Chapter 1

1 The Need to Reuse Knowledge

The motivation for the work reported in this thesis is the belief that not only is it beneficial to reuse knowledge but it is essential if we wish to build knowledge based systems (KBS) that meet the needs of users. Reuse can range over a number of dimensions (Menzies, personal communication) including:

- Reuse by the same/different person.
- Reuse over the same/different task.
- Reuse for the same/different purposes or activities.
- Reuse at the same/different time.
- Reuse over the same/different software application.

The focus of this research is on the reuse of knowledge for different activities. A more appropriate term for the type of reuse of interest is adapting or *repurposing*, a term Clancey has taken from the hypermedia field. Based on a situated view of expert knowledge and action and an emphasis on user-requirements, the goal of this thesis was to see if a knowledge based system could be built that allowed the user to select from a wide range of activities using and reusing knowledge according to their current purpose and decision situation. Activities are seen as related to a purpose with each activity designed to answer different types of questions relating to the same knowledge. An activity is seen as distinct from a task because a task in knowledge engineering (KE) is typically equated with a problem type such as diagnosis, planning or configuration. To differentiate a task and activity further we can view a task as related to making a decision in the world. An activity is not concerned so much with the world but with the individual (or group) and how they arrive at a decision. A task such as room allocation could be achieved through different activities such as consultation, learning about the domain, proposing a solution to have critiqued or exploring different scenarios. To avoid confusion with more conventional meanings of reuse, the type of reuse being pursued is referred to in this thesis as *activity-reuse.* While some have looked at building systems that perform one or a small number of activities, which could include knowledge acquisition (KA), consultation, maintenance, explanation, critiquing or ‘what-if’ analysis, this study is concerned with providing all of these activities and more within the one session because user requirements will differ between users and in different situations for the same user.

In addition to a different focus on the type of knowledge reuse, another distinguishing feature of this thesis is a breakaway from mainstream approaches to KBS development, which emphasize the need to build terminological KBS (T-boxes[1]) at the knowledge level. Such approaches require complex analysis and modelling of the ontologicial, problem solving and/or domain knowledge as prerequisites to system development. Developing complex models is time-consuming and the emphasis on capturing a "good" and comprehensive model before knowledge acquisition can begin contributes to the knowledge acquisition bottleneck. The difficulties associated with capturing ontological or problem-solving models is due to the well recognised fact that much expert action is subconscious and not easily described (Ignizio 1991) and the inherently unreliable nature of models. A descriptive model can be seen as: "merely an abstraction, a description and generator of *behaviour patterns over time*, not a mechanism equivalent to human capacity" (Clancey 1993, p.89). It has been shown that models are imperfect representations that vary not only between experts but also over time with the same expert (Gaines and Shaw 1989).

The work described in this thesis is based on a paradigm that captures a simple model in terms of attribute-value pairs and the related conclusions without the need for modelling of the data into an abstraction hierarchy or identification of the nature of the problem to be solved. The simple model is being captured into an assertional[2] (A-box) KBS which is a performance system that can be executed and validated. In keeping with the situated cognition view, this thesis describes a KBS that supports human-computer interaction in modes that seem to be more in keeping with the way a human may perform that activity. Just as most human action is reflexive (Winograd and Flores 1986) the system described in this thesis performs knowledge acquisition, maintenance and inferencing in a reflexive[3] mode without the need for reflective action on the part of the expert. However, reflective activities such as explanation, tutoring, 'what-if' analysis and causal modelling also need to be supported.

The knowledge acquisition and representation technique used as the foundation of the work reported in this thesis is known as Ripple-Down Rules (RDR) (Compton and Jansen 1990). RDR have been built from a situated standpoint which acknowledges that

---

[1] 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).

[2] Assertional KBS are made up of executable assertions (such as rules) that assert the relationships between terms (such as conditions and conclusions).

[3] The terms reflexive and reflective are used in this thesis to distinguish between acts that are automatic and often subconsciously performed and acts that require deliberative thought often involving the consideration of abstract models, respectively.

the capture of a performance model may be easier and more reliable than reliance on acquiring a complete descriptive model. The use of RDR as the vehicle for the *activity-reuse* research reported in this thesis has resulted in a refinement of the research goals. It was the goal of this research to see if knowledge captured for one purpose or activity such as KA could be reused for another activity such as critiquing with only changes to the user interface. Prior to this study, RDR could already successfully (Edwards et al 1993) handle a number of activities such as KA, inferencing, maintenance and validation all within the same system. These activities are performed in a reflexive manner by the expert with minimal consideration of the model being built. As described in Chapter Three, RDR develops rules by going straight from data to conclusions without the intermediate step of building an abstract model of the data (see the inference structure of heuristic classification given in Clancey (1985)). All rules in an RDR KBS consist of a final conclusion and a set of attribute-value pairs which are the rule conditions. The close link between the use of RDR and the goals of this research meant that the activities already supported required minimal further research effort and led to a focus on the limitations of RDR that were relevant to the *activity-reuse* of knowledge. The first limitation was the existence of repeated knowledge in an RDR KBS. While repetition is not excessive and does not greatly affect system performance, the removal of repetition was seen as beneficial for the activities involving explanation. This work is described in Chapter Four. The second limitation was the lack of higher level models which were necessary for the reflective modes such as critiquing, 'what-if' analysis and explanation. It became apparent that the key to supporting these activities was an understanding of the concepts in the KBS and the relationships between them. Others have found that models of abstraction are beneficial for teaching (Schon 1987) and for explanation (Clancey 1993). Of particular importance was the ability to find the higher concepts which were implicitly represented by the low-level concepts in the form of primitive rules. As described in Chapter Five, formal concept analysis (FCA) (Wille 1982) has been employed not only to uncover the higher-level abstractions in the RDR KBS but also the structure between all concepts in the KBS to support the reflective modes. To support determination of the closeness of concepts, a nearest neighbour algorithm was developed and is reported in Section 3.3.2.1.

Another activity that was not supported by RDR was the ability to compare the knowledge contained in multiple KBS acquired from multiple sources of expertise. The FCA work was extended to allow the development of a single KBS from multiple KBS. The ability to detect and manage conflicts between viewpoints can be seen as a requirements engineering (RE) activity. An RE framework which is an extension to the FCA work is described in Chapter Six. Thus it was the goal of this research to address these limitations of RDR in order to provide a KBS technique that supports a wide range of *activity-reuses*. Chapters Seven, Eight and Nine describe and evaluate the multipurpose system developed.

In the next section the benefits of reuse at a general level beyond *activity-reuse* are considered. Firstly, we look at the material or quantitative benefits of reuse followed by consideration of the more soft benefits of building systems that people want and are able to reuse.

1.1 The Benefits of Reuse

The reuse of software components has become widely accepted due to the following factors:

1. Savings in cost
2. Savings in time
3. An increase in reliability (Hemmann and Voss 1993a).

There are two main methods of software reuse: using what you already have as building blocks or using an existing pattern to create new code. Hemmann and Voss (1993a) term these approaches composition and generation respectively. The former is the more widely accepted practice and appears to be more natural for software engineers. In the knowledge engineering community there is a similar situation with much research into developing building blocks (e.g. Chandrasekaran 1986, Steels 1993) that can be reused for different applications. However, in both software and knowledge engineering reuse research the focus is on the technical person, the software or knowledge engineer (KE). The focus in this study is on the end-user or the expert who is not directly assisted by the building block approach.

There is much similarity between the efforts towards reuse in software engineering and knowledge engineering. The focus in software engineering on the reuse of programs and procedures is similar to the focus on the reuse of problem solving methods in knowledge engineering. Also the benefits of reusing the representation language and the method of inference for different applications are obvious and much research focuses on these aspects (Clancey 1992). However a valuable lesson from software engineering is that different languages will affect whether and how that component can be reused. Analogously in knowledge engineering, different KA and knowledge representation techniques may affect the reusability of that knowledge (Chandrasekaran and Johnson 1993).

Software components are not the only resources that can benefit from reuse. Data are considered to be a major resource and the explosion of the information age confirms its value. Database management systems (DBMS) are an important and necessary development to allow organisations to manage their data. Key benefits of DBMS are a reduction in redundancy and repetition. Data can also be captured once and reused for a whole range of purposes. Knowledge is value-added data and includes such things as experience, insight and skill. The use of data dictionaries to support the reuse of data is similar to the use of common ontologies for describing knowledge. The work done by Jansen and Compton (1988) on the knowledge dictionary is an even closer development of the data dictionary idea.

The benefits of reusing knowledge, which includes both domain and problem solving knowledge, should be even greater than the benefits of reusing data because of its increased value and the problems associated with acquiring knowledge (Ignizio 1991). By capturing expert knowledge once and reusing it a number of times, better use can be made of this scarce resource and the knowledge acquisition bottleneck can effectively be reduced.

As mentioned, this work is interested in reusing knowledge within the same domain to support multiple activities and sees the benefits of reuse for humans as substantially greater than the material benefits outlined above. To justify this claim, the need to support a range of activities must be shown. Let us take a brief look at the some of the limitations of first and second generation expert systems (ES)[4], Human Computer Interaction (HCI) issues for KBS, the importance of control, the types of systems needed and the implications of situated cognition on the development of KBS.

1.1.1 The Limitations of First Generation Expert Systems

It has been shown that even where the knowledge is accurate and reliable, the expertise captured becomes an underutilised and wasted resource if the knowledge can not be accessed by the user in an acceptable way (Langlotz and Shortliffe 1983). An indepth study by Salle and Hunter (1990) considers the lack of attention to computer and user cooperation issues to be the main reason for the poor acceptance of ES technology by end-users. Cooperation includes the user interface but is much more. The user interface is concerned with usability, whereas the mode of interaction is concerned with usefulness (Rector 1989). The term used by Salle and Hunter is modality and is defined as: "referring to different forms the cooperation may have regardless of agents being

[4] The term expert system is an older term that has been mostly superceded in the literature by the term knowledge based system and is often used to avoid the stigma associated with the limitations of first generation ES. Sometimes the distinction is made that KBS encompass more than ES because the term may also refer to other computer systems that use knowledge components. The term knowledge based system and expert system will be used interchangeably in this thesis.

human or electronic" (De Greef, Breuker and De Jong 1988).[5] Collaboration is important because humans and computers have opposite abilities that complement each other. Cooperation aims at man-machine synergy.

Salle and Hunter (1990) consider that most KBS act as a prosthesis rather than support. Due to the subjective and changing nature of many domains such as medicine, they consider the prosthesis approach to be unsuitable. Users need more than prescriptive systems, they need to assess the answer through better explanation and query facilities. Expert Systems have been designed from the machine's viewpoint and lack the scope and robustness needed by users with a wide range of needs. They state "the KBS should be able to respond to a large variety of interactions, not only the commonly supported question" (Salle and Hunter 1990, p.6).

Kidd and Sharpe (1987) go so far as to say that many ES are not useful once in a commercial situation. More understanding of tasks and a theory of cooperative problem solving between man and machine is needed. They argue that current systems do 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 system described in this thesis allows the user to ask a variety of questions by accessing the knowledge in different ways.

Having looked at first generation ES and their various limitations, we take a more up-to-date look at second generation ES and the issues that are relevant in delivering systems that are more user-centred.

1.1.2 The Limitations of Second Generation Expert Systems

To avoid or overcome the problems associated with first generation ES, there has been a general focus on building KBS at the knowledge level (Newell 1982) and interest in generic solutions to KBS development. These approaches to reuse are reviewed in the next chapter. In the next few subsections we consider the issues that are important to the usage and acceptance of KBS from a user perspective. The issues covered concern human-computer interaction, the importance of user control and a discussion of the types of systems users need. Then we look particularly at how a situated cognition view of knowledge affects the system requirements.

---

[5] In this thesis the term activity is used in preference to the word mode due to the possible negative connotations associated with modal systems (Tesler 1981) where for example, the user may become confused whether they are in browse or update mode.

1.1.3 Human Computer Interaction Issues for KBS

Stelzner and Williams (1988) emphasize the importance of the user interface for the acceptance of expert systems. One common interface in KBS is the consultation system, where the user is asked to answer certain questions and is given a recommendation at the end. Clancey (1992) considers the typical consultation mode of ES to be 'superficial' because the line of questioning approach does not allow the user to build a model that they can manipulate. Stelzner and Williams (1988) see a change from the traditional consultation style ES to what they term 'Expert Advisory Systems'. In these systems the human is involved in the loop and therefore it is necessary to support cognitive tasks.

Baroff et al (1988, p.100) consider question-answer style dialogues to be inappropriate for user control, visibility and user initiative. In fields like medicine and electrical engineering, graphical displays, such as visual causal diagrams (VCDs), are used for both the input of data and the display of results. This could be classed as a direct manipulation interface, which Baroff et al (1988) define as interaction via the mouse only, without the use of the keyboard, menus or commands. Direct manipulation reduces the knowledge requirements necessary to use the computer, letting the user concentrate on the task. Baroff et al (1988) argue that such a system provides a better fit with the user’s perception of the domain because they are usually easier to learn, less computer obtrusive and may allow 'what-if' exploration. Such a dynamic "world-of-action" interface provides instant response and update. They also believe that users should be given syntactic and semantic knowledge of computers and semantic knowledge of the domain.

Baroff et al (1988) see graphical user interfaces as the most appropriate for the capture of knowledge because they claim people think in pictures. They also recommend development of the user interface before entering domain rules so they can be validated on entry and the use of procedural control may be necessary to structure the knowledge acquisition process. This idea is exemplified in the work of Zacklad and Fontaine (1993) in their C-Kat system which assists users who are entering knowledge to choose the best conclusion from a range of numerous possibilities. C-Kat is discussed later in more detail in section 3.4.1. A final point made by Baroff et al (1988) is the importance of user involvement in the design of the interface since comprehension of the domain knowledge is dependent on the quality of the user interface.

The aspect of HCI focused on in this thesis is the usefulness of the system rather than its usability. Therefore no particular user interface style is recommended. In fact, this thesis emphasizes the situated nature of ES and that systems must be built to fit the cognitive models of the user while providing the sort of flexibility of interaction with the knowledge base that the user would have if the user was performing that activity without a computer. Due to the relevance of HCI issues for the acceptance of ES, further discussion can be found in section 1.2.1.

1.1.4 The Importance of Control

Since the advent of the PC, users have come of age and want to have control of the application (Pew and Rollins 1975, Schneiderman 1980). Users of experts systems do not want to relinquish decision making to a machine (Ignizio 1991, Langlotz and Shortliffe 1983). In the case where the user is an expert this concept is easily accepted. However various research efforts have shown that novices also wish to be in control of the decision situation. Kidd (1985) studied discussions between experts and novices where it was shown that the novice wanted to be able to propose solutions, evaluate remedies and have the remedy explained. The process was a form of negotiation. Kidd argues that the user approaches the activity with intentions, expectations and constraints. The ES needs to assist the user with exploring the 'remedy space' by a process of evaluation and comparison.

An interesting study by Gunnar (1978) revealed how fear in 12- and 13-month old infants could be alleviated if control of the situation was given and may be one of the reasons why control is so important in gaining acceptance of computer systems. Fear of change or the unknown are identified as causes of non-acceptance of computer systems (Flaaten et al 1989).

Hender and Lewis (1988) consider control to be a key issue in the design of user interfaces for expert systems. Experts want control and want to be able to validate and alter the knowledge. This involves understanding the reasoning process used by the system and the justification for the recommendation. They believe novices are more likely to accept the expertise offered without as much scrutiny but want a better interface to improve access to the knowledge.

1.1.5 The Type of Systems Needed

From the previous sections it can be seen that ES often do not match user requirements. A particular limitation mentioned was the consultation system style. Hender and Lewis (1988) argue that the user interface needs to be tailored to the environment in which it will be used because it will need to fit in with other software interfaces currently being used and may have different data sources. Possible interface alternatives to consultation include "second opinion" system styles which could be something like a critiquing or ‘what-if’ analysis activity. Hender and Lewis see a need for multiple interfaces to be provided with ES.

Stelzner and William (1988) argue that due to the interest in reuse there are new minimal requirements that must be supported by ES. For example, first generation ES were limited to narrow domains, but today with the development of corporate databases the scope of ES is increasing requiring the same knowledge to be used to cover more domains. Stelzner and Williams (1988) echo the thought expressed by many others (Chandrasekaran and Johnson 1993, Silverman 1992a and Swartout and Moore 1993) that deeper models are needed. They see the answer to be model-based reasoning which emphasizes organisation of the knowledge into a well-structured model separated from the inference engine. They see the user interface as the component that must handle the complexity of multiple uses and recommend the possible use of multiple interfaces one for each usage. Due to their emphasis on the importance of the user interface in enabling reuse they suggest that user interface tools need to be developed that may be reused to minimise the effort. This thesis has taken up the challenge to see if knowledge could be reused for different activities with only changes to the user interface required. The complete system developed is described in Chapter Seven.

Stelzner and Williams (1988) see that the difficulty is not in building commercial tools but in matching the way the system behaves related to the user's cognitive tasks. They also argue that different user's will have different needs because their focus is different. A KE is interested in how the knowledge will be represented. The user is interested in the knowledge itself. While Stelzner and Williams have suggested an approach to HCI they admit their approach is still emerging since ES technology is still developing. Below are six features that they consider will be mandatory features available in a KBS:

1. access must be via a natural idiom such as diagrams, charts, networks, text, etc depending on the domain and the user;
2. a format that allows manipulation and experimentation such as spreadsheets;
3. immediate feedback;
4. recoverability which allows the user to play with scenarios without penalty such as loss of current state and actual data;
5. different levels of abstraction and granularity and
6. multiple interfaces for different users and situations such as an intelligent and interactive interface for the user and another interface for the programmer.

The author has previously explored another approach to improving the static nature of ES which was to integrate ES with another technology, Decision Support Systems (DSS), that do provide the features of flexibility and user control. DSS, such as spreadsheets and database packages, were launched in the market place in the same time period as ES, in the 1980s. In contrast to ES, DSS gained widespread acceptance even though these systems are often unreliable due to user bias (Phillipakis 1988). The acceptance of DSS is attributed to the high degree of user control, system flexibility and “user-friendliness” of these systems. “User-friendliness” is seen as a good match between the things the user sees and does with what they feel to be appropriate. In other words interaction is intuitive. This intuitiveness is better described as “transparency of interaction” by Winograd and Flores (1986, p.164).

Many have looked at the development of integrated ES/DSS models and systems (Jones 1986, Turban and Watkins 1986, Kuo 1988, Teng, Mirani and Sinha 1988, Lal 1989, Bannis 1990, Klein and Methlie 1990 and Marvin 1990). These approaches require creating new systems from scratch. Others consider adding a reasoning component to an existing DSS (Leigh and Doherty 1986, Turban and Watkins 1986, Turban 1993). The study by Richards (MAppSc 1994) took a new approach by adding DSS capability to an existing ES knowledge base (KB). The adapted system differed from a traditional ES, or DSS with an ES component, in the user interface, the variables that could be manipulated and the way the knowledge base was processed. Of the six points listed above by Stelzner and Williams (1988) all except for point 5 are supported by the adapted system. The changes made to the system are described in detail in Richards (1994) where a generic approach to adaptation was offered.

The research reported in this thesis continues the work of Richards (1994) and describes a system that offers all of these features including a means of uncovering and displaying different levels of abstraction and a wider range of interfaces. The work in this thesis also builds on the work by Lee and Compton (1995) who speculate that the application of knowledge to a new problem situation will be "one of the most productive approaches to knowledge sharing and reuse" and is a form of "value added transformation".

Many researchers have offered solutions to the problems described above, some of which have already been noted. In a similar vein to many of the above suggestions, Kidd and Sharpe (1987) see proper analysis of the problem as the key to providing KBS that address the users needs. They propose beginning with an initial description of task requirements involving cooperative problem formulation, cooperative generation of alternative solutions and cooperative explanation. The inputs include domain knowledge models, mapping between the model and the machine representation, task specifications, logic descriptions and a framework for performing task and domain analysis. They argue, in addition, if the domain is weak or has no underlying theory then AI will not be able to solve the problem. It can be seen that the solution they outline is exceedingly complex and of limited applicability.

1.2 The Situated Nature of Expert Knowledge and Action[6]

In contrast to the complex approach offered above by Kidd and Sharpe (1987), the solution to the limitations of KBS offered in this thesis is a simple approach which is founded on a completely different premise. RDR were created in the belief that knowledge is not an artifact which only needs to be properly defined in order to be used. Knowledge is not some 'stuff' that can be mined from the heads of experts as was implied by the physical symbols hypothesis (Newell and Simon 1976). Instead the recommendation given by an expert depends on the context in which it is given and does not consist of a description of the expert's thought processes but is a justification of why that recommendation was made (Compton and Jansen 1990).

The emphasis of most ES approaches on building terminological KBS as a prerequisite for the capturing of expertise is not consistent with the way that experts behave: "If we take Heidegger and Maturana seriously, we see that experts do not need to have formalised representations in order to act" (Winograd and Flores 1986, p. 99) and "reflection and abstraction are important phenomena, but are not the basis for everyday action" (Winograd and Flores 1986, p. 97). The act of assigning a conclusion to a pathology report and picking the salient features in the case is an example of what Heidegger (1962) terms *throwness,* which refers to the common human experience of acting in response to being thrown into a situation rather than reflecting first and then acting. Winograd and Flores argue it is the ability to act spontaneously that makes an expert an expert - "The essense of our intelligence is our *throwness* not our reflection" (Winograd and Flores 1986, p.99). From this discussion it appears that most expert action is reflexive. Popper also (1963) implies the impossibility of deduction in an ultimate sense. Since conceptual models are unreliable and difficult to capture an approach which avoids the expert having to articulate their thought processes may be preferrable. Knowledge acquisition, maintenance and inferencing using RDR are simple activities which are provided in reflexive modes that do not require introspection.

Using the first and third person interpretation analogy used by Clancey (1993) the aim is to build a system which performs KA in the first person. Clancey describes the first person as primary learning that is "inventive", requiring reperceiving, and reshaping previous experiences. Primary learning involves automatic recoordination and is a reflexive type of action. An expert viewing a pathology report, assigning a conclusion and picking the salient features in the case to form the rules are viewed as reflexive actions. On the other hand the explanation of why a conclusion and features were

[6] Much of this section is taken from a paper accepted by the International Journal on Human Computer Studies to appear in a Special Issue on "The Challenge of Situated Cognition for Symbolic KBS"

chosen is a reflective action that requires third person interpretation. This is analogous to other situations, such as in social interaction, where models may not be necessary prerequisites but use of them in retrospect is often beneficial for understanding what has occurred (Clancey 1993).

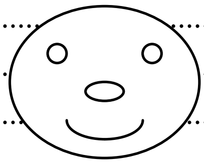
One of the problems with capturing and using expertise is that little is known about how experts arrive at a conclusion. Progress in understanding human cognition has been slow due to: dense information, complexity of the world and the impossibility of observing all the relevant aspects of human cognition (Norman 1993). Situated cognition necessitates interaction with the real world which Vera and Simon (1993) characterise as having real-time involvement; immediate responses to external stimuli and complexity. But the situated view is more than just taking into account interaction between the individual's inner state and the external environment and trying to record all the influencing factors. The problem is that thinking and acting are not two separate activities but thinking is itself an act that modifies further action (Clancey, personal communication). Thus, context in a situated sense is more than just the environment but occurs at a conceptual level that exists within a social setting involving such things as activities, participation, roles, contribution and norms (Clancey, personal communication). Given the complexity of human cognition it may seem inconsistent to advocate simpler techniques. However, by capturing a simple model of the knowledge based on expert behaviour is one way of overcoming or at least avoiding the difficulties associated with the development of high-level models. This does not mean breaking the task down into smaller components in the way that many other approaches have done, such as Generic Task Framework (Chandrasekaran 1986), KADS and CommonKADS (Schreiber, Weilinga and Breuker 1993), Role-Limiting Methods (McDermott 1988) and Components of Expertise and the Componential Methodology (Steels 1993). In the next chapter these approaches and the problems associated with them are considered. First, let us look at the implications of situated cognition for HCI, KBS development, maintenance, verification and validation.

1.2.1 The Implications of Situated Cognition for HCI

The need for complex modeling of the domain before KA can take place has resulted in a preoccupation with knowledge elicitation rather than user requirements (Salle and Hunter 1990). This has resulted in systems designed for knowledge engineers rather than experts or end-users. By simplifying the KA technique and making the user responsible for KA and maintenance we are able to focus more on user requirements elicitation rather than knowledge elicitation.

The ability to capture knowledge for one purpose, such as consultation, and reuse it for other purposes, such as teaching, critiquing or 'what-if' analysis, is more akin to the situated view that knowledge is not a fixed structure that is retrieved and applied from fixed storage positions in memory as many AI programs presume but that knowledge will need to be reperceived and reconstructed for each situation. If we take the view that "perceiving, behaving and learning are one process (Rosenfield 1988) then it makes sense to build a system that does not separate these processes but allows the user to perform KA, find concepts, perform inferencing, learn about and explore the knowledge, and so on, all within the one session. Figure 1.1 shows how the user may choose which activity to perform based on the situation in which they find themselves.

Situation	**Activity**
Develop, Fix KBS	KA, Maintenance
Get advice	Consultation
Evaluate own solution	Critiquing
Consider options	What-if Analysis
Understand advice	Explanation
Learn about domain	Tutoring
Find causes	Causal Modelling
Compare KBS	Requirements Engineering



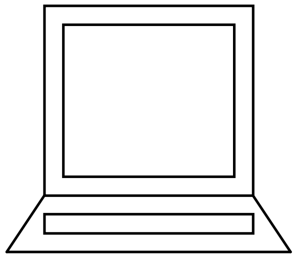


Figure 1.1 The various situations in which a user may find themselves are shown on the left. To solve their current problem they use the features of the system that support the necessary activity on the right. The choice of activity is situation and user-driven. The user can switch between activities as their needs change.

In agreement with Winograd and Flores (1986) and Suchman (1987), human computer interface methods should be reevaluated and changed to support the lessons of situated cognition. This requires understanding the context in which the system is situated (Winograd and Flores 1986). Vera and Simon describe what they term "the soft form of investigation of SA [that] builds AI systems that incorporate principles of representing objects functionally and interacting in a direct and unmediated way" (1993, p. 11). The system described later has taken this form by using an object oriented interface environment and has removed the need for *a priori* analysis of the domain or the use of a knowledge engineer in the KA process.

Not only do we need to study how humans use computers but also their uses and what problems, or "breakdowns" occur (Winograd and Flores 1986). The underutilisation of ES described by Langlotz and Shortliffe (1983) is viewed as a breakdown that requires attention. The following quote describes the features of and alternative to a breakdown:

> *"A system that provides a limited imitation of human facilities will intrude with apparently irregular and incomprehensible breakdowns. On the other hand, we can create tools that are designed to make the maximal use of human perception and understanding without projecting human capabilities onto the computer" (Winograd and Flores 1987, p.137).*

This last point is note worthy. By creating systems that are more usable by humans we are not saying that we are making systems that are more like humans (Winograd and Flores 1987). This is not necessary.

Further to detection, it is necessary to fix or adapt the breakdown. "The ability to resolve breakdowns adaptively is a basic human cognitive capacity" (Vera and Simon, 1993, p.15). The ability of humans to adapt to different situations is considered "perceptive and intelligent" (Clancey 1991b, p.256). Particularly when seeking to reuse knowledge to answer different types of questions, it may be necessary to manipulate the knowledge so that new knowledge can be derived from it. Stroulia and Goel (1994) suggest two approaches to building systems which adapt to the ever changing needs of users and the environment. In the first approach the developer needs to predetermine what the changes may be and try to allow for them in system development. The second approach is to enable "the system to reflect on its own problem solving process" (Stroulia and Goel 1994, p. 395). They have used an adapted version of structure-behaviour-function and Chandraskaran and Johnson's (1993) task-structure framework to design what they term *reflective, self-adaptive problem solvers*. Structure-behaviour-function is used to analyse and describe the problem, which assists in defining the appropriate knowledge and how it is used. Failure driven learning is used to detect when a conflict exists and how to resolve it. Reflection is used to analyse why the problem could not be solved with the plan chosen and to reason about how the plan needs to be modified. The generation of new knowledge from existing knowledge, in the form of new plans from old plans, can be seen as a reasoning task where new knowledge is being derived from existing knowledge.

There is a considerable body of research into adaptive systems. This covers problem-solvers and learning systems, such as Soar (Laird, Newell and Rosenbloom 1987) and Prodigy (Carbonell Knoblock and Minton 1989) and reflective learning systems such as Teiresias (Davis 1977), MetaAqua (Ram and Cox 1994) and Autognostic (Stroulia and Goel 1994). Some of these systems are able to detect a success/failure, assign credit/blame and revise beliefs of/repair the system. While not achieved by all of these systems they all share a common goal, perhaps with the exception of TEIRESIAS, to perform learning independent of human intervention. Since changes in circumstance can not be fully anticipated or described it is difficult to believe that systems can be built which automatically adapt to changing environments. This is why this study has taken the approach of building user-adaptable systems rather than self-adapting systems. The system reported in this thesis learns by helping the user to learn about their domain, reflect on the knowledge and incorporate any new insights into the system. The system offers a wide range of activities and leaves the user to select the tools appropriate for their purpose. With Lave (1988) this thesis would like to move from a "learning transfer" attitude and argues that many approaches to explanation or tutoring do not take the situated nature of learning into account. Allowing the user to decide how to use the system is seen as a better alternative to creating systems that anticipate what the human wants to do, such as systems that include user models (Cawsey 1993, Swartout and Moore 1993).

The simplicity of KA in the system described aims at minimising system breakdowns by providing an intuitive environment. As described later in Chapter Three, the expert is required to look at a case, decide whether the conclusion is correct and if not assign a new conclusion and pick which features in the case warrant the new conclusion. The user is further assisted in their decision making by being shown a list of differences between the current case and the case associated with the rule that incorrectly fired. The system supports direct and intuitive interaction in a wide range of user-selected activities which matches the users desire for system flexibility, system usability and user control.

1.2.2 The Implications of Situated Cognition for System Development, Maintenance, Verification and Validation

Situated cognition offers some exciting as well as some potentially discouraging implications for the development of KBS. Much knowledge is non-verbal and can not be realistically "inventoried" or shared (Clancey 1993, p.109). The immediacy of much human action, the ever-changing nature of contexts, the uniqueness of each act and the complexity of factors that affect an action may lead us to abandon the endeavour to build useful KBS or at least to ignore the need for system verification and validation. This perhaps accounts for the research emphasis on building KBS with little thought for how to maintain these systems once they are in operation (Menzies and Compton 1995, Soloway, Bechant and Jensen 1987). Most approaches to verification and validation support a system development life cycle that assumes the system will be built, verified and validated before it is put into routine use (Kang, Gambetta and Compton 1996). Part of the reason why verification and validation (V&V) approaches focus on pre-implementation is due to the problem that in a typical rule-based system each time a modification is made there may be unwanted side-effects that occur (Grossner et al

1993). It is common in the literature to read of systems that were literally abandoned once the rules became too hard to maintain (Mittal, Bobrow and de Kleer 1988).

A situated view of knowledge places great emphasis on incremental techniques that allow change. RDR have been developed with maintenance viewed as an essential activity and offers online and incremental validation performed by the user.

1.2.3 Summary of the Impact of Situated Cognition

From the discussion above, situated cognition imposes some further requirements on the type of KBS that users need. As noted in 1.1.4, control was an important concern of users which is reinforced from a situated stance since the user will need to be able to adapt their usage of the system to suit their varying circumstances. Taking into account the frame of reference problem (Clancey 1991a) it is also necessary that different users are able to control and adapt the same system according to their different requirements, viewpoints and personal preferences. The need for users to have primary control of the system also extends to the activities of KA, maintenance and validation. In most KBS research, these three activities are the responsibility of the knowledge engineer and the user is a third party. In the RDR approach used in this thesis these activities are performed by the user with no or minimal intervention of a knowledge engineer. The use of contextual knowledge that is grounded in real world examples is another feature that a situated view and RDR support.

Having described the underlying motivation for this thesis and determined the types of KBS that users need we go on to look at the aims of this thesis towards the development of such a system.

1.3 The Aims of this Thesis

It was the goal of this thesis to develop an approach to building KBS that supported the user in a wide range of activities. In keeping with a situated view of human action, the user could determine what activity was appropriate to their current needs. This meant that knowledge that may have been captured for such purposes as inferencing or consultation would need to be reused to support other activities such critiquing or tutoring.

To achieve this a KA technique needed to be found or developed that facilitated capture of knowledge that could be reused in many ways. As is discussed in section 2.3 this involves acquiring the right kind of knowledge, at appropriate levels of abstraction and depth, the context within which the knowledge applies and enough knowledge to cover these different uses. To avoid the limitations of first generation ES and to cope with the evolving nature of expertise, a simple KA method and a reliable maintenance strategy were also highly desirable. To give users control and ownership of the system it was preferable to find a KA technique that could be performed by the user.

It was found that RDR satisfied all of the above criteria except for capturing knowledge at different levels of abstraction. The ability of RDR to support the user in performing KA, maintenance and inferencing, which are referred to as reflexive modes, had already been demonstrated in other RDR research. However, two problems with RDR were noted: the repetition problem and the lack of higher-level modelling. The repetition problem was of concern to reuse because it was expected that a cleaner representation would result in better explanations. Of greater concern was the development of a method for finding the higher-level concepts from the primitive concepts in an RDR rule-base that could be used to support the explanation or reflective modes of interaction. It was the aim of this thesis to address these two limitations.

In addition to these initial goals, it became apparent that the framework that allowed hierarchical conceptual models to be developed from an RDR KBS could be useful for the group activity of comparing KBS and detecting and resolving inevitable conflicts. This added a further goal to be pursued which was to extend the knowledge engineering framework to support requirements engineering (RE). Each of these goals are described further in this thesis together with the work that was conducted in the pursuit of them. In the next section an overview of the thesis structure is given which indicates in what part of this thesis the discussion relating to the various goals can be found.

1.4 The Structure of this Thesis

In this chapter, the need for reuse in general and *activity-reuse* in particular has been considered. In **Chapter Two** the various approaches to KBS reuse are discussed, followed in **Chapter Three** with a description of RDR and related theory. To allow RDR KBS to support *activity-reuse* it was necessary to address two shortcomings of RDR: the repetition of knowledge and the lack of terminological models. This led to the investigation of the use of rough set theory (Pawlak 1991) for the removal of repetition, which is described in **Chapter Four** and the use of formal concept analysis (Wille 1982) for uncovering a T-box from an RDR-initialised A-Box, which is presented in **Chapter Five**. As a further, and significant, outgrowth of the formal concept analysis work is a framework to support the group activity of requirements engineering that is discussed in **Chapter Six**. In that chapter it is shown that RE can be viewed as a maintenance or validation activity over multiple KBS. In **Chapter Seven** we review the different activities and how they are performed in the MCRDR/FCA system developed.

Chapter Eight offers an evaluation of the MCRDR/FCA tool and its general applicability to rule-based systems. The thesis concludes in **Chapter Nine** with a discussion of the contribution of this thesis. We turn now to a review of the knowledge reuse literature.