an explanation system is required since experts must trade off their competing views. FCA allows us to build an explanatory T-Box from an A-box initialised by RDR.

This chapter has opened up a new range of activities that may be supported by the MCRDR/FCA tool beyond those considered necessary for KBS based on a single source of expertise. This does not mean that only one expert was involved but that only one viewpoint, which may have used a combination of expert sources, was captured in the KBS. In the next chapter we return to consider the types of activities an individual will be interested in performing. The various requirements and features of the different activities are explained together with a description of how the MCRDR/FCA system is able to support the range of activities.