Extending the SISYPHUS III Experiment from a Knowledge Engineering Task to a Requirements Engineering Task

Debbie Richards, Tim Menzies

AI Department, School of Computer Science and Engineering,
The University of New South Wales, Kensington
{debbier, timm}@cse.unsw.edu.au

Abstract: The problem statement and scope of SISYPHUS III does not draw attention to one of the major problems faced in knowledge engineering (KE), which is building systems based on multiple sources of expertise. In this circumstance, the KE task becomes a requirements engineering (RE) task. A problem with many RE approaches is that the cost of use is prohibitive, and therefore such approaches are rarely applied. We present an RE strategy designed to handle conflicting perspectives that is an extension to current KE techniques. We implement this approach in the context of formal concept analysis (FCA) and ripple-down-rules (RDR) and describe an instantiation using the SISYPHUS III data. Our evaluation technique shows that the resolution operators have reduced the degree of conflict between viewpoints.

1. Introduction

The SISYPHUS III experiment offers an excellent example of the similarity between the needs of knowledge engineering (KE) 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 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 (KE)? 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.

In this paper, we present a simple RE strategy that is an extension to current KE techniques. Given assertions in rulebases (the A-boxes) from different stakeholders, we generate and critique a concept hierarchy (the T-box). Conflicts recognised in the T-box can be used to drive negotiation strategies amongst the different stakeholders. A general framework for this approach is described together with an implementation using formal concept analysis (FCA) (Wille, 1982) and ripple-down rules (RDR) (Compton and Jansen, 1990).

This paper is organised as follows. Section 2 introduces RE, Section 3 describes our RE framework and our implementation in found in section 4. A case study is presented in Section 5 using seven knowledge bases built from the data for the SISYPHUS III (Shadbolt 1996) project. Our evaluation technique is described in Section 4.6 and is used in Section 5.6 to show how the resolution operators have halved the initial degree of coflict. Related work, future work and the conclusion are presented in Sections 6, 7 and 8, respectively.

2. 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).

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.

Our 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 we call a stakeholder. When managing different viewpoints, conflict between different stakeholders must be handled.

3. Our Framework

Our 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)[1], which we call the performance system, and a terminological knowledge base (T-box), which we call 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.
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.

4. An Implementation

Our general framework in Section 3 has not committed to any particular implementation choices. We continue now looking at this framework but within the context of our instantiation. As a result a number of restrictions on the generality of our implementation are imposed.

[1] Assertional KBS are made up of executable assertions (such as rules) that assert the relationships between terms. Terminological KB consist of terms structured into inheritance networks (Brachman 1979). Their main building blocks are concepts and roles and they reason by determination of subsumption between concepts (Nebel 1991).

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

We take a KBS approach to requirements acquisition. In this paper, we focus on the case 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. We place a further restriction that the A-box must be convertible into a decision table. It has been shown (Colomb 1993) that any decision tree or propositional KBS may be converted into a decision table. Other work (Richards and Compton 1997b) has shown how ripple-down rule (RDR) systems can easily be converted into decision tables. Conversion to a decision table is also suitable for production rule-based systems and has been applied to a number of CLIPS KBS. For our purposes we distinguish between RDR systems and propositional rulebases. We refer to the latter 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). This study proposes the use of multiple classification RDR (MCRDR) (Kang, Compton and Preston 1995) for Knowledge Acquisition (KA) and knowledge representation (KR). We adopt RDR because maintenance in RDR is a simple task that can be performed by the user and does not suffer from the side-effect problem which occurs in typical rule-based systems (Soloway, Bachant and Jensen 1987). KA using RDR has been designed to be performed by the user and as Herlea (1996) points out user involvement is a critical issue in making RE work. An additional benefit of RDR, as mentioned above, is that the rule pathways map directly into a decision table and do not need intermediate conclusions to be mapped to primitive conditions as many rule bases require.

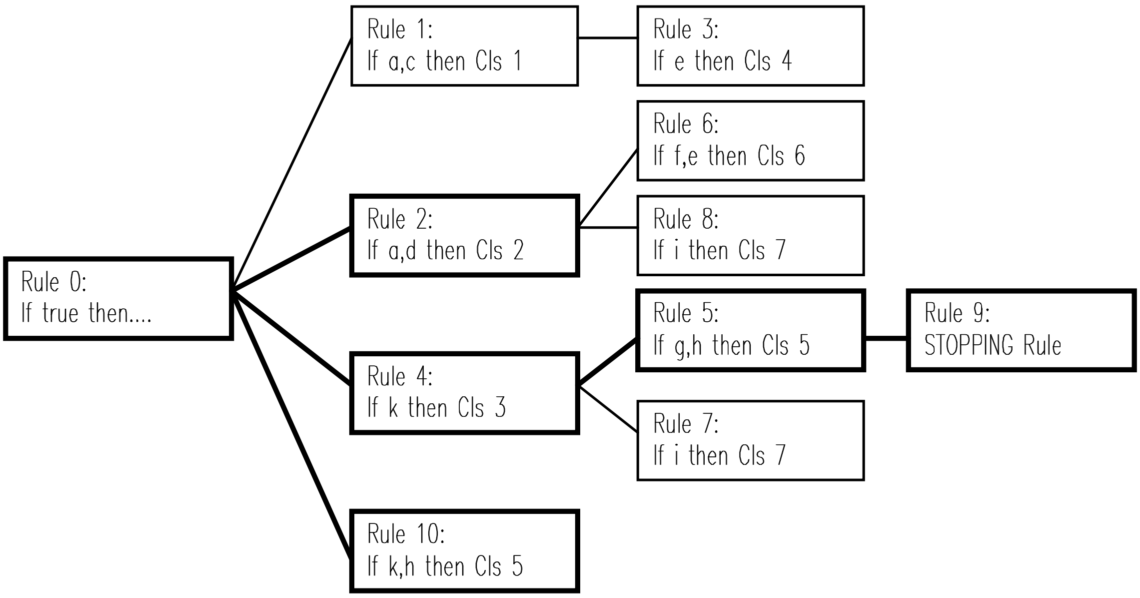

Figure 1. An MCRDR KBS.
The highlighted boxes represent rules that are satisfied for the case {a,d,g,h,k}. We can see that there are two independent conclusions for this case, Class 2 (Rule 2) and Class 5 (Rule 10). Rule 5 had been the cause of a conflict between viewpoints. To resolve this conflict it was decided that attribute g should be dropped. As described in Figure 2, 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.

The RDR approach to KA is to run a case and show the user the system-assigned conclusion/s. If the user agrees with the conclusions given then they process the next case. If they do not agree with a conclusion they take the option to reclassify the case. Reclassification involves specifying the correct conclusion and picking some features in the case that justify the new conclusion. These features form the conditions of the new rule. The new rule is added as an exception to the rule that gave the misclassification. The case that prompts a rule to be added is stored in association with the new rule. When a new rule is added, the rule must distinguish between the present case and all the stored cases that can reach that rule. To do this, the expert is required firstly to construct a rule which distinguishes between the new case and one of the stored cases. If other stored cases satisfy the rule, further conditions are required to be added to exclude a further case and so on until no stored cases satisfy the rule. Stopping rules, which prevent an incorrect conclusion by providing a null conclusion are added in the same way. Surprisingly the expert provides a sufficiently precise rule after two or three cases have been seen (Kang, Compton and Preston 1995). Multiple Classification (MCRDR) is defined as the triple <rule,C,S>, where C are the children/exception rules and S are the siblings. All siblings at the first level are evaluated and if true the list of children are evaluated until all children from true parents have been exhausted. The last true rule on each pathway forms the conclusion for the case. Figure 1 shows an example of an MCRDR.

The greatest success for RDR has been the Pathology Expert Interpretative Reporting System (PEIRS) (Edwards et al 1993). PEIRS went into routine use with approximately 200 rules and grew in a four year period (1990-1994) to over 2000 rules. The system was maintained by the expert and the 2000 rules represent a development time of approximately 100 hours.

As noted in our 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 RDR approach as described above, or other case-based technique, the cases are also used for initial KA and for modification of other views. The way this works is that when a concept is found to be in conflict, as one of our modification strategies outlined below (see Figure 2), we pass the case or cases associated with that concept to the other stakeholder for KA. This should either resolve the conflict or at least ensure that both parties have given their views given the same set of criteria. In Related Work we suggest how we can obtain cases.

4.2 Phase Two: Requirements integration

Requirements integration is the process of ensuring that all viewpoints are in formats that can be compared. Adopting our 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, we convert all KRs into one format so that we only need 2N mapping schemes. As mentioned in Phase One, in our current implementation we can use any representation that maps into a decision table so that a common approach to subsequent phases can be taken. It will become apparent in the next section why we have the decision table format restriction.

4.3 Phase Three: Concept Generation

In our 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 we have 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 our five phases. These structures are:

- One circumvent table for each A-box. This table identifies which rules to skip in future RE sessions.

- One synonym 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 4.5.

We have taken the approach of starting with a performance system (A-box) and using that to derive an explanation system (T-box) because we see that defining and building models is inherently difficult and flawed. 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. We see the difficulty in capturing mental models as a contributing factor to the bottleneck associated with KA. We now show how we use FCA to make the leap from A-box to T-box.

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 *extent* of the concept while the set of attributes forms the *intent* of the concept. Knowledge is seen as applying in a context and can be formally defined as a crosstable. We interpret the decision table in Figure 5 as a formal context where 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.

In Figure 5 we have the formal context "First four MCRDR rules for Card Sort 5" with the set of objects = {1-%NC000, 2-%AD000, 3-%AN000, 4-%BA000}[2] and set of attributes = {1=1[3], grain_size = coarsely_crystalline, silica = very_high, colour = light, grain_size = fine, silica = intermediate, silica = basic }. The crosses show where a relation between the object and attribute exists, thus the set of relations = {(1-%NC000, 1 = 1), (2-%AD000, grain_size = coarsely_crystalline) (2-%AD000, silica=very_high),…,(4-%BA000, silica=basic)}. Each row in the crosstable represents a concept. By finding the intersections of sets of attributes and the set of objects that share those attributes we are able to form new higher level abstractions. The set of concepts can be ordered using the subsumption relation ≤ on the set of all concepts which can be used to form a complete lattice. For a more detailed and formal treatment of our approach see Richards and Compton (1997b).

In Figure 6 the concepts are shown as small circles and the sub/superconcept relations as lines. Each concept has various attributes and objects associated with it. The full labelling for each concept has been reduced for clarity by removing all attributes from a concept that can be reached by ascending paths and removing all objects from a concept that can be reached by descending paths. Thus, the concept lattice provides "hierarchical conceptual clustering of the objects (via the extents) …. and a representation of all implications between the attributes (via its intents)" (Wille 1992, 497).

4.4 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). In this paper, we use the four quadrant model of comparison between experts developed by Gaines and Shaw (1989). This model classifies 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.

---

[2]The object name is made up of the rule number and the conclusion code

[3] 1=1 is the condition for our default conclusion %NC000, as shown in the MCRDR in Figure 1, which stands for No Conclusion. The most common conclusion for a domain can also be used as the default conclusion to reduce the number of rules needed to cover the domain.

Contrast is where there is no similarity between concepts or the terminology used.

In this paper we generally take a broader view of conflict to encompass inconsistencies that include the states of contrast, correspondence and conflict. Gaines and Shaw's model, however, does offer us greater precision in describing the nature of the conflict which is important in deciding how it can be handled. We more formally define the states of consensus and contrast according 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 5.4.

Consensus $\{V_1 .P_i .B_j\} = \{V_2 .P_k . B_L\}$ where $B_j = B_L$ and (1)
$\{V_1 .P_i .O_j\} = \{V_2 .P_k . O_L\}$ where $O_j = O_L$

Contrast $\{V_1 .P_i .B_j \cap V_2 .P_k . B_L\} = \varnothing$ and (2)
$\{V_1 .P_i .O_j \cap V_2 .P_k . O_L\} = \varnothing$

As shown in (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. Discovering the consensus between conceptual models is important for establishing common grounds from which differences can be viewed. There is no point considering differences if there is no similarity in the first place. Contrast 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 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 our approach it would be up to the stakeholder 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 our crosstable could be mapped into, can be used to identify where terminology is the cause of inconsistency. Having detected various conflicts we now consider how to resolve them.

4. 5 Phase Five: Conflict Negotiation

Before we can decide how to fix a detected inconsistency we need 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 (1989) 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 we extend our approach 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 covers the actions we have 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 being lost or altered. A simple and more appropriate solution is to use a synonym table which maps terms from individual views into a shared terminology which are then used for comparison.

Another way in which conflict may be resolved is through the addition or deletion of attributes or objects. The conflict may be that the set of attributes or objects are 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 our 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 2(a) and (b) we provide a summary of the strategies applicable to standard rules and MCRDR. Note that:

- In each strategy for handling attributes or objects from standard rules we have added the requirement that some sort of checking should be done to ensure that there are not unknown side effects elsewhere in the rulebase. With MCRDR we get this checking for free because we know that 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 2(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 2(a) The resolution strategy for handling attribute and objects.

	Adding Real Cases	Deleting Real Cases	Adding Hypo-thetical 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 2(b) The resolution strategy for handling cases.

We plan to strengthen our negotiation strategies 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. If we consider the concept lattice structure where objects belonging to a concept are reached by descending paths and attributes are reached by ascending paths then we can say that if an attribute at the bottom of a pathway or an object at the top of a pathway is to be removed then we know that no other concepts will be affected and this 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.

We also accept that living with inconsistency will be necessary and use 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 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, we mark the concept 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 we only offer the strategy of circumvention to support avoidance of certain unstable parts of the requirements by tagging a rule as "circumvented" in the individual A-boxes. When the new T-boxes are generated these rules will not be included.

The resolution strategies shown in Figures 2(a) and (b) are also applicable to the strategy of amelioration. However, the result is not consensus but a reduction in the extent of the conflict. Amelioration results in bringing concepts closer together. If we think in terms of the concept lattice we would be shortening the distance between the two concepts.

4. 6 Evaluation

To determine that our RE strategy is resolving conflict we need to employ some measures of 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 we can check that the degree of conflict after the RE process is less than at the start. We assign of score of 0 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 conflict we take the number of attributes (conditions) that they have but do not share divided by the total number of attributes. This assumes that the two concepts 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. For concepts in a state of contrast (no partial or complete match in the other viewpoint) we assign a score of 1, which is the same result as if we used the conflict measure since the number of attributes not shared divided by the number of attributes is equal to one. Concepts in a state of correspondence are treated the same as concepts in conflict since we are ignoring the reason for the differences and are just interested in the size of the difference. Once terminology differences are reconciled such concepts will move into one of the other states and be handled accordingly.

We have used these measures in the section 5.6 to determine how well our RE strategy is working. 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 we do not know which concept in the other view represents a similar concept. We either need to get the experts to label their concepts so that, for example, two concepts labelled "cat" would be compared or we can 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 we did not need to resolve this problem and further work on distance measures is part of our future work.

5. Requirements Engineering using SISYPHUS

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 our 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.

We extend the problem description 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. Thus, we not only want to develop a system that classifies a set of igneous rocks correctly and offers an explanation of the conclusion but we want to use data from multiple and conflicting sources of expertise to build this system. These multiple sources of expertise impose a further requirement on the project not indicated in the problem statement above which is to develop a strategy for handling any differences. Thus, the SISYPHUS III data provides an ideal source of different models covering the same domain in which to test out our RE framework.

The RDR submission produced a number of KBS from a combination of these sources. The systems were built by a geological and KE novice in the same manner that is described in section 5.1. 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 our novice KE thought was right. It seemed reasonable that the approach outlined in this paper for reconciling conflict in multiple viewpoints would 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.

In the next subsections we work through the five phases already presented showing how the proposed framework can be implemented. Note that the process is implemented 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 we are dealing with, in particular representing 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. One of the benefits is that as we begin to resolve conflicts, for example reconcile differences in terminology through the synonym table or tag concepts that should be ignored, there will be less errors or differences as we progress.

5.1 Phase One - The Data and Knowledge Acquisition

To compare the models represented in each of the data sources mentioned above, we had our KE novice develop individual KBS using MCRDR for each of the card sorts and laddered grids. For the purposes of this case study we did not use the other material even though supporting or extra information had originally been gleaned from them and added to the final KBS. We did this because we wanted the models to represent only the data directly specified as related to them and to avoid inclusion of the KE's own interpretation. 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 we 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 personal construct psychology, where subjects prefer to use their own constructs over ones supplied to them (Shaw 1980).

```
#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 3: Extract from Card Sort 5 Case File.

The data in the card sorts were used directly to develop cases. In Figure 3 we can see the first three records in the Card5 case file. For the restricted study performed in this paper, a case file was developed only for Card Sorts 1, 2, 3 and 5 and Laddered Grids 1 and 3[4], 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 that 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. The MCRDR rule file developed from the Card5 case is shown in Figure 4. For the rest of this example, each KBS is treated as a different expert viewpoint.

[4] To make our examples using the conclusion %AD000- Adamellite in Figures 6, 8 and 9 more complete, we have developed a partial KBS for Card Sort 4 to cover that rock.

```
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 4: 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 after the colon.

5.2 Phase Two: Requirements Integration

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

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

5.3 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[5] 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 all. However, handling such a large context has its problems. As previously mentioned:"the diagrams for activities of any reasonable complexity become very difficult to visualise and understand" (Gaines and Shaw 1993, p. 59). To address this problem the MCRDR/FCA system provides a selection screen with 13 combinable options that allow the user to narrow their focus of attention to selected part/s of interest in the knowledge base. This approach requires a methodical approach to comparing the various parts of each KBS and may miss some conflict that is not being picked up because the relevant selection criteria was missed. However, if our major goal is correct classification of a particular rock then it seems reasonable to explore each conclusion individually from the combined viewpoints to see what conflict exists. Figure 6 shows the concept lattice for the conclusion %AD000- Adamellite.

---

[5] Our 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 10 the first conclusion would read 2-C5-%AD000 which stood for rule 2, source Card Sort 5 and conclusion code %AD000.

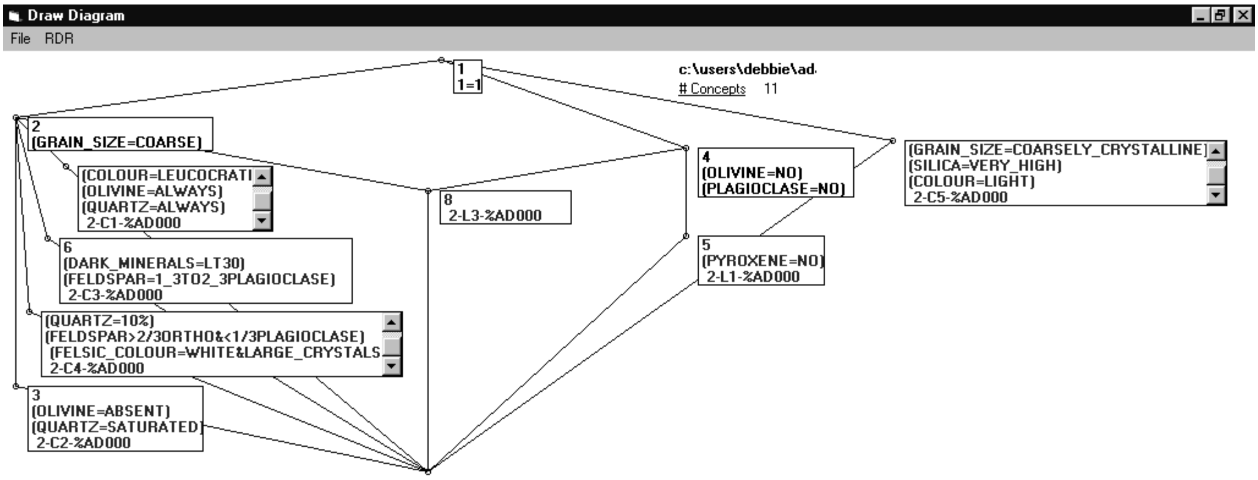

Figure 6: The Concept Lattice for the Conclusion %AD000- Adamellite based on seven KBS.

5.4 Concept Comparison and Conflict Detection

The concept lattice in figure 6 may be analysed and the nature of the differences between concepts determined. The C5 viewpoint for Adamellite is in contrast since none of its attribute are shared with any of the other viewpoints. However, some of these difference 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 DARK_MINERAL=LT30 also 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 synonym table to map to a common term as shown in Figure 7.

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&<1/3PLAGIOCLASE for C4. These various conflicts need to be resolved which takes us to our next stage.

5.5 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 7: The Synonym Table

Reconciliation of conflicts in terminology is a good place to start in the negotiation process. Those terms identified to be equivalent are put in the synonym table, see Figure 7. We update this table and go through phases 1-5 again. In Figure 8 we can see the effect of the synonym table on the original conflict in Figure 6.

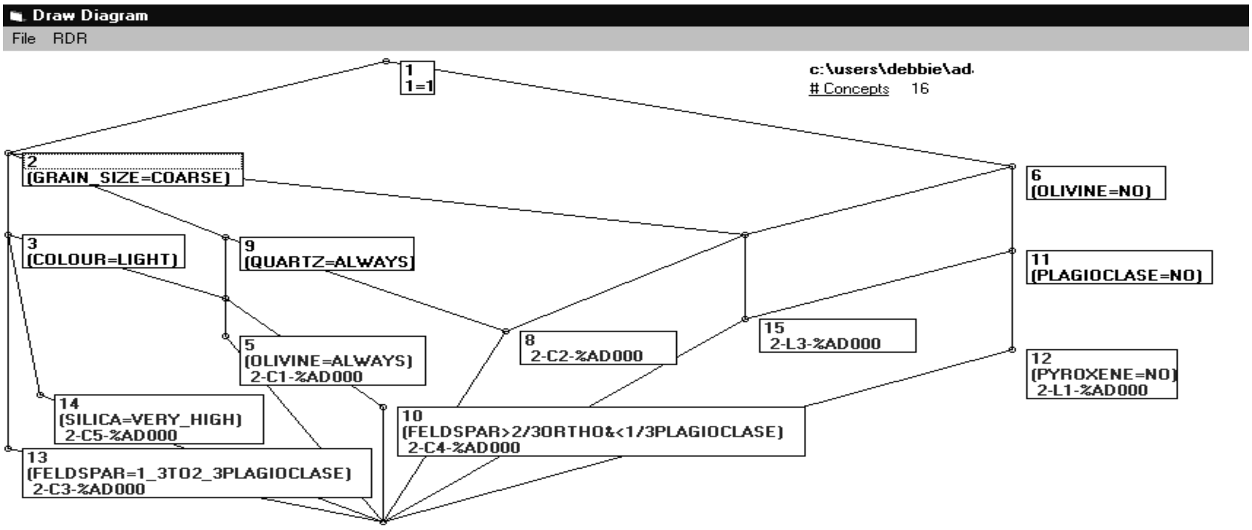

Figure 8: The updated Line Diagram after the Synonym Table in Figure 7 has been included.

In addition to terminological differences we have some conceptual differences. To simulate how some of these differences can be resolved the following scenario is offered:

- Since sources C2, L1 and L2 agree that OLIVINE=NO, the expert in C1 decides that he has made an error and changes OLIVINE=ALWAYS to OLIVINE=NO. As described in Figure 2(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.
- The L1 expert agrees that GRAINSIZE=COARSE should be included so he amends his KBS by adding an exception rule to Rule 2, as outlined in Figure 2(a), where the conclusion is given as %AD000 and the condition as GRAINSIZE=COARSE.
- A feature of our approach is the ability to offer counterexamples that can be used in negotiations. In Figure 10 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 4.3 and 4.5. 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 10.
- To circumvent an object (rule) the related rule is flagged before a context is generated. In these examples we circumvent all rules that conclude %NC000 - the no conclusion rule.
- To ignore or delay a concept we use the Ignore Tag Code "I" or Delay Tag Code "D", respectively, as shown in Figure 9. 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". Two delayed concepts are shown in Figure 10.
- 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 only 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 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 10 that these attributes are no longer shown. If desired, these concepts can be reinstated and shown.

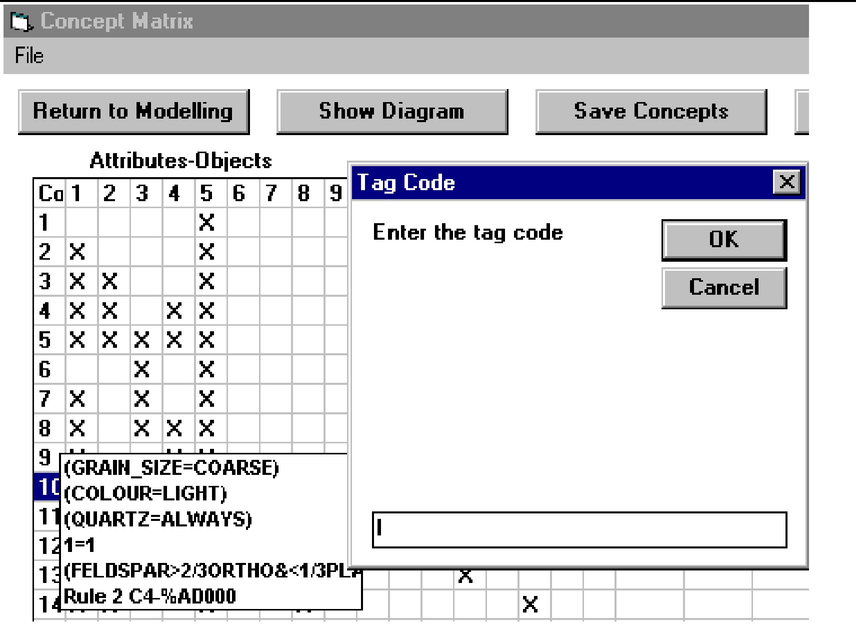

Figure 9: Users can flag a concept to be "ignored" on the CMatrix Screen by clicking on the Concept No and entering the Tag Code into the pop-up InputBox.

All of the changes mentioned above are reflected in our final diagram in Figure 10. Note that even though the number of concepts has only reduced by 1 the concepts are much less complex. In figure 6 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 that previously only appeared in one viewpoint. As shown visually in Figure 10, the viewpoints in Card Sorts 1,2,3, and 5 are more similar to each other than the viewpoints in Laddered Grids 1 and 3 which are similar to each other.

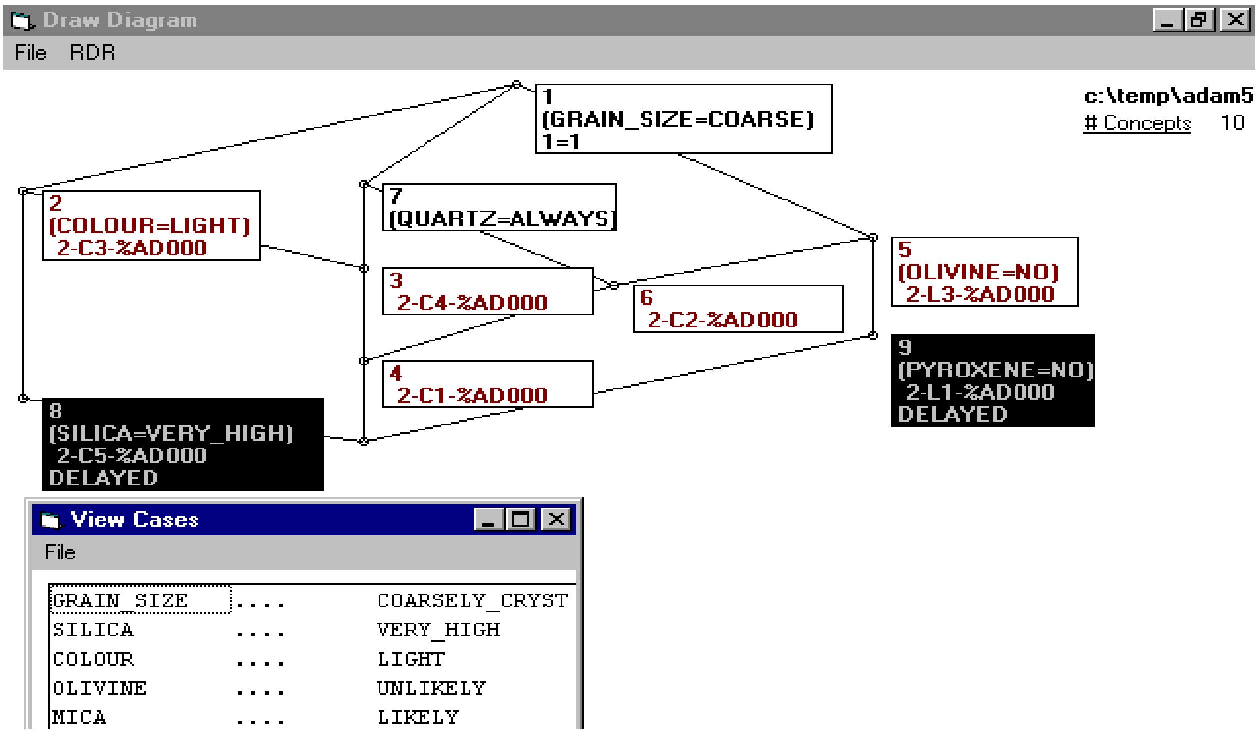

Figure 10: The final Line Diagram screen from this round of negotiations.
Some attributes have been dropped or added to views, concepts have been tagged to be circumvented (not shown) or delayed (shown). The case for concept 8 is shown to assist with negotiation. There is considerably less conflict now than in Figure 6.

For those finding the diagram unfamiliar 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

As a further test on the amount of agreement in our T-boxes we can load all six A-boxes, recall only six were complete, and run cases to see the amount of agreement. In Figure 11 we tested our case for Adamellite and found that C1(KB1), C2(KB2), C3(KB3), C5(KB4) and L3(KB6) all conclude %AD000 . The rule for Adamellite in L1 does not fire because it has the condition (PYROXENE=NO) which is not present in the case. It 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 11 against the concept lattice in Figure 6 treating the attributes that can be reached by an ascending pathway from a conclusion as the rule conditions.

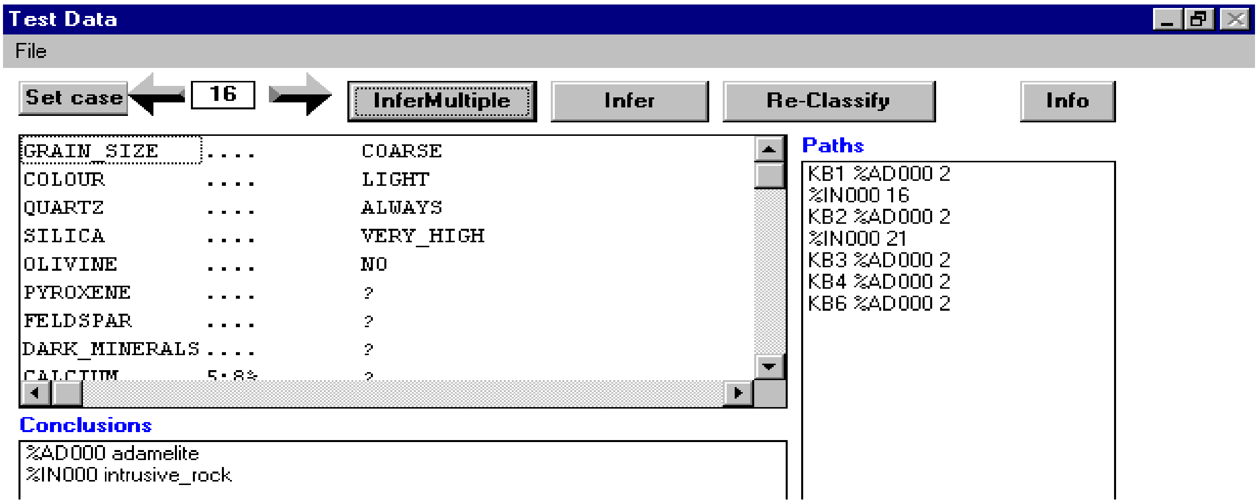

Fig 11: Running a Case against Multiple KBS as a Test of the Degree of Agreement

Choosing a conclusion is only one of the many ways that viewpoints can be compared. We see in figure 12, that comparisons made by attribute, in this case GRAIN_SIZE can give us information on differences in terminology and measurement. We can use this diagram to detect possible synonyms (eg COARSE=COARSELY_CRYSTALLINE) and to compare the closeness of viewpoints. In 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. We can also see that experts C2 and C5 use "?" and UNSURE for the rock %KE000, Kentallenite. These two values are obviously equivalent. The other experts do not use this attribute for classifying this rock and the other two may decide that it is not a useful classifier and these attributes (conditions) may be dropped from the relevant rule. We can also compare each conclusion 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 C1and 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.

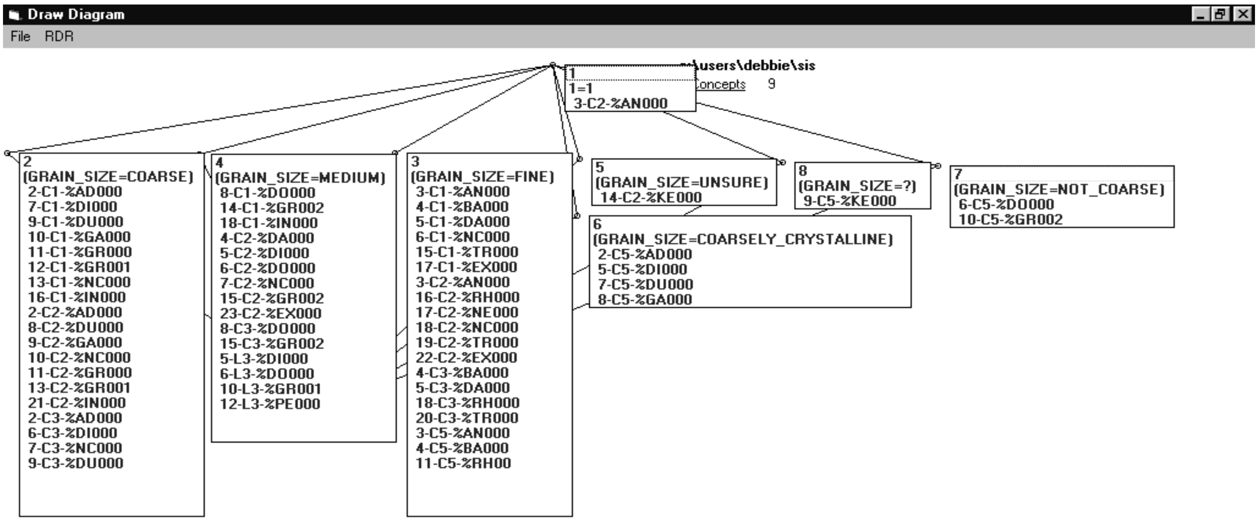

Figure 12 - Line Diagram in MCRDR/FCA for the selection - "attribute = grain_size without related attributes".

5.6. Evaluation

Using the measures described in 4.6 we computed the degree of conflict at three points, at the beginning, after updating the synonym table and at the end. The only concepts in consensus are 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 fractions 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 1: The degree of conflict between each viewpoint for the %AD000-Adamellite conclusion before RE.

In Table 1 we can see that all viewpoints are in conflict with others, with views C1, C2, C3, C4 and L3 having similar degrees of conflict. From this table we can see 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 2: The degree of conflict after the synonym table.

In Table 2, we can see that the total degree of conflict for each viewpoint is lower after similar terms have been reconciled using the synonym 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 we began our resolution strategies to 24.88 after we applied our first strategy of reconciling terms. 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, which we have already discovered in our previous discussions. Very little of the conflict in L1 appears to be terminology related and as is shown in Table 3 after we have completed this round of negotiations, L1 remains the viewpoint most in conflict with the others. After we have applied the resolution strategies in Figure 2 and used our various resolution Tag Codes to produce the final concept lattice in Figure 10, the degree of conflict remaining is shown in Table 3. As noted before we have not removed all conflict but it has reduced by 53% from 34.74 to 18.49.

	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 3: The degree of conflict between each viewpoint for the %AD000-Adamellite conclusion after RE.

6. Related Work

Starting with a performance system and deriving an explanation system, is in complete contrast to mainstream KA research where the focus is on building problem-solving (Chandrasekaran and Johnson 1993, McDermott 1988, Puerta et al 1992, Schreiber, Weilinga and Breuker 1993 and Steels 1993) and/or ontological (Guha and Lenat 1990, Patil et al 1991, Pirlein and Studer 1994) models first and using these to develop a performance system. As explained earlier, conceptual models are difficult to capture and unreliable and we prefer to capture simpler models based on the performance knowledge that can be demonstrated and observed. RDR has given us a reliable method for capturing and maintaining performance knowledge and FCA is the mechanism that lets us derive the explanation system.

In our framework, cases play a critical role and we have assumed that cases are available. In our use of the SISYPHUS III data we have had a source of cases but this may not be always be so, particularly when dealing with software specification requirements. One viable option for the purposes of RE is *use cases* since they satisfy our 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 we don’t forget parts of the system we first enumerate the actors, which are external agents that require services from the system, and then 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. We can equate the domain model and application model to our T-box and A-box, respectively. What we are advocating is to use the cases to build the A-box, the rules, from which we derive the T-box, the concept hierarchy.

Prior work in this area has been presented as verification or maintenance research (Compton et al 1992, Richards and Compton 1997a). We here claim that an RE technique could be viewed as a technique for knowledge maintenance (KM) , validation and verification (V&V). Menzies (1997) 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 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, we must rely on inspection tools that allow the expert to reflect over their knowledge. This article has assumed limited behavioural knowledge and has hence focused on inspection tools.

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

7. Future Work

We have defined here a preliminary exploration of our framework and shown parts of the toolkit in operation. 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). Such an implementation could be an extension of Richard's existing MCDRD/FCA tool (as in Figures 6 and 8-12).

Our approach aims to minimise the conceptual distance between A-boxes from different experts. Our decision to accept that some conflict will always remain means that we need an evaluation and termination strategy that lets us test whether our framework is reducing the conflict and when enough conflict has been resolved to allow the specification to be used. As demonstrated in Sections 4.6 and 5.6 we want to track over-time whether which each subsequent cycle we are moving towards consensus and these statistics should be computed when the resolution strategies for the current cycle have been applied. However, the statistics we used were easily computed because we knew which concepts to use for comparison. As pointed out in Section 4.6, 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. Currently, we are working on a graph-theoretic distance measure between T-box lattices.

8. Conclusion

We have argued here for a novel view of RE. Standard RE is an early-software lifecycle issue. The viewpoint resolution technique discussed here can be performed whenever we have 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. In either case we can apply our technique. That is, our "RE Tool" can be applied right throughout the system development life cycle.

We have described an RE framework as a small extension to an existing KE approach. RDR and FCA were used as subroutines within our 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. We envisage that full implementation would require no more than a total of 500 extra lines. Thus, our 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 our discussion on resolution strategies (Figure 2), we 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.

To instantiate our RE framework we have used the SISYPHUS III data. It has not been our goal to evaluate the accuracy of the KBS developed or their usefulness for classification or instruction, although we have shown that the MCRDR KBS are executable and argue that the concept lattices could act as a tutoring tool in helping the user learn about the domain. The goal of this paper 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. We have provided measures that show our 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.

References

Balzer, R. (1991) Tolerating Inconsistency *Proceedings of 13th International Conference on Software Engineering (ICSE-13)* Austin, Texas, USA, 13-17th May 1991, 158-165; IEEE Computer Society Press.

Brachman, R.J. (1979) On the Epistemological Status of Semantic Networks In Findler, N.V. (ed) *Associative Networks: Representation and Use of Knowledge by Computers* Academic Press-50.

Chandrasekaran, B. and Johnson, T. (1993) Generic Tasks and Task Structures In David, J.M., Krivine, J.-P. and Simmons, R., editors *Second Generation Expert Systems* pp: 232-272. Springer, Berlin.

Colomb, Robert.M. (1993) Decision Tables, Decision Trees and Cases: Propositional Knowledge-Based Systems *Technical Report No. 266* Key Centre for Software Technology, Department of Compter Science, The University of Queensland, Australia.

Compton, P. and Jansen, R., (1990) A Philosophical Basis for Knowledge Acquisition. *Knowledge Acquisition* 2:241-257

Compton, P., Edwards, G., Srinivasan, A., Malor, P., Preston, P. Kang B. and L. Lazarus (1992) Ripple-down-rules: Turning Knowledge Acquisition into Knowledge Maintenance *Artificial Intelligence in Medicine (*4)47-59

Easterbrook, Steve (1991) Elicitation of Requirements from Multiple Perspectives *PhD Thesis*, Department of Computing, Imperial College of Science, Technology and Medicine, University of London, London SW7 2BZ.

Easterbrook, S. and Nuseibeh, B. (1996) Using Viewpoints for Inconsistency Management *BCSEEE Software Engineering Journal* January 1996:31-43.

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

Finkelstein, A.C.W., Goedicke, M., Kramer, J. and Niskier, C. (1989) Viewpoint Oriented Software Development: Methods and Viewpoints in Requirements Engineering In *Proceedings of the Second Meteor Workshop on Methods for Formal Specification* Springer Verlag, LNCS.

Finkelstein, A., Gabbay, D., Hunter, A., Kramer, J. and Nuseibeh, B. (1994) Inconsistency Handling in Multi-Perspective Specifications *IEEE Transactions on Software Engineering* 20(8):569-578.

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

Gil, Yolanda and Tallis, Marcelo (1997) A Script-Based Approach to Modifying Knowledge Bases In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Application of Artificial Intelligence Conference* AAAI Press/ MIT Press, Cambridge, Massachusetts.

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

Herlea, D.E. (1996) User's Involvement in the Requirements Engineering Process *Proceedings 10^{th} Knowledge Acquisition Workshop,* Banff, Canada.

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

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

Menzies", T. (1997) 35 Kinds of Knowledge Maintenance, *Computer Science Technical Report. Submitted to the Knowledge Engineering Review.* Available from URL http://www.cse.unsw.edu.au/~timm/pub/docs/97km.ps.gz.

Narayanaswamy, K. and Goldman, N. (1992) "Lazy Consistency": A Basis for Cooperative Software Development *Proceedings of International Conference on Computer-Supported Cooperative Work (CSCW'92)* Toronto, Ontario, Canada, 31 October- 4 November, 257-264; ACM SIGCHI & SIGOIS.

Nebel, B. (1991) Terminological Cycles: Semantics and Computational Properties In John Sowa (ed) *Principles of Semantic Networks: Explorations in the Representation of Knowledge* Morgan Kaufmann Publishers, Inc. California, 331-361.

Patil, R. S., Fikes, R. E., Patel-Schneider, P. F., McKay, D., Finin, T., Gruber, T. R. and Neches, R., (1992) The DARPA Knowledge Sharing Effort: Progress Report In C. Rich, B. Nebel and Swartout, W., *Principles of Knowledge Representation and Reasoning: Proc.of the 3rd Int. Conference* Cambridge, MA, Morgan Kaufman.

Pirlein, T and Struder, R., (1994) KARO: An Integrated Environment for Reusing Ontologies *European Knowledge Acquisition Workshop '94*, Springer Verlag

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

Ramesh, B. and Dhar, V. (1992) Supporting Systems Development by Capturing Deliberations During Requirments Engineering *IEEE Transactions on Software Engineering* 18(6):498-510.

Richards, D and Compton, P, (1997a) Finding Conceptual Models to Assist Validation *Proceedings of AAAI'97 Verification and Validation Workshop* Technical Report WS-97-01, July 28, Providence, Rhode Island. AAAI Press: Menlo Park, CA, pp:53-62.

Richards, D. and Compton, P. (1997b) Uncovering the Conceptual Models in Ripple Down Rules In Dickson Lukose, Harry Delugach, Marry Keeler, Leroy Searle, and John F. Sowa, (Eds) (1997), *Proceedings of the Fifth International Conference on Conceptual Structures (ICCS'97),* August 3 - 8, University of Washington, Seattle, USA, Lecture Notes in Artificial Intelligence, Springer-Verlag, Number 1257, Berlin pp:198-212

Rumbaugh, James (1994) Getting Started: Using Use Cases to Capture Requirements *JOOP* September 1994:8-23.

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

Schwanke, R.W. and Kaiser, G.E. (1988) Living with Inconsistency in Large Systems *Proceedings of the International Workshop on Software Version and Configuration Control* Grassau, Germany, 27-29 January 1988, 98-118;B.G. Teubner, Stuttgart.

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

Shaw, M.L.G., (1980) *On Becoming a Personal Scientist: Interactive Computer Elicitation of Personal Models of the World* Academic Press, London.

Soloway, E, Bachant, J. and Jensen, K. (1987) Assessing the Maintainability of XCON-in-RIME: Coping with Problems of a very Large Rule Base *Proceedings of the Sixth International Conference on Artificial Intelligence* Vol 2:824-829, Seattle, WA: Morgan Kaufman.

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

Strauss, A. (1978) *Negotiation: Varieties, Contexts, Processes and Social Order* Jossey-Bass Publishers, San Francisco, CA.

Thomas, K. (1976) Conflict and Conflict Management In Duneette (ed) *Handbook of Industrial and Organisational Psychology* Rand McNally College Publishing Co.

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

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