

Giving Experts Ownership and Control over their Knowledge

Debbie Richards
Department of Computing
Division of Information and Communication Sciences
Macquarie University
richards@ics.mq.edu.au

ABSTRACT

A high degree of user control has long been recognized as an important factor in the acceptance of data and information-based computer systems. Users of knowledge-based systems (KBS) also desire control but the complex approaches developed to address some of the limitations of KBS has resulted in systems controlled by knowledge engineers rather than the end-user. We offer a knowledge acquisition and representation technique, known as Ripple-Down Rules, where the user can easily and rapidly capture, maintain and validate knowledge and perform inferencing. Through the recent incorporation of a set-theoretical approach to data analysis known as Formal Concept Analysis the user is also able to interact with the knowledge to perform the activities of critiquing, 'what-if' analysis, tutoring, explanation and modelling. The user chooses the type of interaction according to their decision situation and personal preference.

1. A brief History of HCI in KBS

A high degree of user participation in the system development process and user control have been long recognized as important factors in the acceptance of data and information-based computer systems [35]. The dramatic rise of the microcomputer in the 1980s, led by the spreadsheet package Lotus 123 and database package Dbase IV testified that users want to have ownership and control, even if the systems they build are unreliable. When it comes to knowledge, control is even more important to users [12] whether they be domain experts or novices [18,20,21,22]. Like the spreadsheet and database, knowledge based systems (KBS), known more commonly then as expert systems (ES), were launched into the market in the 1980s in an atmosphere of great expectation. However, KBS did not meet with widespread acceptance. Many, small and large, knowledge-based companies collapsed as KBS were

unable to deliver the anticipated goods. While some of the failure was due to overmarketing and promising something that the technology was not yet capable of, limited acceptance of first generation KBS by users was largely due to a lack of consideration of human-computer interaction (HCI) issues [21,33]. Some developers had assumed that because KBS were to be primarily used by professional people that they were exempt from HCI concerns and that the typical question-answer-conclusion style of interaction was adequate. Clancey [5] considers the typical consultation interface of KBS to be superficial because the line of questioning approach does not allow the user to build a model that they can manipulate. Kidd and Sharpe [20] also argue that earlier systems did not really solve the users' problem. Users don't only want to know "what is the fault?" or "what is the remedy?" They want to ask questions and negotiate a remedy. The problem has been that KBS have been designed from the machine's viewpoint and lack the scope and robustness needed by users with a wide range of needs [33].

The saddest part of this story is that the history of HCI in KBS has changed very little since the 1980s. While KBS research has progressed and some of the technical limitations of earlier systems have been resolved there has been an even greater shift in focus away from the end user. In the 1980s many, if not most, of the ES available were *shells*, which provided an inference engine, an interface and an empty knowledge base (KB). The user was guided by the system to enter conclusions, attributes, values and to develop production rules in the form IF... THEN... While the user had little or no control over the way the knowledge was presented and the interfaces and explanations tended to be poor, at least the user did have ownership of the knowledge. In addressing the various limitations of first generation KBS, researchers have focused on the technical or usefulness problems and ignored usability issues. Three major problems experienced by first generation KBS include: the knowledge acquisition (KA) bottleneck, the

problems associated with the maintenance of medium-large KBS and the brittleness problem [18]. In attempting to address these issues, the focus in the 1990s of most KBS research has been on complex modeling at the Knowledge Level [23]. The techniques offered to address these problems include the reuse of general problem solving methods and ontologies. While, reuse approaches do not overcome the inability of experts to articulate their knowledge they offer patterns of knowledge, which provide structure and guidance. Reuse also means that KA needs to be performed less often. However, the methodologies developed were complex and required a skilled (team of) knowledge engineer(s) to develop the system. Many approaches require the knowledge engineer to acquire the knowledge from domain experts and encode it for use by the KBS. This has exacerbated the lack of attention to HCI issues and has resulted in systems where the end-user is not a direct participant but must interact with the system via the mediation of the knowledge engineer. This third party situation does not promote a sense of ownership or control by the end-user.

This paper suggests that reuse of knowledge is a key issue for end-users but not the type of reuse mentioned above. The system described in this paper seeks to give the user ownership and control of their knowledge. Since control is made possible via by the user interface [16] we offer a number of alternative interfaces that may be accessed within the same session. The knowledge captured for one purpose, such as for consultation, may be reused by the user to explore and view the knowledge in a wide range of interaction styles. The user chooses the type of interaction according to their decision situation and personal preference. Through the use of a knowledge acquisition and representation technique known as ripple-down rules (RDR) [8] it is possible for the user to easily and rapidly capture, maintain and validate knowledge and perform inferencing. The range of activities supported by RDR have been increased through the incorporation of a set-theoretical approach to data analysis known as Formal Concept Analysis (FCA) [40]. FCA allows an abstraction hierarchy of concepts, known as a concept lattice, to be automatically generated from an RDR assertional KBS. The additional knowledge uncovered supports the activities of critiquing, 'what-if' analysis, tutoring, explanation and modelling.

While modeling still predominates KBS research, there are some, including the authors, who are concerned with the context-dependence and socially situated nature of expertise [1, p.240]. Clancey [6] and Collins [7] argue that what we do and say only makes

sense in our social context. An expert cannot exist without a social context responsible for conferring expert status. This new focus is important for KBS HCI because it acknowledges that human factors are essential parts of the system. A social view of knowledge prompts the question "how can we embed our concepts into a computer and use them when computers don't share our social view?" Collins [7] suggests treating the KBS as a valuable assistant where the expert takes advantage of the machines ability to store and process more information and heuristics than a human mind but the inputs and outputs are blended by the expert to fit the social context. A social focus requires a change from the traditional consultation style KBS to what can be termed Expert Advisory Systems [37]. Involving the human in the loop, however, places a new requirement for KBS to support user cognitive tasks. We have borne this in mind in the development of our system by making interaction with the system as close as possible to the way they perform that task manually.

The approach we propose is based on the view that knowledge is never complete and is made-up to fit the situation. The technique is particularly designed to allow easy maintenance of the knowledge by the domain expert. Cases are used to provide context and to ground the knowledge in the real world. The initial model captured is very low level and is based on these cases. We do not seek to understand the reasoning processes of the expert we simply capture their expert behaviour. Unlike systems based on user models, we impose minimal structure on how the user interacts with the system and do not offer explanations based on our view of the knowledge or of them.

2. The Many (inter)Faces of RDR

The RDR system described in this paper has many alternative interfaces. These interfaces do not simply equate to different screens but are different ways of interacting with and exploring the knowledge base. Figure 1 shows a Menu-list of options available to the user. The first 11 options (up to Utils) were available in the original multiple classification RDR (MCRDR) prototype system. The remaining 10 options were added in seeking to provide a wider selection of interaction modes. Typical of many KBS, the HCI focus in the original MCRDR system was on acquisition of the knowledge and performing inferences on that knowledge, which were provided by the first 5 menu options. The RDR approach to KA is novel in comparison with other KBS and is discussed in greater detail in section 2.1. The remaining 6 options provided other functions such as statistics and rule browsing.

Various statistics about such things as the size and shape of rules, the number of rules that gave a particular conclusion, a list of rules which used a particular condition, were also available. The explanations provided consisted of rule traces. Rule traces have been found by users to be hard to follow and unlike human language. The traces offered by RDR were typically less confusing since RDR do not use intermediate rules or disjunctions of conditions and more intuitive due to the exception structure which has been considered to be a useful mediating representation between machine and human [2]. The approach offered by RDR involved minimal analysis and was a simple technique designed to be used by the domain expert. This worked very well for KA and inferencing but we found that the lack of higher level models made it difficult to support certain (re)uses of the knowledge. Models are appropriate components of explanation, instruction and exploration interfaces by assisting the user in understanding the conceptual structure of their domain [5,34,41]. The approach we have developed involved the retrospective and automatic generation of abstraction hierarchies, using formal concept analysis (FCA) [40] and cluster analysis, based on the primitive knowledge in the RDR KB. This hierarchy can be used to support such tasks as inferencing and KA in a critiquing interface, letting the user explore different scenarios by manipulating input values to support 'what-if' analysis, showing the user how a new rule they are proposing to add (in the form of conclusions and rule conditions) fits in or appears inconsistent with the existing knowledge and to provide an improved explanation facility for such purposes as tutoring or causal modeling.

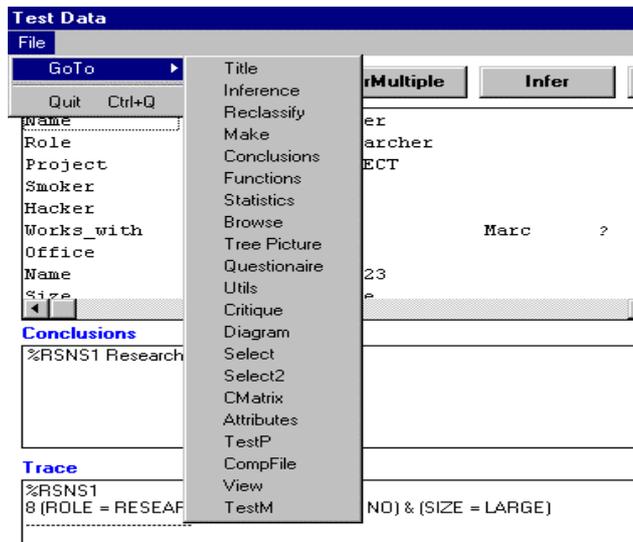


Figure 1: The Inference (Test Data) Screen with Menu Droplist of other available interfaces.

2.1 Knowledge Acquisition, Maintenance and Validation

The founders of RDR [8] wanted to provide a representation and KA technique where the maintenance effort was reduced to a manageable task that could be performed by the domain expert. A major consideration to the approach developed was the observation that experts did not provide explanations of why they chose a particular course of action but rather they gave a justification for their decision and the contents of that justification depended on the context, specifically the audience [9,25].

Interaction with the system is simple and the user does not need to switch between applications to perform system development, KA, maintenance and inferencing as they are treated as parts of the one task. New rules are added (KA and maintenance) in response to a case being misclassified (an inference). KA involves the expert viewing a case with a system assigned conclusion. If the user does not agree with the classification, he/she assigns the correct classification to the case and picks some feature/s in the case that justify the conclusion and which form the conditions in the new rule. In most KBS the person responsible for KA would be different to the person who uses the system for inferencing. To assist the user with KA and provide context and on-line validation, the case that prompted a new rule to be added is stored in association with the new rule and is referred to as the cornerstone case. When a misclassification occurs, the cornerstone case for the rule that fired is shown along with the current case and the user must pick some feature/s which distinguish the two cases. This ensures that no previously correctly classified case becomes misclassified.

The ease with which systems can be developed without the need for complex analysis of the domain as a prerequisite or major assistance from a knowledge engineer is what makes RDR such a different paradigm to mainstream KBS research. We see the reliance on models for KA as error-prone due to the difficulty of capturing such models and the inherent unreliability of models [4,14]. In accordance with the observed social and situated nature of expertise, we prefer to capture knowledge in the way that experts exhibit their expertise, that is, going straight from data to conclusions without modeling as a prerequisite to the capture of expertise as rule-base implications. See [9] for a discussion of analysis-free KA. The argument over whether models must be developed first is relevant to HCI because the complex and time-consuming effort involved in developing these models has resulted in a KBS research focus on describing the domain and

problem solving method knowledge at the expense of user-driven and centred research. By taking a simple, yet reliable, approach to KA we can concentrate more on user needs.

Other work based on Personal Construct Psychology [19] using Repertory Grids [14] and formal contexts in Formal Concept Analysis [40] also minimizes reliance on models and knowledge engineer support but does not support on-line development and the evolution of knowledge since they require up-front consideration of the whole domain.

The major empirical support by users for RDR is in the medical domain of clinical pathology in the Pathology Expert Interpretative Reporting System (PEIRS) [10,11]. The system covers up to 200 analytes (substances measured) with the results for about 20 analytes at five sample time intervals on each report. For a description of our handling of time-course data see [26]. This system went into routine use in a large Sydney hospital with 198 rules and grew on-line to over 2000 rules in a four-year period (1990-1994). Maintenance was performed by the medical expert without the intervention of a knowledge engineer. PEIRS provided comments for about 20 percent of the 500 reports issued each day. Each report was reviewed by a medical expert resulting in four to five corrections each day meaning the system is 95% accurate. Each rule took about three minutes to add, most of that time being taken up in deciding on the wording of the conclusion or in locating an existing suitable conclusion. This constitutes a development time of around 100 hours for the 2000+ rule-base. This result is in marked contrast to the two-to-three rules per day typically associated with the maintenance of medium to large KBS [8].

The success of RDR in the pathology has largely been attributed to the way HCI is handled [11]. Maintenance, which is initiated and performed by the user, satisfies the criteria of user ownership and control. The natural way in which knowledge is acquired fits the mental model [24] of the expert, that is the process of “assign conclusion - pick some features in the case to form the rule” is compatible with how the expert performs that task in real life. The system produced a substantial savings in the time of the expert and an increase in reliability since there were in effect two experts, one human the other machine, that were agreeing on a diagnosis. The requirement for cases as the basis for KA is compatible with the pathology domain. The stipulation that the user review each comment on the pathology report is not seen as a limitation but an approach consistent with the situated view of KBS as a support not a prosthesis [33].

2.2 Critiquing and ‘What-if’ Analysis

“Second opinion” system styles, like critiquing or -if’ analysis, are an alternative to the more static consultation interface. Both system styles give the user a different way of interacting. The choice of which interface to use depends on the user’s decision style, current situation and personal preferences. ‘What-if’ analysis lets the user pose the questions. Critiquing lets the user pose the answer which the system critiques against its own solution. Both approaches let the user focus on what is of interest and relevance to themselves.

‘What-if’ analysis is supported in RDR by allowing the user to alter any case or create their own case. This is done without the need to swap into a maintenance mode or to other screens. The changes made to the case are only temporary and will not affect the actual data. If the user wishes to add the new case permanently they can do so by updating the case/data file. Having changed the input data, the case, they then proceed as usual.

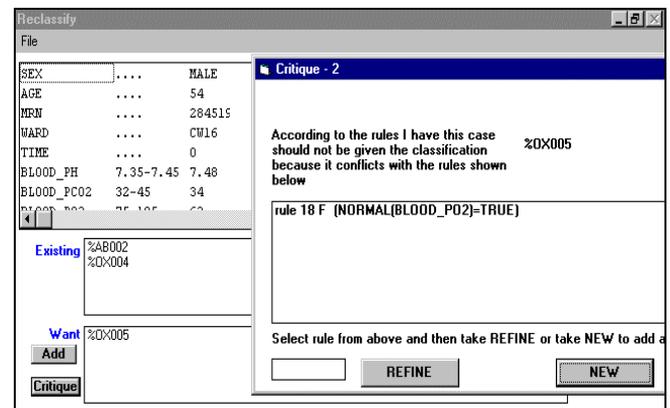


Figure 2: Using MCRDR/FCA to critique the proposed conclusion %O005 for the current case against the rules that already give that conclusion.

A more sophisticated type of ‘what-if’ query, described in more detail below as a type of critiquing, is the ability of the user to add a potential rule and then see how that knowledge fits in with the existing knowledge and build models using the proposed rule. If they decide not to add the rule they carry on with their next task and the temporary rule is not added. If they decide to go ahead and add the rule they click ADD the same as if they had not performed ‘what-if’ analysis with the rule.

There are two main concepts in the KBS that are critiqued: the conclusion and the rule conditions. The user may choose one of two ways to critique a conclusion. In figure 2 the user has run an inference on a case, decided the conclusion was incorrect, taken the command button RECLASSIFY, entered the new

conclusion and then clicked the CRITIQUE command button. If the user preferred they could have chosen an alternative inference screen which let them input the conclusion/s they thought most appropriate and then have the system advise them of which conclusions were in agreement or disagreement with the system's inferred conclusions. The latter method of performing critiquing is more in keeping with the Attending [22] and Oncocin [21] critiquing systems where the user inputs their plan first and then the system determines if there is a discrepancy. Regardless of how the user initiates critiquing, the pop-up window in Figure 2 shows the user which existing pathways, if any, in the KBS conflict with the proposed solution. The user may amend one of the rules displayed by adding an exception rule to it, adding a new rule higher in the tree or deciding not to add the critiqued conclusion to the current case.

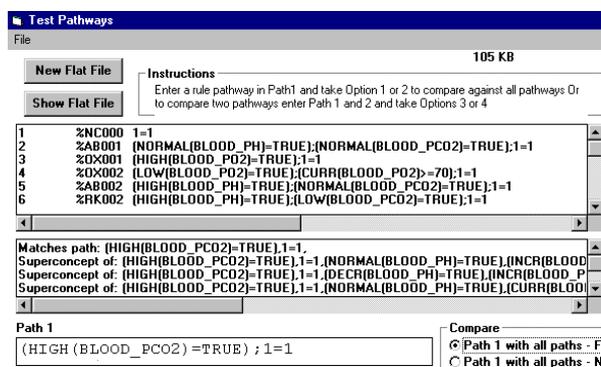


Figure 3: Critiquing proposed rule conditions against existing knowledge in MCRDR/FCA.

If the user decides to add a new rule, be it an amendment or new pathway, they click the command button MAKE RULE and, as explained previously, pick the features in the current case which differentiate it from the cornerstone cases associated with the rule that gave the misclassification. Once the rule has been formed the user may add that rule immediately or they may take an EVALUATE RULE command button which shows them how the proposed rule fits in with the existing knowledge. The user can select to compare the proposed rule with the concepts derived using Formal Concept Analysis. This option shows a listbox of subconcepts, superconcepts or matches on the proposed pathway (rule). See Figure 3. Alternatively, they can use a nearest neighbour algorithm that provides a score between 0-1 of the similarity between each existing pathway and the new pathway. When the user chooses either the formal concept analysis or nearest neighbour algorithm option, the user is shown the pathway (or concept) and conclusion being compared so that they can

assess intuitively whether the new rule appears to fit in appropriately. A benefit of using a critiquing interface is that they can offer more useful and focused explanations because they don't attempt to explain the whole system but only those parts of the system relevant for the plan under review [22].

2.3 Explanation, Tutoring, Querying and Views

The division of uses into interfaces is arbitrary and, as mentioned above, critiquing and 'what-if' analysis can be viewed as types of explanation. In this subsection we extend our notion of explanation to include such interfaces as tutoring and modeling (causal modeling being a particular case of modeling which may not be relevant to all domains). While intelligent tutoring systems typically have a model of the user to guide the learning process we argue that individualized models are costly to capture and become outdated and stereotypical models may be inappropriate and hard to apply. From a situated view of knowledge, we prefer to give the user sufficient freedom and tools that allow exploration of the knowledge by letting the user select different views and ask a wide range of questions according to their needs as a means of teaching. The explanations provided through the line diagrams drawn using Formal Concept Analysis can be based on a wide range of views. There are currently 13 different options including selection by conclusion, attribute, attribute-value pairs, conclusion families and so on. These views may be combined. Our motivation for providing such an extensive range of views is twofold. Firstly, we believe users need the flexibility offered by the choices and secondly because if the focus of attention is not narrowed then the diagrams produced will have too much information and will be incomprehensible.

Explanation in RDR may be given in two main ways. The first is the well-known rule trace. Clancey [3] found that for explanation or learning purposes KBS "need to articulate how rules fit together [and] how they are constructed" [3, p.59]. This is supported by the exception structure of an RDR KBS but is difficult to determine in conventional KBS because of complex interaction of rules and the numerous possible pathways to arrive at a conclusion. Another point made by Clancey [2] is that students should not just be able to confirm a diagnosis but should be able to learn under what circumstances that conclusion should be considered and what other possible diagnoses explain the data. Similarly, Swartout and Moore [38] argue that in most systems the *deeper knowledge* needed for explanation is *compiled out* and lost at development time. This loss of information is not so severe in an RDR KBS since a

history of the evolution of the rule base is stored in the tree structure and associated cornerstone cases.

Even more powerful tools for explanation are the concept matrices and lattices that can be derived using formal concept analysis. The concept lattice gives a graphical representation of the concepts embedded in the rules of the KBS, showing both low level concepts as well as a hierarchy of abstractions. This ability to show the abstractions in the KBS that have not been explicitly encoded is significant. It is this sort of knowledge that is used by experts and needed by novices but it is also knowledge that is often difficult for experts to articulate or novices to learn.

To give a taste of how the modeling tools may be useful for explanation we consider the following query: "What conclusions are conceptually close to the conclusion %OX005 and what are the relevant higher concepts or abstractions?". To ask this question the user enters the conclusion code in the appropriate input box and clicks the CONCLUSION radiobutton. In Figure 4 we can see that the top concept contains rule 18 which concludes %OX005 and subsumes the rules and conclusions on descending paths.

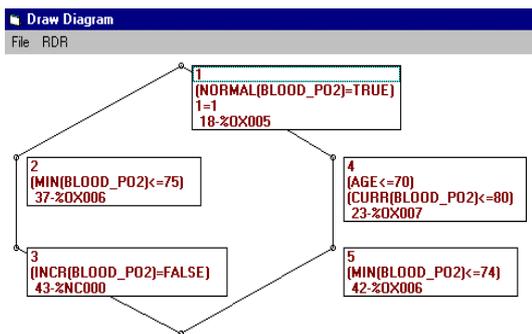


Figure 4: The line diagram showing which rules are close to the conclusion %OX005 for the blood gases domain.

The rule condition for concept 1 is (NORMAL(BLOOD_P02)=TRUE) and is inherited by all lower concepts, as all rule conditions for a concept are reached by ascending paths. Thus, concept No. 4 includes the conclusions %OX007 and %OX006 from rules 23 and 42, respectively, and has the rule conditions: (NORMAL(BLOOD_P02) = TRUE; (AGE <= 70); and (CURR(BLOOD_P02)<=80) & (MIN(BLOOD_P02)<= 74). Thus in answer to our original question we can say that conclusion %OX006 and %OX007 are close to the conclusion %OX005. The attributes shown on the line diagram explain in what way and to what degree the conclusions are close. This

also gives the user the insight into what attributes would make the conclusion swing from one diagnosis to another which they could further explore using 'what-if' analysis. If a particular attribute or group of attributes appeared to warrant further investigation they could add those attributes to the view so that then they could look at the parts of the knowledge close to the conclusion %OX005 as well as those parts using the specified attributes.

3. Future Directions

There are a number of enhancements that can be made to the MCRDR/FCA prototype tool. Some general changes which should be made to the user interface relate to the limitations on the amount of information that can be displayed comprehensibly at one time. Important enhancements include the ability to drop nodes from the lattice or to zoom in and out using selections made via the concept lattice to determine what new contexts and concept lattices should be developed. Use of the fish-eye graph-drawing technique [17] may be desirable.

Little has been said regarding the utility of the line diagram. Evaluation has been performed along a number of dimensions and reported in detail in [28]. In a study involving 12 computer science graduates it was found that reading the line diagram could be learnt and the knowledge represented and reasoned about in minutes. Four case studies were also conducted in four different domains: agriculture, chemistry, geology and pathology. In these studies a domain expert was asked to comment on the knowledge gleaned by a domain beginner (a level lower than novice) when exploring KB's built for these domains by other people. It was found that the concept lattice opened up a channel of communication and opportunity for learning which was not possible by simply looking at the rules or rule traces. The generality of the approach to other KR and the value of the concept lattice as a taxonomic and ontological representation have also been evaluated [29].

Extensions have been made to MCRDR/FCA which allow the user to develop a number (currently up to four) concept lattices and then to be shown a concept lattice of the similarities and differences between the four models. The user may also load the four KBS so that they can view the case associated with a particular rule to be used as a counterexample in possible discussions between owners of the different KBS. This feature was preliminary work to more recent work in the area of requirements engineering [30,31] which also offers various negotiation strategies and resolution operators. We are currently seeking significant funding for this

application to allow further development and extensive evaluation with real users.

While HCI issues have taken a backseat in the KBS community, the work being done at Stanford is a noteworthy exception. The PROTÉGÉ family of systems [15] grew from a focus on the role of the domain-expert and a desire to reduce reliance on a knowledge engineer. However, PROTÉGÉ's approach follows mainstream research and emphasizes the role of ontologies and methods to provide mappings between the KB and problem solving method. Additionally, PROTÉGÉ is primarily concerned with the task of knowledge acquisition (KA) [27] rather than providing multiple ways of accessing and exploring knowledge by the user for a wide range of tasks. From an HCI point of view, the work in KNAVE [36] is similarly oriented to the system described in this paper. KNAVE facilitates acquisition, maintenance, sharing and reuse of temporal-abstraction knowledge in a visual environment. The use of visualisation and multiple views is similar to the work reported in this paper but is based on a very different KBS paradigm. Another difference is that temporal abstraction has not been explored using MCRDR/FCA. The approach offered in [26] to handling time-course data involves preprocessing of the data before inferencing to determine such things as TEMP=INCREASING. For the tasks of planning, monitoring and explanation it may be desirable to use the unprocessed data as the input to the formal context and to discover temporal abstractions in the data. This may also provide further insight into the acquisition of temporal knowledge.

The concepts behind the MCRDR/FCA prototype are more noteworthy than the tool itself. The approach demonstrates that unlike most KBS approaches, it is feasible for domain experts to incrementally, rapidly and easily acquire knowledge and to later model that knowledge for alternative uses such as explanation or tutoring. Just as RDR stresses the incremental nature of KBS development, MCRDR/FCA systems are not expected to be complete but able to evolve. This evolution allows new activities to be added according to user requirements. RDR allows the user to capture a simple observable model of their world using attribute-value pairs and conclusions. The user has ownership of the KB developed. Using FCA, the user is able to retrospectively view and explore their underlying conceptual models. The simplicity of the approach and the range of activities and views supported in MCRDR/FCA make the tool applicable to a wide range of decision styles and situations with the user in control of the way they interact with their knowledge.

Acknowledgements

Thanks to Paul Compton and Rudolf Wille for their encouragement in this work. RDR research is supported by various grants from the Australian Research Council.

REFERENCES

- [1] Agnew, N.M., Ford, K.M. and Hayes, P.J. (1997) Expertise in Context: Personally Constructed, Socially Situated, and Reality-Relevant? Feltovich, Paul J., Ford, Kenneth, M. and Hoffman, Robert R. (eds) *Expertise in Context: Human and Machine* AAAI Press/MIT Press, Cambridge, Massachusetts, 219-244.
- [2] Catlett, J. (1992) Ripple-Down-Rules as a Mediating Representation in Interactive Induction *Proceedings of the Second Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop*, Kobe, Japan, Nov 9-13 1992, 155-170.
- [3] Clancey, W.J., (1984) Methodology for Building an Intelligent Tutoring, Kintsch, W, Miller, J. and Polson, P. (eds) *Methods and Tactics in Cognitive Science* Lawrence Erlbaum Associates, New Jersey, pp.51-83.
- [4] Clancey, W.J., (1992) Model Construction Operators *Artificial Intelligence* 53:1-115.
- [5] Clancey, W.J., (1993) Situated Action: A Neurological Interpretation Response to Vera and Simon *Cognitive Science*, 17: pp.87-116.
- [6] Clancey, W. (1997) The Conceptual Nature of Knowledge, Situations and Activity In Feltovich, Paul J., Ford, Kenneth, M. and Hoffman, Robert R. (eds) *Expertise in Context: Human and Machine* AAAI Press/MIT Press, Cambridge, Massachusetts, 248-291.
- [7] Collins, H.M. (1997) RAT-Tale: Sociology's Contribution to Understanding Human and Machine Cognition In Feltovich, Paul J., Ford, Kenneth, M. and Hoffman, Robert R. (eds) *Expertise in Context: Human and Machine* AAAI Press/MIT Press, Cambridge, Massachusetts, 293-311.
- [8] Compton, P. and Jansen, R., (1990) A Philosophical Basis for KA. *Knowledge Acquisition* 2:241-257.
- [9] Compton, P., Kang, B., Preston, P. and Mulholland, M., (1993), Knowledge Acquisition without Analysis, *European Knowledge Acquisition Workshop*, Springer Verlag
- [10] Edwards, G., Compton, P., Malor, R, Srinivasan, A. and Lazarus, L. (1993) PEIRS: a Pathologist Maintained Expert System for the Interpretation of Chemical Pathology Reports *Pathology* 25: 27-34.
- [11] Edwards, Glenn (1996) Reflective Expert Systems in Clinical Pathology *MD Thesis*, Uni. of New South Wales.
- [12] Freidson, E. (1994) Nourishing Professionalism *Professionalism Reborn: Theory, Prophecy and Policy*. Chicago: University of Chicago Press, 199-216.
- [13] Gaines, B. R. and Shaw, M.L.G. (1983) Dialog Engineering In *Designing for Human-Computer Communication* Academic Press, :23-53.

- [14] 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.
- [15] Grosso, W. E, Eriksson, H., Ferguson, R.W., Gennari, H. Tu, S.W. and Musen, M. (1999) Knowledge Modelling at the Millenium (The Design and Evolution of Protégé-2000) *In Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management (KAW'99)*, 16-21 October, 1999, Banff.
- [16] Hender, J.A. and C. Lewis, (1988) Introduction: Designing Interfaces for Expert Systems in Hendler, J. A., *Expert Systems: The User Interface* Ablex Pub.Corp.
- [17] Hollands, J.G., Carey, T.T., Matthews, M.L. and McCann, C.A. (1989) Presenting a Graphical Network: A Comparison of Performance using Fisheye and Scrolling Views *Proceedings of 3rd International Conference on Human Computer Interaction* September 1989, 313-320.
- [18] Ignizio, James, P. (1991) *Introduction to Expert Systems: The Development and Implementation of Rule-Based Expert Systems* McGraw-Hill Inc, New York.
- [19] Kelly, G.A. (1955) *The Psychology of Personal Constructs* New York, Norton.
- [20] Kidd, A. L. and Sharpe, W. P. (1987) Goals for Expert System Research: An Analysis of Tasks and Domains *In Expert System IV* Cambridge Uni. Press, 146-152.
- [21] Langlotz, Curtis P. and Shortliffe, Edward E. (1983) Adapting a Consultation System to Critique User Plans. *Int.l Journal of Man-Machine Studies* 19:479-496.
- [22] Miller, P.L. (1986) *Expert Critiquing Systems: Practice-Based Medical Consultation. by Computer* Springer-Verlag, NY.
- [23] Newell, A. (1982) The Knowledge Level *Artificial Intelligence* 18:87-127.
- [24] Norman, D.A.(1986) Cognitive Engineering In Norman, D.A. and Draper, S.W. (eds.) *User Centered System Design: New Perspectives on Human-Computer Interaction* Lawrence Erlbaum Associates.
- [25] Patel, V.L. and Ramoni, M.E. (1997) Cognitive Models of Directional Inference in Expert Medical Reasoning In Feltovich, Paul J., Ford, Kenneth, M. and Hoffman, Robert R. (eds) *Expertise in Context: Human and Machine* AAAI Press/MIT Press, Cambridge, Massachusetts, 67-99
- [26] Preston, P., Edwards, G., Compton, P. and Litkouhi, D. (1994) An Expert System for Time Course Data with Refinement in Context *Artificial Intelligence in Medicine: Interpreting Clinical Data AAAI 1994, Spring Symposium Series*, Stanford University, pp:121-124.
- [27] Puerta, A. R, Egar, J.W., Tu, S.W. and Musen, M.A. (1992) A Multiple Method Knowledge Acquisition Shell for Automatic Generation of Knowledge Acquisition Tools *Knowledge Acquisition* 4(2):171-196.
- [28] Richards, D. (1998) The Reuse of Knowledge in Ripple Down Rules Knowledge Bases Systems, *PhD Thesis*, University of New South Wales.
- [29] Richards, D. (1999) Visualising Knowledge-Based Systems *Proceedings of the Third Workshop on Software Visualisation*, Dec 3-4, 1999, University of Technology, Sydney, 1-8.
- [30] Richards, D. and Zowghi, D. (1999) Maintaining and Comparing Requirements *Proceedings of the Fourth Australian Conference on Requirements Engineering (ACRE'99)* 29-30 September, 1999, Macquarie University, Sydney, 115-130.
- [31] Richards, D. and Menzies, T. (1998) Extending the SISYPHUS III Experiment from a Knowledge Engineering to a Requirements Engineering Task In Gaines, B. and Musen, M. (eds) *11th Workshop on Knowledge Acquisition, Modeling and Management (KAW'98)*, Banff, Canada, SRDG Publications, Departments of Computer Science, University of Calgary, Calgary, Canada, Vol 1:SIS-6.
- [32] Roth, E.M., Bennett, K.B. and Woods, D.D. (1987) Human Interaction with an 'Intelligent' Machine *International Journal of Man-Machine Studies* 27:479-525.
- [33] Salle, J.M. and Hunter, J., (1990) Computer/User Cooperation Issues for Knowledge-Based Systems: A Review, *Tech. Rep. AUCS/TR9003*, Aberdeen University
- [34] Schon, D.A. (1987) *Educating the Reflective Practitioner* Jossey-Bass, San Francisco, CA.
- [35] Schultheis, Robert and Sumner, Mary. (1992) *Management Information Systems: The Manager's View*. 2nd ed. Homewood: Irwin Publishers.
- [36] Shahar, Y. and Cheng, C. (1998) Model-Based Visualisation of Temporal Abstractions *Proceedings Fifth International workshop on Temporal Representation and Reasoning (TIME'98)*, Sanibel Island, Florida.
- [37] Stelzner, M and Williams, M.D., (1988) The Evolution of Interface Requirements for Expert Systems in Hendler, J. , *Expert Systems:The User Interface* Ablex Pub. Corp.
- [38] Swartout, W. and Moore, J. (1993) Explanation in second generation ES In David, J.M., Krivine, J.-P. and Simmons, R., editors *Second Generation Expert Systems* pp: 211-231. Springer, Berlin.
- [39] Vogt, Frank and Wille, Rudolf (1995) TOSCANA - A Graphical Tool for Analyzing and Exploring Data *Knowledge Organisation* 22:2,:78-81.
- [40] Wille, R. (1982) Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts In *Ordered Sets* (Ed. Rival) pp:445-470, Reidel, Dordrecht, Boston.
- [41] Zeitz, C.M. (1997) Some Concrete Advantages of Abstraction: How Experts' Representations Facilitate Reasoning In Feltovich, Paul J., Ford, Kenneth, M. and Hoffman, Robert R. (eds) *Expertise in Context: Human and Machine* AAAI Press/MIT Press, Cambridge, Massachusetts, 43-65.