CORRECTUS IICI

A User-Oriented Critiquing Methodology Supporting Adapted and Optimised Learning

by

Jenny Eriksson Lundström

Thesis for the Degree of Master of Science
Majoring in Computing Science, 10 credit points

Department of Information Science
Computing Science Division
Uppsala University
P.O. Box 513
S-751 20 UPPSALA
Sweden

Abstract

When learning problem-solving or decision-making strategies, a human needs to be active and develop an own mental model of the problem-domain. To optimise the learned knowledge, clear, timely and individually adapted comments on the own findings are needed. In a computerised learning environment, the adaptation to the user is a complex task, often handled by a knowledge base interpreting the user. Thus, the user's own capabilities of guiding the user diagnostic feature are left unemployed.

This thesis investigates a critiquing approach on utilising the user's own capability to support adapted and optimised learning. A user-oriented critiquing methodology is introduced, which through its interaction with the user is able to support the generation of sound, dynamic and individually adapted critique. The utilisation of the critiquing methodology is described by its integration in an existing semi-formal architecture. The provided support of adapted and optimised learning is illustrated through a session with a knowledge-based hypermedia system. As a foundation of the proposed critiquing methodology, a theoretical study of human cognition and learning, artificial intelligence, knowledge representation and presentation as well as interaction paradigms, is presented.

Keywords

Critic, Expert Critiquing Systems, Cooperative Problem Solving, User-System Cooperation, Adapted and Optimised Learning, User-Oriented Critiquing Methodology

To My Family

1 Introduction

In the information-intensive and rapidly changing society of today, quick, innovative and correct decisions in key-areas are crucial for the rise or fall of corporations (Liautaud & Hammond, 2000). Due to the increasing complexity and the vast information flows, the need of systems able to optimise and support their users' learning of problem-solving and decision-making knowledge is more prominent than ever.

In other words, there is a need of an easy and still adequate way of automating the essence of educational interaction. A need of a straightforward, dynamic and user-adapted approach on what Wenger (1987) describes as the purpose of transferring problem-solving knowledge. Namely, "[…] to capture the very knowledge that allows experts to compose an instructional interaction in the first place" (Wenger, 1987 p 5).

1.1 Background

When learning problem-solving or decision-making strategies, a user needs to be active (Anderson, 1993) and receive critique of his/her own findings (Andresen, 1991). Otherwise, the user might accept and learn the system's recommendation without a real understanding of the solution provided (Waern, Hägglund & Rankin, 1988).

Hence, in order to achieve a true transfer of problem-solving knowledge, a key factor of success is the user's development of an own model of the problem-domain and the problem-solving strategy. To support this process, a system needs to clearly present and explain its area of expertise and reasoning knowledge. In addition, the system needs to hold the ability of criticising the user's problem-solving strategy in a timely and appropriate way, and adapt the critique to the specific user (Silverman, 1992 b).

Traditionally, a knowledge base, called a student model, has been used in educational computer systems to adapt the generated critique to the individual user (Harrius, 1993). Not surprisingly, designing a student model is difficult and takes time (Andresen, 1991). The effort in time and mastering this complexity might not be feasible to the commercial system designer. Moreover, the quality of the support generated from the system, and hence the user's problem-solving ability, becomes dependent on how well the student model has been able to interpret its user (Moore & Swartout, 1988).

To find an easy and still adequate way of optimising the transfer of human problem-solving knowledge, the author believes that the user's responsibilities must change. Hence, this thesis elaborates on a user-oriented generation of adaptive and usable problem-solving strategies.

1.2 Aim

In this thesis, the critiquing approach, a cooperative user-system paradigm on supporting the transfer of human problem-solving knowledge, is explored. The focus of the investigation lies with the elaboration on the user's own potential to aid and guide the cooperative generation of adapted and dynamic critique.

1.3 Perspective and presumptions

The early use of expert critiquing systems was mainly to criticise the user's solution (Guerlain et al., 1999). The critiquing approach was seen as "[…] a form of explanation, with the advantage that it focuses the system's analysis around the particular concerns of the user in a very direct and natural way" (Miller, 1984 p 2).

In the second generation expert systems of today, the explanations provided by the system are rich enough to provide the user with information about the domain and the problem-solving in a way preferable for the individual user (Edman & Mayiwar, 2003). Hence, the proposed critiquing methodology is based on the assumption that the critiquing feature can be specialised on providing to decision-support systems and educational knowledge-based systems a way of dynamically supporting the user's learning of the problem-solving structures of the knowledge domain.

Moreover, the view taken in this thesis, originating from Hamfelt (1992), is that the interpretation of the domain theory of a knowledge-based system is incomplete without the user's own interpretation of the domain (Hamfelt, 1992, in Edman & Hamfelt 1997; 1999). This implicates to the author that the user's contribution is to be regarded as essential for the critique generating process as well. Since the system is unable to adapt itself to the individual user's viewpoint in the way human teachers do (Wenger, 1987); the author argues that the user should be more directly involved in the system's interpretation of the user's own answers. Thus, this thesis builds on the assumption that the user-system cooperation approach is necessary as well as diminishes the need of an external student model in expert critiquing systems. Instead, the author suggests that the user interpretation could be handled by the meta-theory interacting with the user as an equal partner of the user-system cooperation.

1.4 Method and technique

The method used in this thesis is a theoretical study of human cognition and learning, artificial intelligence, knowledge representation and presentation as well as interaction paradigms utilised in computer systems. The study identified three important factors when providing user support for the transfer of problem-solving knowledge. These factors are user activity, user-oriented support, and true utilisation of the system's and the user's different strengths in aiding the problem-solving process, i.e. regarding the user and system as equally contributing agents. The factors are considered in the inductive development of a user-oriented critiquing process methodology in support of the user's learning of problem-solving knowledge. Without incorporating a student model, i.e. a knowledge base interpreting the user, the critiquing process methodology should support a sound, dynamic and individually adapted critiquing feature.

The main technique of developing the process methodology is explorative, while the illustration of the applicability of the process methodology is applied through the extension of an existing semi-formal architecture for knowledge-based hypermedia presented by Edman and Hamfelt (1997; 1999).

For the theoretical frame of the thesis, the main method is description.

The main technique of conducting the theoretical research has been literature-, article- and report-search, of the databases and libraries reachable within, or from, Uppsala University. The topics were searched using keywords thought to be relevant for problem-solving support, critiquing and knowledge-based systems. The result of this initial search was evaluated and the acquired information was browsed for relevant references. An extended search was performed based on this first result, also adding keywords related to information on architecture and framework.

1.5 Limitations

When providing a complete critiquing model, a close investigation of several areas such as knowledge acquisition, problem-solving, solution evaluation, user modelling and human-computer interaction aspects are needed (Harrius, 1993). The penetration of these areas is a time consuming and intricate, albeit interesting, task. The thesis does not attempt to cover all these fields in detail.

The focus of this thesis is set on the essence of transferring problem-solving knowledge, i.e. supporting the user's understanding and learning of an optimised way of reasoning from an end-user perspective. Although the author believes that the proposed critiquing methodology could be beneficial to the knowledge acquisition phases, the expert perspective is not directly addressed.

The incorporation of the proposed critiquing methodology into an existing architecture is to be seen as an illustration and first application of the proposed equal user-system critiquing approach. Further evaluation of whether and in which aspects the proposed critiquing methodology actually supports and enhances user learning, is considered out of range of this thesis.

1.6 Structure

The following two chapters, 2 and 3, aim at setting the frames of this thesis through a presentation of the theoretical framework, the areas of elaboration and a summary of previous research on expert critiquing systems. Chapter 4 holds the proposed critiquing methodology. Chapter 5 illustrates the possibility of incorporating a critiquing feature in a knowledge-based environment by describing the extension of a semi-formal meta-theory of an existing system architecture for knowledge-based hypermedia. Additionally, an exemplifying sample session is provided. The thesis is concluded in chapter 6 with an evaluation of the proposed critiquing methodology and the chosen approach of this thesis, together with suggestions on further work.

2 Learning problem-solving knowledge

The chapter presents cognitive implications and educational approaches judged important for the aim of this thesis. Firstly, concepts of learning, expertise and problem-solving are viewed from the constructivistic perspective. These definitions are followed by a brief elaboration on user activity. Reflections on the differences and strengths of different educational approaches and issues on learning problem-solving from educational computer systems conclude the chapter.

The idioms of student and user are used interchangeably according to if the context is discussing human cognition and learning or artificial intelligence.

2.1 The constructivistic approach to learning and expert problem-solving

There have been many attempts to define human learning and explain how humans learn. One of the main paradigms is the constructivistic approach. It has influenced cognitive psychology and had major impact on the design of educational systems (Gobet & Wood, 1999).

The constructivists view cognitive growth as a result of the interaction between the individual and the surrounding world, hence a function of learning (Marton, Dahlgren, Svensson, & Säljö, 1999). Thus, human learning could be defined as the development of the individual (Edman & Mayiwar, 2003).

According to the constructivistic approach, humans learn by iteratively developing and extending mental models of concepts, objects and their relations. The knowledge is stored as rules and overlaying patterns (Steels, 1987). Besides these structures, human experts are thought to hold additional schemata with structures of cues, i.e. keywords used to access and store information (Gobet & Wood, 1999), triggering the use of heuristics, i.e. shortcut rules or surface knowledge, that is based on experience (Durkin, 1994).

An expert is not defined solely by his/her use of heuristics. Among other skills, human experts possess an ability of utilising deep knowledge, i.e. domain theories and problem-solving knowledge, in situations where no heuristics yet have been learned (Steels, 1987). Hence, it is important that a student has learned both deep as well as surface domain-knowledge in order to apply the learned problem-solving strategies efficiently in unfamiliar situations (ibid.).

2.2 User activity and self-explanations

According to a branch of the constructivistic approach, the student could benefit from self-explanations (Gobet & Wood, 1999). The student's own activity in creating the self-explanations is said to help increase the number of retrieval cues of the problem-solving knowledge. The activity is also preferable since any conflicts and failures in the student's problem-solving easily emerge and thus, can be corrected.

To prevent the self-explanations from resulting in the student's learning of irrelevant or misleading knowledge, it is important that the moment when the student generates his/her self-explanation is optimised (ibid.). Additionally, different individuals have different preferences, i.e. learning styles, when it comes to learning (Gardner, 1983 in Edman & Mayiwar, 2003). Thus, to provide all students with the opportunity to learn in accordance with their abilities, the education needs to be individualised to an extent beyond a traditional classroom situation (ibid.). Hence, the capability of computer systems to provide enriched learning environments has attracted education to technology (Kraus, Reed, Fitzgerald, 2001).

2.3 Learning problem-solving from educational computer environments

Throughout the history of Artificial Intelligence, researchers have attempted to automate the transfer of human problem-solving knowledge by building knowledge-based systems that emulates human experts and their reasoning (e g. Swartout, Shortliffe, discussed in Wenger, 1987). Early use of knowledge-based systems in educational settings (Clancey, 1979; 1983), establishes the knowledge-based system as a potentially excellent educational tool, thanks to its declarative approach, and its need as well as its ability to explain its own reasoning (Wenger, 1987).

Traditionally, the knowledge-based systems aimed at presenting the user with an optimal solution, leaving to the user to criticise the system and its way of reasoning (Fischer, Lemke & Mastaglio, 1990; Guerlain, Smith, Obradovich, Rudmann, Strohm, Smith, Svirbely & Sachs, 1999). Still, only a very active and motivated user could through repeated why- and how-explanations of why the system needs a particular value or how the actual solution was inferred, perceive the domain structure and be able to judge the system's solution (Andresen, 1991).

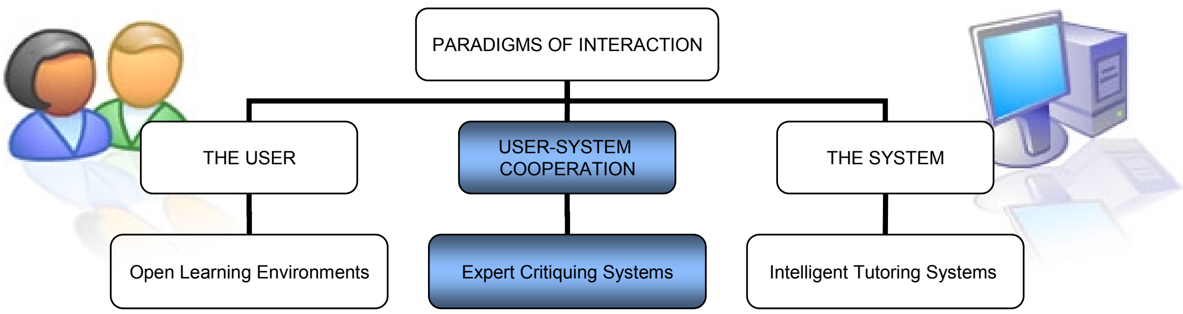

Figure 1. Interaction paradigms.

To enable the user to gain the necessary knowledge on how the system had inferred the solution from the domain, open learning environments (see figure 1, the user) were developed. In these computerised, open "real world" environments, the students hold full control and hence, are able to identify, manage, and present the information that they have accumulated and assembled according to their own preferences (Means, 1994 in Kraus et al., 2001). Here, research on letting the user create his/her own knowledge-based system has been attempted with positive results (Andresen, 1991).

Albeit, in order to facilitate the achievement of a rich understanding of the problem-domain a human needs confirmation on his/her findings when learning problem-solving strategies (ibid.). Therefore, relevant feedback, as given from the tutor to the student in a tutored learning situation, was lacking in the open learning environments.

The intelligent tutoring systems (ITS) addressed this shortcoming by taking control of the user's learning process, and adapting the training to the system's interpretation of the user's need (see figure 1, the system). The educational environments of intelligent tutoring systems try to model and replace a human tutor (Andresen, 1991). The system decides the sequence and type of problems to be solved by the student. It judges the user's answers, determines the user's progress and controls the definition of the decisions and the diagnoses. The students are active in the sense that they propose solutions to problems, but the system holds the overall control of the learning process (ibid.).

In addition to the restricted user control, another issue with the intelligent tutoring systems have been that the human learning process is very complex and not yet fully understood. Good human teachers seem to "[..] have an unmatched ability to perceive the students view and to adapt their behaviour accordingly" (Wenger, 1987 p 16)[1]. Due to the complexity of modelling the human teacher in a learning situation, the intelligent tutoring systems have not really been able to leave the research laboratories in any larger aspect (Andresen, 1991).

Conclusively, both open learning environments and intelligent tutoring systems hold attractive as well as less attractive features when it comes to transferring problem-solving knowledge. Thus, the potential for an interaction paradigm utilising the strengths from both approaches seems large. A cooperative problem-solving approach, which incorporates many of these features, is discussed in the following chapter.

---

[1] Ulrika Leimar, a junior-level teacher, describes this ability when explaining how she teaches young children reading- and writing-skills "…läraren (måste) försöka leva sig in i hur det enskilda barnet använder språket, dvs. hur det *tänker*". Eng: "…the teacher (must) try to adapt to the specific child's own use of his/her language, in other words, adapt to the child's way of thinking." (interview in Lärartidningen 1976-03-01, cited in Marton et. al 1999 p. 164, translation by the author).

3 The critiquing approach - a cooperative user-system paradigm

In a cooperative environment, the user and the system participates together in the generation of the resulting product (see figure 1, user-system cooperation). The initiative in the dialogue, i.e. the control of the progressing work, is shared between the user and the system. This mixed-initiative approach allows both parties to contribute according to their strengths and knowledge of the goals and domain (Fischer et al., 1990). Together, the system and its user “[...] incrementally evolve an understanding of the task“(Fischer, Lemke, Mastaglio & Morch, 1991 p 125). To enable a constructive and focused interaction between the system and its user, a critic is an important component of a cooperative problem-solving system.

In the subsequent sections, definitions of critics and a brief presentation of some critiquing implementations are provided together with guidelines on areas of use. Next, presentation considerations and issues on representing critique are viewed, followed by an elaboration on the processes of expert critiquing systems. Finally, issues concerning the utilisation of the critiquing approach in supporting the user’s learning and optimisation of problem-solving knowledge are reviewed.

3.1 Critics, expert critiquing systems and usability issues

In this section, an introduction to expert critiquing systems is provided. It is followed by a presentation of some previous critiquing studies. The section is concluded by a paragraph on when to use critics.

3.1.1 Expert critics

A critic is an agent, human or system, which holds the capability of analysing the student’s solution and is able to provide the student with a reasoned opinion on the student’s choices and strategies (Fischer et al., 1990).

Critics do not intend to deliver learning, but merely to support the student during a learning situation (Andresen, 1991). The main aim of critics is to recognise deficiencies like errors and suboptimal choices in the student’s solution and communicate these findings to the student (Fischer et al., 1990). Thus, the student is not only allowed but also required to develop an own line of reasoning and a subsequent solution.

Despite the required user-activity, the student is not alone during the problem-solving. The critic quietly monitors the student’s work until the student’s strategies or solutions could be improved. Hence, in contradiction to passive help systems, the critic does not require the student to formulate a question to initiate the critiquing dialogue.

3.1.2 Automating critics

An automated critic can be defined as a critiquing module, i.e a set of axioms, which interacts with an application providing support to the user's problem-solving, i.e. a decision support system or educational environment. Hence, an expert critiquing system (ECS) could be seen as an expert system, i.e. a system with knowledge in a domain and knowledge on how to evaluate the user's problem-solving strategy in this domain, which is extended with a critiquing module.

The automation of critics was approached by Miller (1984) as a mean of supporting physicians in managing anaesthetic therapy. Miller felt that the critiquing approach brought advice to the physician in a form he/she could most practically use. Since the system would not compete with or threaten the physician's own diagnosis in the very subjective domain of anaesthetic therapy, the system's advices would also be more acceptable to the physician (ibid.). Miller (1984) argued that the user-system interaction was more like a human tutoring or advising situation, where the physician gathered the necessary data, placed an own diagnosis and only then asked a colleague or tutor to evaluate his findings.

According to Guerlain et al. (1999), an automated critic must have a model of *expert performance,* a model of the *types of errors to detect, means to detect the errors,* and *means to notify the user of a detected problem*.

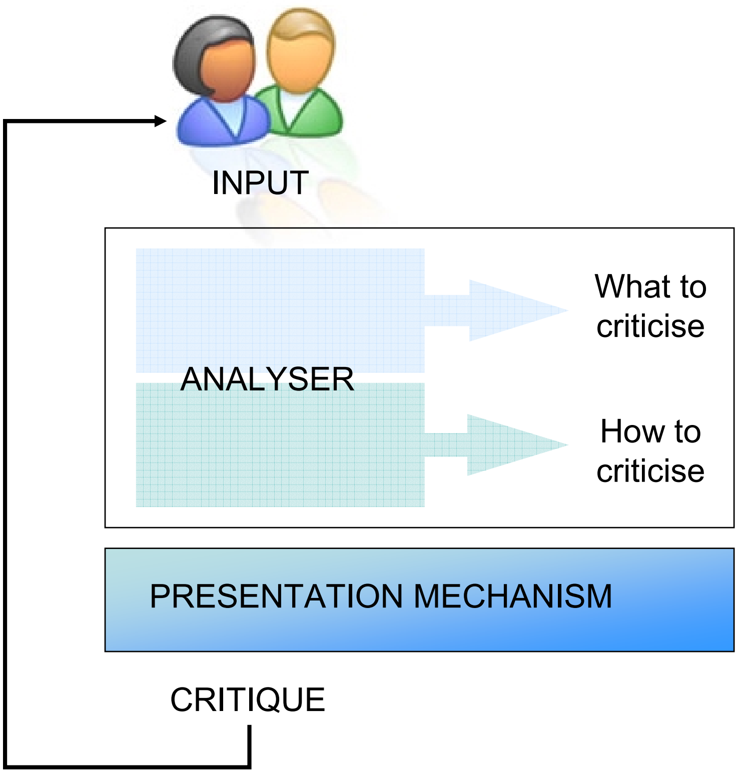

Figure 2. The critiquing module.

Hence, a critiquing module consists of two main components: an *analyser,* which determines what and how to criticise the user's solution through comparing the user's input with the expert and error models of the system; and a *presentation mechanism*, which holds the responsibility of presenting the critique to the user (see figure 2). These components give the automated critic the ability to analyse and generate critique of the user's problem-solving strategies and proposed solution.

As opposed to a traditional expert system, which generates an own solution to the problem at hand, the starting point of the expert critiquing system is the user's problem-solving strategy and solution. Hence, the system's output is the critique, discussing the pros and cons of the user's strategies in a way adapted to the user's thinking and proposed solution (Miller, 1984). Additionally, most implementations of critics make suggestions on how to correct and improve the strategies, allowing the user to refine the solution in an iterative manner.

3.1.3 Critiquing studies

Several attempts of developing critiquing systems or knowledge-based systems with a critiquing ability have been made. This presentation attempts to provide an overview of critiquing studies of interest to the aim of this thesis.

Diagnosis systems

ONCOCIN

The ONCOCIN-system created by Langlotz and Shortliffe (1983) was initially a diagnostic expert system intended for aiding physicians in the treatment of cancer patients. ONCOCIN provided suggestions to the user on the chemotherapeutic management plan. Albeit, the users complained that they had to change or override the system whenever they disagreed even slightly with the systems solution (Langlotz & Shortliffe, 1983). To address these shortcomings of the system, a critiquing feature was introduced. The critic was quietly monitoring the user during a session, solving the case in parallel with the user. When the system discovered a significant difference between its own solution and the user's, the user was presented with the system's solution and a justifying explanation. The level of disagreement was handled by a method called hierarchical plan analyser.

Langlotz and Shortliffe thought that the critiquing approach, i.e. to let the user's solution be criticised by the system, instead of the other way around, would be a more natural and less offensive way of providing advice (ibid.). However, the results of this altered approach are unreported (Guerlain, 1999).

ATTENDING

The first implementation directly aimed at the critiquing approach was the ATTENDING system by Miller (1984). It was a computer system critiquing physician's management plans for anaesthetic therapy. The ATTENDING system attempted to provide feedback to the physician to help him evaluate and refine his own solutions.

Miller elaborated on the critiquing approach, implementing a whole family of ATTENDING systems. The systems held critiquing features for hypertension, ventilator management, and pheochromocytoma analysis using the same conceptual model as the first ATTENDING system. Finally, a general system-shell called E-Attending was presented.

Miller's research on the ATTENDING systems has contributed greatly in identifying guidelines of developing critiquing systems further utilised in following implementations (Silverman, 1992 b).

Design-oriented applications

LISP-CRITIC

The LISP-CRITIC by Fischer (1991) supported programmers to improve their LISP code in either a more cognitively appealing or more machine efficient way, i.e. making the code either easier to read or the programs smaller or faster to execute.

The LISP-CRITIC was a passive critic, waiting for its user to invoke its analysis. Moreover, the critic held an explanation feature, which presented explanations to the user in an incremental fashion. Due to the breadth of possible application-fields of the code, the LISP-CRITIC required that the user confirmed the suggestions provided. Thus, it was implemented into a semi-formal architecture. The system incorporated a user-modelling component to customise the critique to the individual user. This user-model acquired and maintained information of the user's goals.

JANUS

JANUS was a design critic-module for architectural design of kitchen interiors. It was developed by Fischer, Lemke, Mastaglio and Morch (1990) and aimed at being integrated in design environments, assisting its user in the design of suitable kitchens.

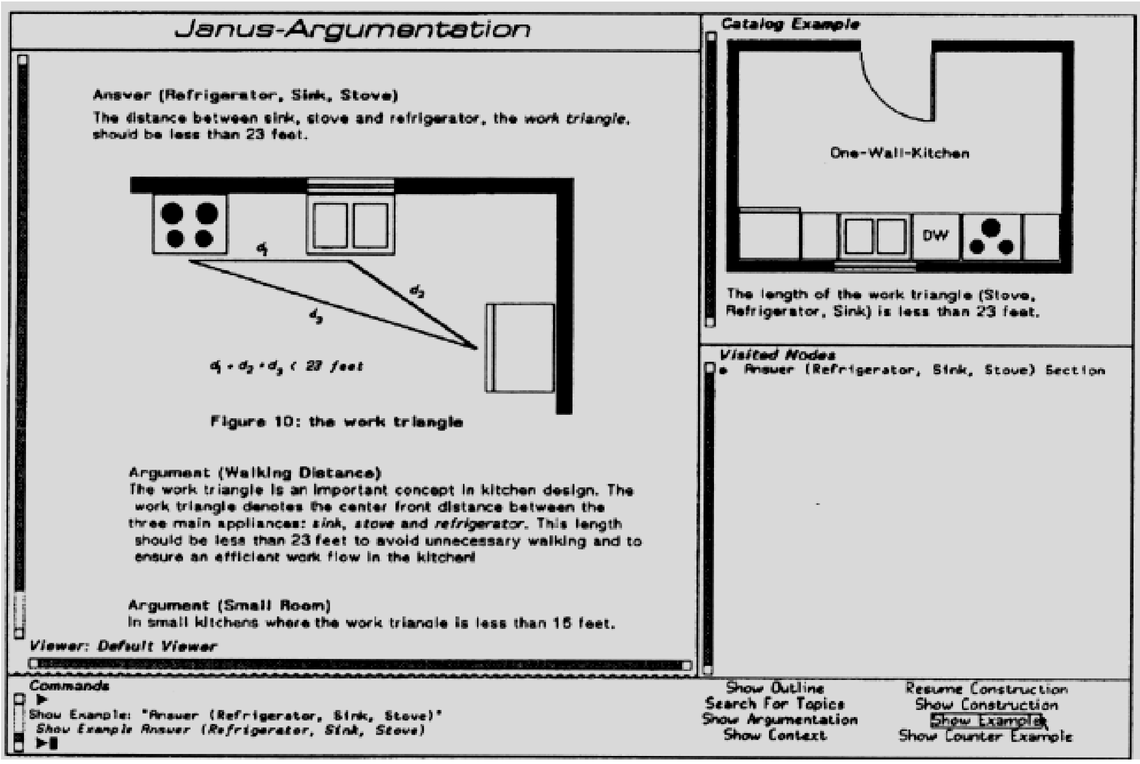

Figure 3. The JANUS Argumentation – a hypertext system used to explain critique and the concepts of the design environment of JANUS (Source: Fischer et al. 1990 p 340, ©1991 ACM).

The critiquing module contained several different critics. Each contained a separate piece of knowledge on how to distinguish between "good" and "poor" kitchen designs. This knowledge was implemented as condition-action rules. These rules were checked when the user altered the current design. When an inferior design decision was discovered, critique was generated and displayed to the user. The user could also activate an additional explanation (see figure 3).

Fischer and his associates made several implementations of embedded design-oriented critics. The researchers argued that the cooperative approach utilising the synergy of different system-components was important to "empower the user" in his/her activities.

Educational critics

WEST

WEST system by Burton and Brown (1982) was an educational system in arithmetic. It pioneered fundamental ideas that the critiquing paradigm incorporates (Fischer et al., 1990). WEST provides a bridge between open learning environments and tutoring since it provides a guided discovery learning with adapted user advice. The adaptation of the critique was made through the interaction with a user model (Fischer et al., 1991).

STEAMER

STEAMER/Feedback Mini-Lab by Forbus (1984) was an educational environment for simulating steam plant controllers. Like ONCOCIN, it was augmented with a critiquing feature, and thus not initially incorporating an expert critiquing feature. The critique was provided after the user has constructed a device.

Decision-support critics

TIME and the COPE Environment

Silverman (1992 b) conducted a study comparing alternative designs for critiquing systems which aimed at teaching students probability theory. The first system used *debiasers*, i.e. provided critique after the user had delivered his/her conclusion. Explanations were displayed to inform the user of deficiencies in his/her reasoning. If the user pursued the incorrect reasoning, explanations with arguments from three different levels where displayed. The second system used both debiasers as well as *influencers*, i.e. guiding explanations displayed beforehand, which aimed at foreseeing common mistakes. Thus, the user received influences before answering the question at hand.

Silverman proposed that to be effective a critiquing system needed both debiasers as well as influencers. Additionally, a director-feature, which demonstrated the strategies to the user, was said to be needed. The critiquing should be directed according to triggers maintained in a library module holding the most common errors for a given problem and different levels of critique for these errors.

Silverman wanted to merge the areas of decision making and artificial intelligence in his critiquing approach. With this intent, he elaborated on and incorporated error models of cognitive psychology and decision support systems.

AIDA

A prototype critiquing system called AIDA was developed by Guerlain, Smith, Obradovich, Rudmann, Strohm, Smith, Svirbely & Sachs (1999). It aimed at assisting blood bankers in identifying alloantibodies in patients' blood. An unobtrusive error-monitoring of the user's intermediate and final conclusions was implemented. The system monitored its users for errors resulting from using incorrect knowledge and leaving out or missing to use knowledge**,** failure to follow a complete protocol, answers inconsistent to the users own previously input data and finally, answers inconsistent with prior probability information in the domain.

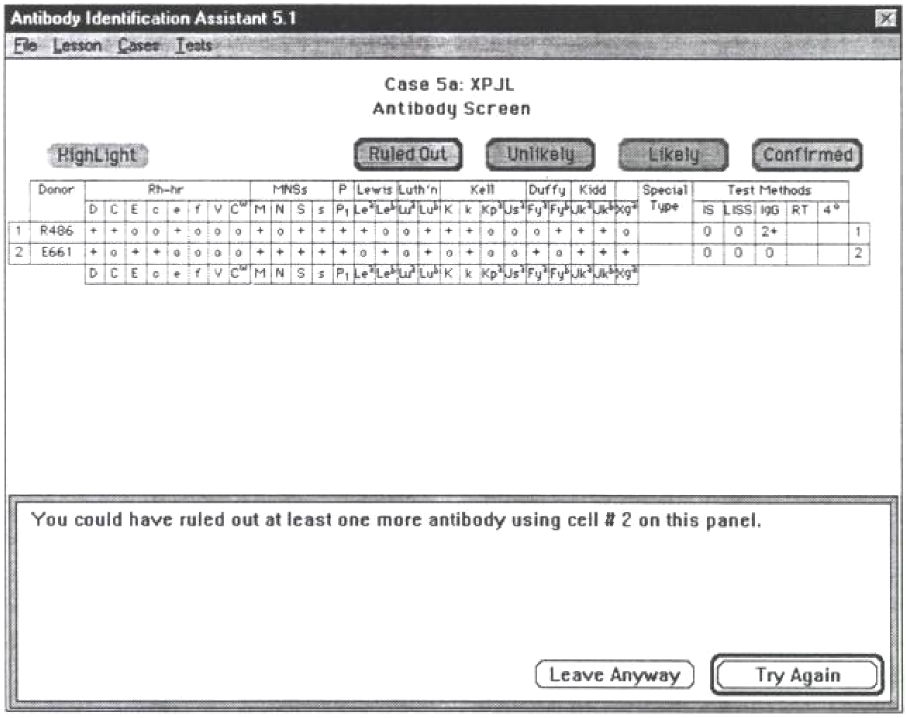

Figure 4. Sample error message in AIDA – Antibody Identification Assistant (Source: Guerlain et al., 1999 p 79, ©1999 Human Factors and Ergonomics Society)

The critique could be displayed to the user or merely log the interaction for evaluation purposes (see figure 4).

The system and its critiquing approach were evaluated during tests. Guerlain and her colleagues found that the test group using the critiquing system had significantly better performance than the control group. Noteworthy was that this was also true for brittle test-cases, i.e. problems not sufficiently covered by the system's expertise. Guerlain and her colleagues claimed that this empirical evaluation demonstrated that the critiquing approach was suitable as decision support for routine work in complex domains, and also, unlike traditional expert systems could be of significant value in situations in the periphery of the system's main knowledge domain.

3.1.4 When to use critics

Critique is usable as a form of explanation when a more traditional explanation of the domain or current problem is too general or the explanation specifically needs to address details in the user's line of reasoning. The critic thus can handle situations where the user's solution is very poor, since it is able to answer questions of "why not" instead of "why" a condition or specific line of reasoning is to be preferred (Miller, 1984).

According to previous research, the critiquing approach is best suited for small and constrained domains with multiple solutions; where each solution requires complex decision processes and where subjective judgement must be frequently applied (ibid.). Hence, the critiquing feature is suitable as decision support for such complex decision processes that hold lots of knowledge on risks, benefits, side effects and costs, since it is under these conditions human practitioners tend to make errors if unaided (Silverman, 1992 b; Guerlain et al., 1999).

Moreover, expert critiquing systems are said to be preferable for domains where legal or ethical responsibility could not be placed on a system as well as situations where incorrect decisions might be dangerous or very costly (Miller, 1984; Puppe, 1999). In such domains, e.g. medical diagnosis, the use of a traditional expert system is limited due to the importance of leaving the responsibility of the final decisions with the physician and not the system (Miller, 1984). Thus, the user needs full knowledge to judge a solution provided by the system (ibid.).

In complex areas, like anaesthesiology or other diagnosis applications, the system needs to incorporate an own problem-solving capability to be able to generate the critique (Puppe, 1999). Albeit, the expert critiquing system's knowledge of the problem domain does not always have to be complete to be able to provide relevant critique. This makes the critiquing approach also suitable for vast domains, domains where multiple solutions exist, e.g. design, as well as domains not yet sufficiently understood, e.g. human cognition (Fischer et al., 1990).

3.2 Presentation of critique

Regardless of the domain features, how well a critic supports its user's learning and problem-solving is dependent on how the critique is presented to the user. As earlier stated, a critic must have means to notify the user of a detected problem. The interaction with the user has been regarded as a problem (Miller, 1984; Guerlain et al., 1999). It is necessary for the user interface to provide the system with sufficient data to detect errors in the user's problem-solving, create critique as well as enable the system to perform its normal work. At the same time, the user interface must be usable and explaining to the user (Guerlain et al., 1999).

The section will provide a brief elaboration on the presentation of critique. The first paragraph holds different presentation techniques. The second paragraph presents demands on the presentation of the critique.

3.2.1 Presentation techniques

Critique can be presented as text, sounds, diagrams, films, a combination of several media etc. (Harrius, 1993). In several critiquing systems, the presentation is text-based and utilises pre-stored canned text strings, template filling using variables or code transformation as means to customise the text presentation.

Other attempts rely on graphical presentations or combinations of text and graphics in graphical user interfaces. For instance, AIDA (Guerlain et al., 1999) utilises the form metaphor for its interactions (see figure 4). The design resembles the paper variant utilised when the identification of alloantibodies in patients' blood is performed manually. JANUS (Fischer et al., 1990) partly presents its critique through drawings of "good" design examples (see figure 3).

3.2.2 Demands on the presentation of critique

The main issue of critique presentation is the user's perception of the presented critique. It is important that the critique is presented in a way which helps the user *assimilate* to the critique (Harrius, 1993). Thus, the critique must be relevant to the user's needs. The user must also be able to *understand* the critique presentation. Hence, the presentation should be given in a form, which is easily grasped by the user. This implicates that the presentation must be adapted to the user as well as be pedagogically presented. Finally, the presentation must provide the means to *explain and justify* the critique to the user (ibid).

Help the user assimilate to the critique

Normally, a human critic or tutor presents his/her analysis in an interactive manner (Moore & Swartout, 1988; Cawsey, 1993). This allows the human critic or tutor to adapt the critique to the user during the presentation. Thus, the focus of the critique can be placed on sorting out misunderstandings or penetrating areas, which the student finds difficult. In other words, interactive presentations maximise the student's understanding of the critique by focusing the critique on the user's current need and adjusting the critique and its explanations accordingly.

Unfortunately, not many systems incorporate this approach for providing interactive critiquing or explanations. Instead, they aim at providing a "complete" batch-presentation, ignoring the user's feedback as guide to the presentation (Moore & Swartout, 1988).

Help the user to understand the critique

According to Moore and Swartout (1988), in many situations the student does not fully understand the tutor's initial explanation. Hence, the student repeatedly asks follow up questions to clarify or elaborate on the tutor's explanation. Often, the student is unable to formulate a real question. He/she merely states, "I don't understand" or repeats the tutor's last words. This also supports that the critique needs to be presented in an interactive fashion and preferably in a way and form easy to reach and comprehend.

Moreover, to help the user understand the critique, it should be presented in its most natural form, i.e. as text, sound, images etc. (Edman, 2001). Hence, the presentation component of the system, i.e. the user interface needs to support the different media.

In order to make the dialogue with the user more fluent and dynamic, coherent multi-sentence explanations emanating from natural language generation have also been utilised in critiquing implementations (Harrius, 1993). Although an attractive form of interaction and presentation, the utilisation of comprehensive natural languages is complex and so far only partly understood. Therefore, the presentation should preferably be made through visual metaphors and direct manipulation of the work item (Silverman, 1992 b).

Provide access to knowledge explaining and justifying the critique

When we learn, it is not always the strategies themselves that is needed as part of an explanation. Instead, we normally need explanations as justifiers, complementary context and term descriptions (Swartout & Moore, 1988; Edman, 2001). These three categories of explanations hold information, which justifies the provided critique to the user, provides a context to the critique and explains the meaning of the used terms. It allows the user to create additional cues, which strengthens the user's mental model of the problem-domain and hence optimises the understanding of the problem-solving strategies.

To be useful, these explanations must be easily available to the user, and be available within a time when they can make a difference. Moreover, they need to be clear in their content to avoid the user from being disrupted from his/her primary work.

Since the chosen means of presentation only can display information available to the system, the knowledge representation is important when it comes to the quality of the generated critique. The modelling of knowledge is the first step when building any system (Newell, 1982). Therefore, the knowledge, the knowledge representation as well as the processes handling the represented knowledge must be considered when developing a satisfactory presentation component.

3.3 Representing critique

The knowledge representation of an expert critiquing system needs to hold knowledge modelling the *expert problem-solving structures* as well as a model of the *types of errors to detect*. In this section, different knowledge representation techniques are presented together with a short elaboration on the knowledge models supporting the critiquing module.

3.3.1 Knowledge representation forms

Knowledge can be represented in various forms. In expert systems, the most common representation forms are frames, semantic networks, logic, rules and object-attribute-value triplets (Durkin, 1994). The various implementations of expert critiquing systems have utilised these representation forms, but often augmented with other structures because of the analysis and presentation aspects.

The ATTENDING-system held no distinct knowledge base, instead the knowledge was represented in the inference rules, frames and augmented transition networks (ATN). These augmented transition networks originate from the natural language representation field (Miller, 1984). Other systems have utilised decision-nets (Silverman, 1992 a). Additionally, extended rhetorical structure theory-graphs (RST-graphs) have been used to enable the generation of multi-sentenced critique (Harrius, 1993). This representation form, like the augmented transition networks, is borrowed from the area of natural language generation.

Still, the use of these augmented structures results in that any existing knowledge bases need to be re-formalised before use. Often, the utilised representation forms originate from prototypical representations or research test beds and thus, are not commonly known to the commercial system developer. This places high demands on the development and maintenance of the critiquing feature. Hence, the expert model needs a suitable and transparent representation, which holds the knowledge in a way not constraining the user's actual work, the critique generation or the user's accessibility of explanations of the problem domain.

3.3.2 Model of expert performance

Since expert systems tries to emulate the reasoning of a human expert, the similarities between how to construct adequate critique on the user's problem-solving strategies and explaining the reasoning in expert systems are large (Harrius, 1993).

Problem-solving knowledge can be defined as plans that express how a certain task can be accomplished. In other words, problem-solving knowledge contains which goals a particular plan achieves and the strategies needed to accomplish that goal (Swartout & Smoliar, 1987).

When modelling the expert's problem-solving, it is important for the quality of the generated critique to avoid an insufficient knowledge representation. It could allow the system to mimic a human expert, without being one (ibid.). Hence, a knowledge representation supporting the critiquing approach should hold the deep as well as surface domain knowledge depicted in various levels and perspectives (Steels, 1987; 1990; Marton, 1999). In other words, to be a real model of an expert, able to explain and justify the created critique, the system needs to have access to knowledge of that domains terminology, the domain facts and problem-solving methods, seen from different levels and perspectives of the domain.

3.3.3 Error model

During problem-solving a student could use poor strategies, hence resulting in poor or non-optimal solutions. This could be the result from using incorrect strategies, slips or inefficient problem-solving techniques by examining too much information (Guerlain et al., 1999).

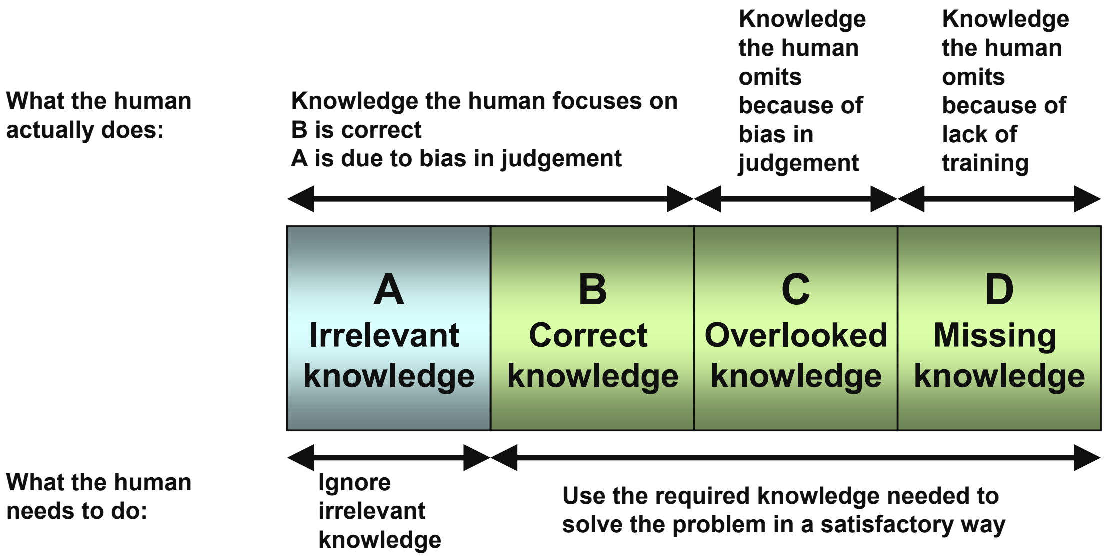

Figure 5. Categorising a human expert's knowledge. The model shows the differential between the human's actual strategy and the needed strategy to accomplish the task at hand. A, B, C, D are to be seen as logic variables, and hence holders for semantic information and cues. (Source: Silverman, 1992 b, adapted by the author).

Silverman (1992 a) investigated human expert knowledge and proposed a model of errors (see figure 5).

To support the user, the critic should notify the user if the user's solution consists of irrelevant knowledge (figure 5 [A]) and explain why not to use it. The critique should also present information on overlooked knowledge (figure 5 [C]) in a way, which shows the user how this knowledge is related to the problem. Finally, the critic should make the user aware of missing knowledge (figure 5 [D]). With the correct content of the critique, the user is said to move his focus from irrelevant knowledge [A] and correct knowledge [B], to the union of the correct knowledge [B] and previously overlooked knowledge [C]; and stepwise merge the missing knowledge [D] into the user's own set of correct knowledge [B]. Hence, the correct critique could aid the user in developing an own mental model, which provides an optimal understanding of the problem (Silverman, 1992 a).

Moreover, the critique generated should be sound, i.e. be coherent with the expert knowledge kept in the domain. Hence, the construction of adequate and sound critique can be complex and involves several system-components. This stresses the importance of using a well-structured process when generating the critique. Thus, high demands are placed on the process utilising the knowledge presentation and representation components.

3.4 The critiquing process

According to Fischer and his associates, the sub-processes of critiquing are: *applying a critiquing strategy, goal acquisition, product analysis, adaptation, explanation* and *advice* *giving* (Fischer et al., 1990). In an implementation of the critiquing approach, not all these processes must be present to generate critique. Still, preferably these different aspects should be considered in a well-structured critiquing methodology. Therefore, this section holds a presentation and an elaboration on the critiquing strategies and the different sub processes of critiquing.

3.4.1 Critiquing strategies

Often, a user model and a set of critiquing strategies control the selection and presentation of what to criticise, and when and how the critique is to be generated and presented (Fischer et al., 1990). The main decision regarding what is to be considered an appropriate strategy could be determined from the goal of the critiquing feature, i.e. *the focus of the critic*, and the context of its use. The strategies should be modelled to make the user feel comfortable with the critic's interventions. The individual user should welcome the critique. Hence, the user's perception of the *intrusiveness* of the system, i.e. how disturbing the critiquing process is to the user's primary task, as well as the *intervention strategies*, i.e. when and how to interrupt the user's work, need to be considered.

The user model

In order to customise the dialogue and the generated critique, the user's previous knowledge and hence, the level and amount of arguments to use, can be established by a user model (Silverman, 1992 b). A simple version of a user model is to divide the user-group into several static classes depending on the user's perceived knowledge. Other, more dynamic approaches utilises user models, which are similar to the student models utilised in natural language generation systems and intelligent tutoring systems (ibid.). These user models need to be dynamic and persistent, changeable as well as transparent to the user for modification and inspection purposes. Hence, the modelling can be complex and has shown difficulties in providing a true interpretation of the user at all situations (Moore & Swartout, 1988).

The focus of the critic

The system needs strategies, which express the general goal of the critic. Depending on if the goal is to help the user in producing a better product, i.e. being a *performance critic*, or if the goal is to support learning, i.e. the critic is an *educational critic*, the strategies differ (Fischer et al., 1990).

The goal of the *performance critic* is to help its user to create a high-quality product, within the least amount of time and using as few resources as possible. Therefore, most performance critics base their strategy on critiquing the product as a whole. In some cases, the user can decide which aspects the system should focus on in its analysis (ibid.). E.g. in the LISP-CRITIC the user decides whether the critique should address cognitive preferences or machine efficiency (Fischer et al., 1991).

The prime objective of an *educational critic* is to support learning. The critic should concentrate on improving the user's future performance. Thus, the strategy focuses on maximising the information retention and motivation of its user (Fischer et al., 1990).

Most critics, both performance and educational critics, perform their work in the negative mode, basing their presentation on deficiencies in the user's solutions (ibid.). These deficiencies can be said to correspond to the categories [A], [C] and [D] of Silverman's error model (see figure 5). A positive critic also informs its user of good choices, corresponding to the category [B] in Silverman's error model (see figure 5). This enables the user to retain good aspects of his/her strategy and aids learning.

The intrusiveness of the critic

The presentation of the critique could be made *intrusive* or *unobtrusive*. In the first case, the user would be prevented to continue if the critique is not acted on. *Unobtrusive critique* could be displayed in a separate window, allowing the user to check the critique when he/she feels it to be necessary. Still, the latter situation could result in the user simply forgetting to check this window, and hence, receive the critique too late. In graphical user interfaces, advertising, i.e. using visual effects to draw the user's attention to the criticised area, has been utilised to address the problem of an inattentive user (Fischer et al., 1991).

Intervention strategies and timing of the critique

There are two main approaches to intervention, *active* and *passive* critics. The *active critics* provide the critique at a time they find appropriate according to the strategy applied. To be able to do this, the critics are constantly activated and are monitoring their user. The *passive critic*, on the other hand is invoked by the user when the user him/herself finds it appropriate to generate and receive critique. Passive critics usually evaluate the current outcome of the user's problem-solving, not the intermediate steps that resulted in the outcome.

The *time* to invoke critique is an intricate decision. Silverman (1992 a) suggests that different erroneous behaviours could be avoided depending on the correct selection of when to display the critique. The critique could be displayed *before, during* or *after* the user's work.

Providing preventive critique, i.e. *before* the user's action, will probably help the user avoid committing errors. This could be done by informing the user of common errors; or make the different consequences of the available alternatives transparent to the user. Still, if presenting too much or irrelevant information, the user could accept the system's advice without creating an own understanding of the problem. If the information is irrelevant or not adapted to the user, the user could start disregarding the critic's advice. If the critique is displayed *during* or immediately after the incorrect action has been taken, the user has the problem and context active and remembers the decision. The disadvantage of immediate critique is that the user's thinking process could be disrupted causing short-term memory loss (Fischer et al., 1990); or that the user creates local solutions to the critique and not addresses the whole problem (Silverman, 1992 a). On the other hand, providing the critique *after* the user finished his actions, i.e. to delay the intervention, could instead make the critique come too late, feel out of context, and become a greater hindrance of the user's progress than necessary. According to Silverman (1992 a), it could also make the user less open to the critique, defending his/her own solution to the problem (ibid.).

3.4.2 Goal acquisition

To be able to criticise the user's problem-solving, the system must hold or acquire some knowledge of the problem domain. Depending on the knowledge available to the system, the critic could reason on static characteristics or relations of the domain objects, or provide a more elaborate analysis of the user's solution.

To enable the more elaborate critique to be generated, the knowledge must provide some understanding of the user's specific goals and the situation in which the knowledge is used (Fischer et al., 1990). The knowledge could be *inferred from the system's own knowledge base* or be established by *asking the user*.

In the first case, the understanding of the user's goal can be acquired in two ways: Firstly, by the *implicit goal acquisition approach*, where a general goal is built into the system, e.g. as a desire to set a "correct" market value on a real estate. Secondly, the goal is acquired by *goal recognition*, where the system observes the user's actions or the product this far. For example, by providing information on the qualitative assets of the real estate, like wooden floors, sunlit balcony and panorama view of the city silhouette, one suggests that the user intends to investigate the value of the real estate for an upcoming sale and not for taxation purposes.

In the latter case, where the system gains its understanding by communicating with its user, an *explicit goal acquisition* is utilised. As an example, the user could explain that the analysis is intended for a loan application and hence, the analysis needs to consider the aspects desired by the bank.

3.4.3 Product analysis

The system uses the found goal specification to evaluate the product. Depending on whether the critic opts at a full understanding of the problem domain, i.e. is a *differential critic;* or merely checks for a set of predefined problems in the user's solution, i.e. is an *analytical critic*, the strategy of analysing the user's input differs (Fischer et al., 1990).

Differential Critiquing

With a *differential critiquing approach,* the system first generates its own solution. Then, critique pointing out relevant differences between the user's and the system's solution is generated by the use of heuristics. An example of a system utilising the differential approach is the ATTENDING-system, the anaesthetic therapy application by Miller (1984).

The differential approach is complex to use in domains where the possible solutions are radically different and not suitable for a comparison emanating from the user's point of view (Fischer et al., 1990). The ATTENDING system handles the ordeal through parsing the user's solution into a hierarchy and by evaluating each subtask in a top-down manner. In other implementations, metrics have been used to handle these shortcomings (Fischer et al., 1990). Still, this issue is complex since the metrics could be subjective and sometimes conflicting.

Analytical Critiquing

In the *analytical critiquing approach* the system's analysis is based on knowledge of predefined features desirable or non-desirable in a good solution in the domain (Fischer et al., 1990). The critique is generated through a pattern match between the user's solution and the guidelines. Other techniques include finite state machines, augmented transition networks or expectation-based parsers (Fischer et al., 1991). Understanding of the full domain model is not needed and hence makes the system suitable for design applications where no one optimal solution exists. JANUS – the kitchen design critic by Fischer and his associates (1990), adopts the analytical critiquing approach.

3.4.4 Adaptation

The use of an intrusive critic in a learning environment, where the critic persistently criticised the user's somewhat different viewpoint could destroy the user's motivation to learn. Hence, a critiquing system must be able to adapt the critique to different users, their different learning styles and to accommodate different levels of skill.

The adaptation capability could be either *adaptable* or *adaptive*. *Adaptable* systems allow the user to change the system's behaviour. An *adaptive* system on the other hand, is automatically changing according to observations or inferred information. Often it could base its settings on a user model. In practice, many of the implemented expert critiquing systems are statically directed at an intermediate user (Harrius, 1993). Thus, the novice or average user could have problems assimilating the critique. The expert user, on the other hand, would probably find the critique too general and hence of less value (ibid.).

3.4.5 Explanations and justifications

The critic needs the ability to explain and justify its interventions to the user. The explanations help the user to incorporate an own understanding of the deep knowledge and reasons firing the critique, as well as whether to accept or ignore the provided critique (Fischer et al., 1990).

Moreover, the explanation and justification of the content of the critique, should present the user with convincing arguments for and against the user's solution and preferably the corresponding relations between them (Harrius, 1993). For instance could critique on a real estate market value analysis that does not includes the status of the roof of an older house, be explained by informing the user of the life-length of a house roof and the cost of renovation.

In the case of a differential approach on critiquing, where the system also generates an own solution to the problem, the differences between the system's and the user's solution need to be justified as well as the system's solution in itself.

3.4.6 Advice giving

In extension to the identification of inferior strategies in the user's problem-solving, some critics are capable of suggesting alternatives to its user to improve the user's solution. These critics are called *solution-generating critics* (Fischer et al., 1990). An example would be if the critic detected that, the user has left out the area of the real estate. A solution-generating critic could have the capability to suggest different sources for acquiring the area.

3.5 Issues of consideration in the critiquing approach

The presented findings on utilising the critiquing approach when supporting human problem-solving, indicates that the critiquing paradigm offers many advantages. The user needs to be active and is allowed to share initiative in the dialogue while receiving critique on his/her own findings. Still, some issues remain.

The critique should aim at supporting and optimising the user's own model of problem-solving. Therefore, the focus of the critiquing process should be on identifying correct, irrelevant and overlooked or missing knowledge in the user's solution. Hence, the knowledge representation needs to provide the possibility of rich explanations and justifications of the critique. Moreover, the knowledge representation needs to be easy to create and maintain, and still enable the same expressivity as the knowledge representation of the ordinary environment (Harrius, 1993). The system must also ensure that the generated critique is sound, i.e is coherent with the expert knowledge of the domain environment.

Additionally, the critique needs to be useful and adapted to its user and thus, presented in a way and time not perceived intrusive to the user. Since natural language is complex, the presentation of the generated critique should be made in a form communicative to the user and still easy to construct and represent. Often, the critique is generated as a complete batch-presentation and ignores the user's capability of guiding the adaptation of the presentation. Instead, a dynamic model of the user is provided through the interaction with a student model, which is complex to develop and not always provides a correct interpretation of the user.

A conceptual model of a critic addressing these issues is presented in the following chapter.

4 CORRECTUS IICI - Proposing a user-oriented critiquing methodology

The chapter contains considerations on a user-oriented critiquing methodology, CORRECTUS IICI[2], i.e. a conceptual model of a critic. To optimise the user-system synergy, the critiquing methodology utilises the user's as well as the system's different strengths. By employing the user as an equal partner in the problem-solving, the critiquing methodology is thought to provide means to support and optimise the user's learning of problem-solving strategies from a knowledge domain. Hence, the user's own activity and interpretation-skills are key factors in providing the user-adapted decision support and learning.

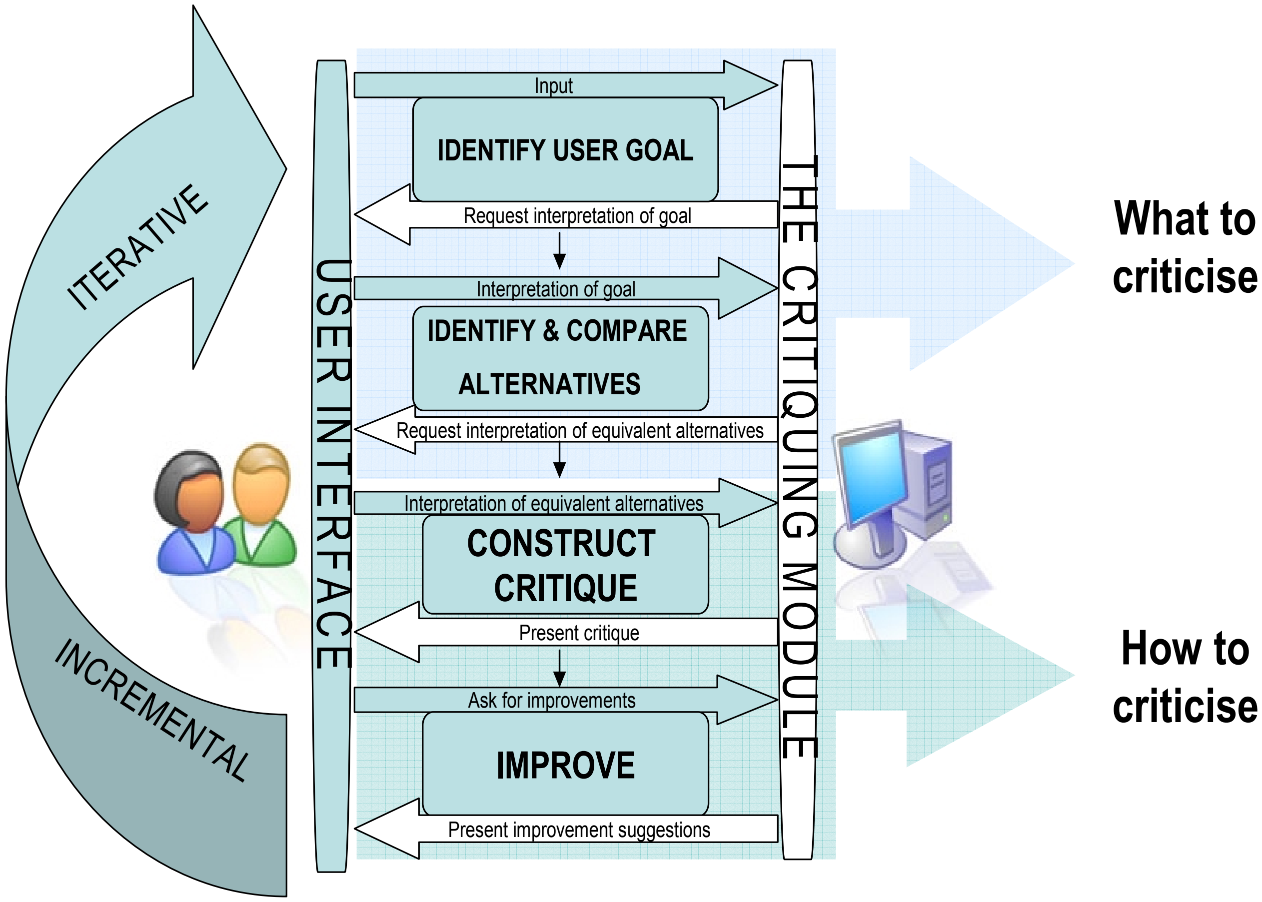

Figure 6. CORRECTUS – a user-oriented critiquing methodology. The critique is generated in an iterative and incremental way through user-system cooperation.

The proposed methodology (see figure 6) addresses the two main constituents of critique; *What to criticise* and *How to criticise* the user's problem-solving strategy and solution (see figure 2) through the interactive user-system cooperation. Moreover, the critiquing

[2] IICI is the first letter in each of the different phases of the model. The Swedish pronunciation of IICI is iisi (like the English "easy"), Hence, Correct us easy.

methodology also addresses the presentation content and user interaction aspects of the critic. These considerations will together with the author's own reflections be discussed in this and the following chapter.

The notions used in this chapter are *object type, object theory, goal object type, functionally equivalent object, meta-theory* and *informal theory*.

In a rule-based system, the *object type* should correspond to the conclusion part of a rule and the *object theory* to the set of rules formalising the domain knowledge. A *goal object type* is an object type which is expressing the user's intended goal. A *functionally equivalent object* is any premise, given the same specific goal and context, which will result in the same conclusion and perform the same functions for the given situation as the user's premise. Hence, the functionally equivalent object and the user's object are to be regarded as equivalent substitutes to each other (Newell, 1982; Lucardie, 1999). The *meta-theory* is the set of rules providing the system's inference mechanism and the holder of the critiquing module. The *informal theory* is the complementary knowledge not formalised as rules, but still needed to understand the systems reasoning and learn from the domain.

The user-oriented critiquing process generates the critique in four steps (see figure 6): Identify Goal, Compare, Construct Critique and Improve.

4.1 Determining what to criticise

The first phase of generating the critique focuses on determining *what* to criticise. This is done through establishing how to interpret the user's solution and problem-solving strategies as well as the relevance of any alternative solutions to the user's goal. The key to the analyses is the user.

4.1.1 Identify user goal

The critiquing process starts when the user inputs his/her solution to the problem at hand. The system's *meta-theory* (MT), i.e. its inference mechanism, analyses the user's input. The meta-theory tries to find an *object type* for the goal the user is trying to prove through investigating the *object theory* (OT), i.e the formalised domain knowledge. In order to enable a correct understanding of the goal and its connection to the object theory, the meta-theory asks the user of his/her interpretation of the object theory perceived to be connected to the object type matching the goal. It is done by displaying to the user the object theory and its context, i.e. a rule and its consequences to the reasoning. In the example of setting the "correct" value of a real estate, the different value analyses like market value, taxation value et c. could be displayed together with explanations on their respective areas of use.

The meta-theory asks for the user's interpretation on whether that object type matches the user's intended goal and the specific context to the problem at hand. Then, the user is able to confirm or reject the connections.

In this way, the user establishes an explicit connection between his/her own intended goal, here finding the correct market value of the real estate, and the domain knowledge formalised in the system, the rule (conclusion) corresponding to market value analysis. Thus, the critiquing methodology employs an explicit goal acquisition strategy. In this way, the methodology can also provide the system with knowledge on non-connected structures of the domain, if any such alternatives exist. In addition, the system gets confirmation on the true intention of the user from the source most able to provide it, namely the user him-/herself.

4.1.2 Identify and compare equipotent objects

After the user has provided the system with his/her interpretation of the object theory, the user's solution is connected to the object theory in a way affirmed by the user. The meta-theory now holds a *goal object type* and the information needed to match the user's solution with other object types. This matching allows the meta-theory to establish any alternative constructions of the goal object type.

Wenger (1987), states that the expert module by necessity embodies a specific view of the domain. "The representation language it uses and the concepts it takes… are choices made by the designer that bias the entire presentation." (Wenger, 1987 p 16) This implies that the system needs to reason on *functionally equivalent objects* and not aim for an exact match (Lucardie, 1999). Hence, a sunlit balcony is functionally equivalent to a garden terrace if it has the same implications to the goal object type, i.e. the analysis of market value.

If any alternatives are discovered, the system asks the user for his/her interpretation of these additional object types. The resulting answer is either that the user confirms the equipotence of this alternative construct and the goal object type, or that the alternative construct is refused on the basis of the user's interpretation. This enables the system to establish the subset for the analysis.

Thanks to the user's contributions when interpreting the functional equivalence of the alternative rules, the system is able to infer solutions which are functionally equal to the goal object type although not exactly matching the user's explicit statement of his/her goal. Thus, the system can find a more exhaustive subset of the rules, which provides the same or an equivalent result to the goal object type.

In other words, the system is given the means of making a more comprehensive collection of alternative solutions; and is able to gain a better understanding of the user's solution. Hence, the system does not need to develop an independent solution to the problem; it merely checks the user's solution for consistency and accuracy with the domain theory, i.e utilises an analytical critiquing approach. This ensures the soundness of the generated critique.

By investigating the functional equivalence of any objects in the domain, the meta-theory is also able to generate an error model for the user's solution. The irrelevant, overlooked and missing knowledge thus can be identified through the mentioned consistency check of the domain. Hence, the existing knowledge representation of the system can be utilised, provided it holds deep as well as surface knowledge and the correlations between them.

4.2 Determining how to criticise

Now, the critiquing methodology enters its second phase. To optimise the user's learning of problem-solving skills, a critiquing module needs the ability to present the user with individually adapted critique. It also needs to justify its interventions as well as the critique itself.

4.2.1 Construct critique

From the results of the previous phase, the meta-theory now can determine any weaknesses in the user's solution. The incorrect knowledge, e. g. including the current owner's nice collection of Monet paintings, as well as any overlooked or missing knowledge, e. g. leaving out information on the status of the roof, can be identified, and also the alternatives preferable in this case, i.e. what in the user's input solution to criticise. Thus, the critic is ready to construct the critique.

The critique is provided to the user as a presentation of the knowledge formalised in the object theory subset, which corresponds to the user's strategy and solution. The presentation preferably allows the correct, incorrect as well as overlooked or missing knowledge to be identified and understood by the user.

Normally, in order to adapt the critique and presentation to the user, the level and amount of arguments is claimed necessary to be established by a user model. Instead, by using a minimalist explanation strategy, the critiquing methodology provides the tools for the user to provide self-explanations for the presented critique. Hence, the user incrementally could unfold different layers and perspectives justifying and explaining the critique. This allows the user to form an own understanding of the validity of the critique.

Conclusively, convincing arguments for and against the user's solution, i.e both positive and negative critique, and also explanations of the corresponding relations between them is provided to the user as tools for self-explanations, reflections and further inquiries.

4.2.2 Improve solution

The user can ask the system for suggestions on improvements. These improvements are identified from the user's own solution after a comparison to the equivalent object theory subset. The improvements are presented as suggestions on how to exclude irrelevant knowledge and to incorporate overlooked and missing knowledge in the problem-solving strategy. In the case of an object theory based on rules grouped according to certainty factors of MYCIN type (Clancey, 1979), the recommended improvements can be based on the rules supporting a higher degree of certainty than the rule corresponding to the user's initial solution. In the provided example, the user could be informed that including information on proximity to the city and communications could make the correctness of the set real estate value more certain.

4.3 An interactive user-system cooperative approach on critique generation

The proposed critiquing methodology handles the user diagnostic feature in an easy and straightforward way. The user's own solution is the focus of the analysis and guides the generation of the system's critique.

The proposed methodology bases its analysis of the user's solution on a coherency check of the user's input and the problem domain. Hence, it applies a dynamic approach of analytical critiquing. Correct, irrelevant, overlooked or missing knowledge are identified and allows positive critiquing as well as highlighting any deficiencies in the user's reasoning.

Through the interactive dialogue, the content of the critique could be focused on the individual user's current need. Thus, no need of interacting with a user model to generate hypotheses about the possible basic misconceptions of the user's error or establishing the user's goal is required. Moreover, the user is provided with the means and control of generating customised self-explanations to justify and explain the critique. This is thought to strengthen the user's own conceptual model of the problem domain and hence optimise the user's problem-solving knowledge.

Still, the presentation of the generated critique was considered as a complex issue in the previous chapter. According to Fischer et al. (1991), the critiquing modules were preferable for a cooperative user-system approach and are best utilised while embedded into design-environments. Hence, the utilisation of an existing work environment and its components for representing, presenting and explaining the critique needs to be further illustrated. An elaboration on such integration of the critiquing methodology is discussed in the following chapter.

5 A framework supporting transfer of problem-solving knowledge

It is judged important that the proposed critiquing methodology is utilised in a system-architecture able of supporting the needed presentations, sufficient explanations and user-interaction. In accordance to the view of the user as an equal partner in the process, a semi-formal system architecture seems appropriate (compare to the LISP-CRITIC by Fischer, 1990). In the semi-formal architecture, the system presents information to the user, who provides an interpretation.

Consequently, the CORRECTUS IICI-methodology is incorporated into an existing semi-formal meta-theory of a hypermedia architecture developed by Edman and Hamfelt (1997; 1999). As the previous research on Steamer, and to some extent, ONCOCIN has shown, this approach seem possible (Langlotz & Shortliffe, 1983; Forbus, 1984). The extended architecture is thought to illustrate the utilisation of the critiquing methodology as means of supporting and optimising the transfer of problem-solving knowledge. An example session with an existing implementation of the architecture is provided as a realisation of a critiquing system based on the CORRECTUS IICI-methodology. Thus, some remaining issues on utilising the user-oriented critiquing methodology are handled in this chapter.

5.1 The system-architecture

The architecture of Edman and Hamfelt (1997; 1999) is defined as schemata on a meta-level. This is useful since the schemata can be used as a high-level blueprint in designing critiquing systems, which according to Aiello, Cialdea & Nardi (1991), is seen as important.

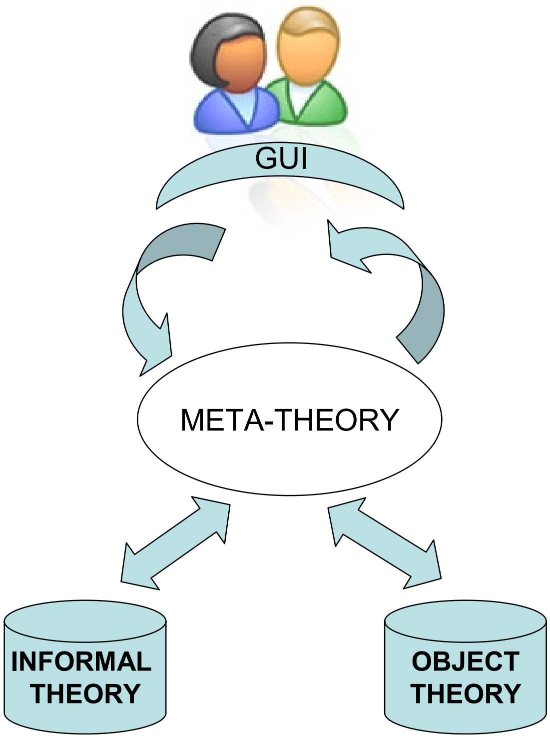

Figure 7. The semiformal meta-theory and its interrelations (Source: Edman & Hamfelt, 1999, p 1011, adapted by the author).

The system-architecture consists of four main components (see figure 7). The informal (domain) theory, the object theory, the meta-theory and the graphical user interface (GUI).

The informal theory and the object theory hold are the knowledge sources in the utilised framework. The object theory is a collection of axioms in first predicate order logic, which represents the problem-solving knowledge, e.g. in the form of rules. The informal theory contains reproductions of the domain context in the form most appropriate e.g. as images, sound, diagrams and text.

The reasoning of the object theory and the informal theory, i.e. the deduction mechanism, is handled by axioms in the meta-theory.

The meta-theory is also handling the user interaction and the incorporation of the user's input to the system. Through the graphical user interface, the presentation of the system's knowledge is presented to the user (Edman, 2001).

By extending the generic meta-theory with a critiquing module, i.e. a set of axioms incorporating the CORRECTUS IICI-methodology, critique can be generated in coherence with the domain theory.

5.2 Analyse More

In order to illustrate the utilisation of the proposed critiquing methodology in the semi-formal architecture, an existing knowledge-based system called Analyse More is presented. The system was developed at the former department of Computing Science at Uppsala University. The system, designed in co-operation with experts at the Swedish Environmental Protection Agency and teachers, supports students in measuring and estimating status of a lake and educates the students in such environmental issues (Edman, Lindman & Sundling, 1993; Edman, 2001).

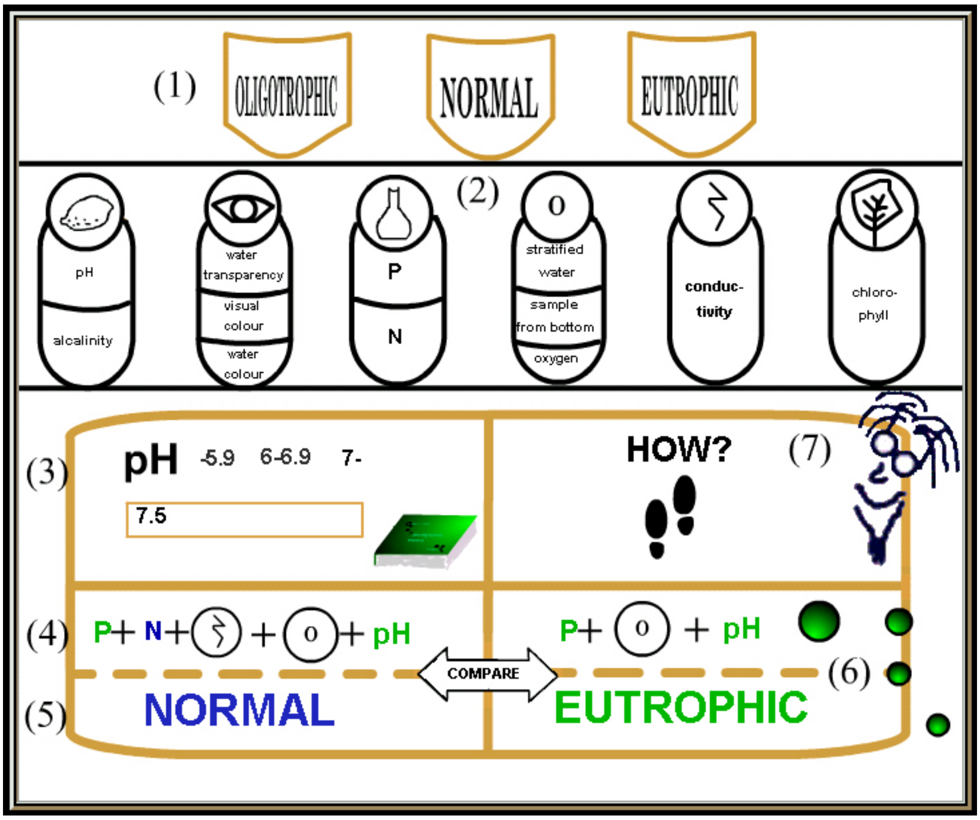

Figure 8. The user interface of the chemical-physical domain of Analyse More (Source: Eriksson Lundström, 2003 in Edman & Mayiwar, 2003)

By using different chemical physical or biological observations and indicators, the knowledge-based system provides a diagnosis of how nutritious the examined lake is. Here the chemical physical domain is utilised in a realisation of the proposed critiquing feature. A presentation of the domain and user interface of Analyse More is given, together with a discussion on introducing a critiquing feature in the existing system.

5.2.1 The domain

The domain is small and constrained, and requires complex measurements and observations of the indicators. Due to the different ways of extracting the measurements, a user could make several mistakes or errors such as choosing a poor method, material or simply using broken equipment. These errors could result in the insertion of conflicting or incorrect measurements as well as make the user exclude or overlook important information. If a mistake or an error occurs, the system needs a way of explaining the problem to the user in order to enable a correct diagnosis of the lake.

Moreover, according to the educational aim of the system, the goal of the system is to provide the user with maximum knowledge on the domain and the solution. Hence, it is important to correct and guide the user in situations where he/she is incorrect or uncertain in his/her own diagnosis of the found measurements, and to provide the user with a transparent view of the domain and reasoning strategies.

The possible categorisations of the lake are oligotrophic (poor of nutrition), eutrophic (nutritious) or normal (see figure 8 [1]). The chemical physical indicators are pH-value, alkalinity, water transparency, visual colour, water colour, phosphorus (P), nitrogen (N), stratified water, sample from bottom, oxygen, conductivity and chlorophyll (see figure 8 [2]). This knowledge, i.e. the object theory, is stored as rules in a form similar to the ones of this very simplified object theory:

```
[1]   Oligotrophic:-
      AcidityHigh,
      Chlorophyll >15.

[2]   AcidityHigh:-
      Alkalinity <0.05,
      pH <6.

[3]   AcidityLow:-
      Alkalinity>1,
      pH >7

[4]   Normal:-
      pH >=6 & =<7
      Phosphorus >=10 & =<20.

[5]   Eutrophic:-
      pH >7,
      Phosphorus >20.

[6]   Eutrophic:-
      AcidityLow.
```

For instance, the first rule [1] states that the lake is `Oligotrophic`, if `AcidityHigh` is true and `Chlorophyll` is greater than 15.

The domain knowledge represented in the system's object theory is used by the meta-theory in correlation with the reproduced contextual knowledge in the informal theory and the user's input answers. Depending on the indications from the input values of each session, different combinations of the indicators are used in the diagnosis (Edman, Lindman & Sundling, 1993). For instance, to be able to conclude that a lake is `Oligotrophic` according to rule [2] in the above provided example, the test results of alkalinity and pH are combined to make an intermediate conclusion of the condition of acidity, i.e. `AcidityHigh` or `AcidityLow` in the lake. It is this value, i.e. the indication of the combination of the two parameters, that is used by the system (together with the singleton parameter `Chlorophyll`), to infer that the lake is `Oligotrophic`.

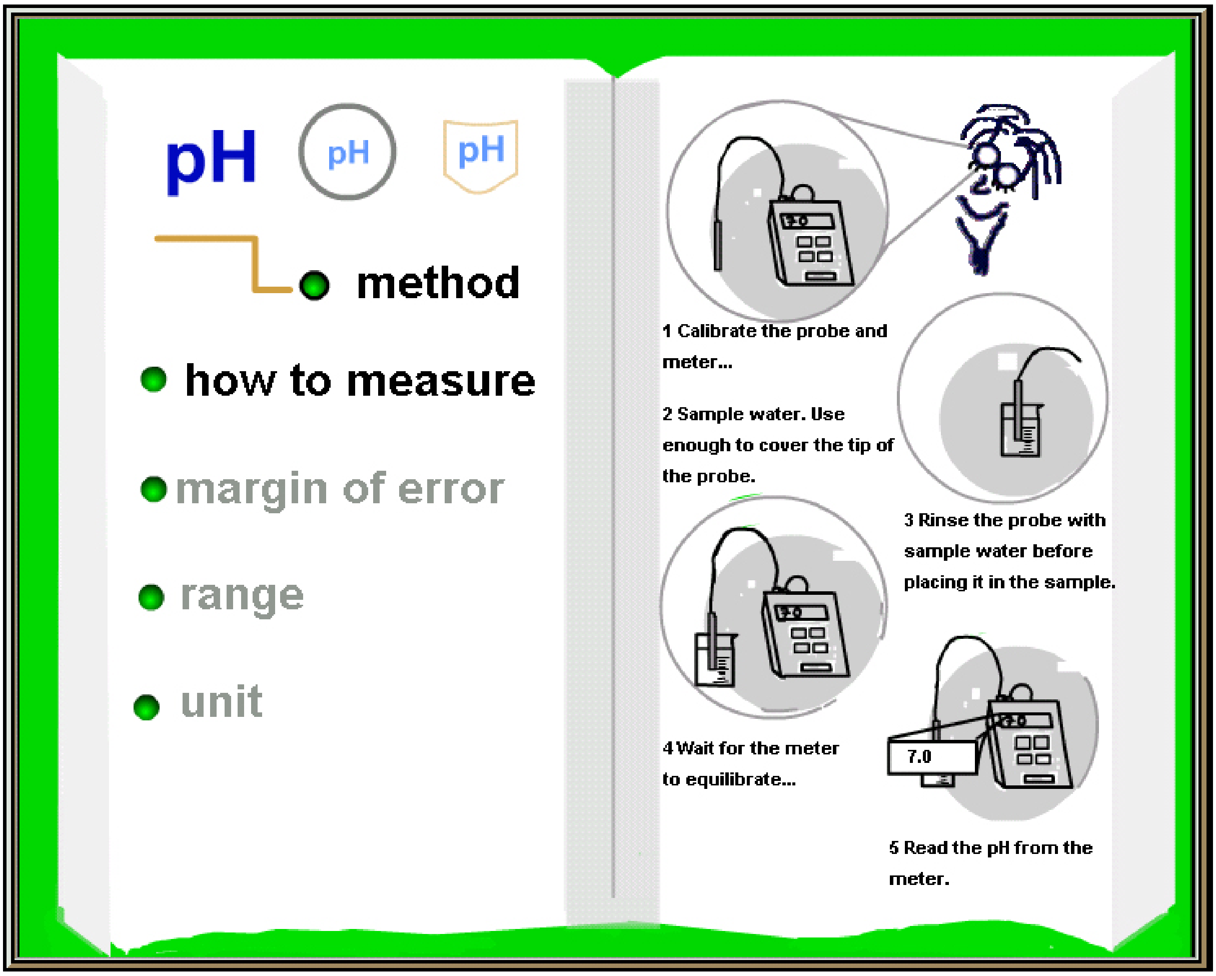

Figure 9. The reference book, depicting a why-explanation of the method of how to measure pH-value. (Source: Eriksson Lundström, 2003)

The knowledge reproduced in the informal domain theory is stored in different formats, as earlier stated, as images, sound, diagrams or text. This knowledge is not directly used in the reasoning, but provides the user with complimentary information necessary to make the user understand the domain and the system's way of reasoning (Edman, 2001). For instance, it could be a description of how to extract a needed measurement (see figure 9). This additional information supports the user's understanding of the domain and may increase the quality of the measurements provided.

The domain knowledge represented in the object theory and the knowledge that is reproduced in the informal theory, enable the system to provide its user with context-based explanations (Edman, 2001). These explanations contain information on the system's reasoning structures, complimentary information, which justifies the data used in the reasoning; the systems general control strategies and finally, term descriptions like unit and interval of measurements. The explanation feature is elaborate and offers the system user a comprehensive range of explanations.

Still, the author believes that by incorporating a critiquing feature based on the proposed critiquing methodology of CORRECTUS IICI, the explanations could be extended and user-oriented in a way favourable to the diagnostic and educational goals of the system. The first and second phases of CORRECTUS IICI identify the user's intended goal with his/her input data and make the user aware of better or functionally equivalent alternatives of the goal and sub-goals during the problem-solving session. Hence, in an incremental and iterative way, details from the user's own perspective at the specific moment and current way of reasoning could be explained. This is thought to give the user a more transparent view of the current alternatives and their consequences, and thus, facilitates the user's creation of a mental model of the domain. In turn, it decreases the risk of the user inputting incorrect data and possibly enhances the user's learning of the domain knowledge and the system's problem-solving strategies, since the critique and improvement suggestions are provided from the user's own perspective. The understanding of "why" but also "why not" a certain measurement should be used is also facilitated through the critiquing feature, thanks to the possibility of displaying functionally equivalent alternatives and improvements.

5.2.2 The user interface

The user-system interaction is conducted through the user interface. The emphasis in the user interface is given to the visualisation and reinforcement of the conceptual structure, connections and concordance among important objects, i.e. the indicators, of the domain, through an interactive map (see figure 8 [2]) (Eriksson Lundström, 2003). The interactive map supports and maintains a mental model in the mind of the user, providing a structure and frame for the domain knowledge acquired (ibid.).

Initially, the user is introduced to the different solutions (figure 8 [1]) as well as the indicators of the domain (figure 8 [2]).The indicators are depicted as icons allowing the user to input his/her measures and solution. The indicators conceptual concordance is shown through the grouping of the indicators into intermediate objects. The properties of each intermediate object are represented through an icon. For instance, a lemon represents the intermediate result acidity.

According to Löwgren (1993), the expressive power of any computer system is dependent on how well the user interface design enables the user's usage of the system feature. Hence, the design of the interface is very important to the user's perception of the critique as well.

Still, the interaction with the existing hypermedia user interface (see figure 8) is thought to provide means to handle the extended user communication and critique administration, possibly with a simple addition of a dialogue frame. According to Silverman (1992b), due to the complexity of natural language generation, the user interface of a critiquing system should enable the use of visual metaphors and direct manipulation of the work item (Silverman, 1992b).

The existing user interface not only supports the use of visual metaphors, but also the use of various media when reproducing the object theory as well as the domain context from the informal theory. Hence, the user can choose the presentation form that he/she considers most convenient since alternative presentation forms are available e g sound, images and text strings. Furthermore, the hyperlinks of the hypermedia user interface allow the user to explore the explanations of the knowledge underlying the critique through an incremental approach by the user's own choice. It allows the adaptation of the critique to the individual user since the accommodation of different user preferences and learning styles can be considered (Edman & Mayiwar, 2003).

The user's initial solution and the system's critique can be administered by the explicit domain structure and the use of colour-codes. The icons together with the user's input and intermediate results contain sufficient tools to capture the user's intentions and strategy since the use of direct manipulation and the colour-codes provides the critiquing module with unobtrusive access to the user's cognitive processes (Guerlain et al., 1999).

Conclusively, the chemical physical domain of the system Analyse More and its user interface seems well suited for a complementing critiquing feature based on the proposed critiquing methodology of CORRECTUS IICI.

5.3 A realisation of a critiquing system

Incrementally and iteratively, the proposed critiquing methodology of CORRECTUS IICI is thought to enable the user to understand and improve his/her solution in handling the given problem. In Analyse More, the aim is twofold, to both improve the diagnostic quality as well as maximise the user's learning. By making the user aware of which information to include, keep and exclude, the user's problem-solving strategy and thus, the user's understanding of the domain and the system's problem-solving strategies, is thought to be improved.

5.3.1 Phase I: Identify user goal

The user inputs data and an own solution through the interactive map. In the example of figure 8, the user has given values for the indicators Phosphorus (P), Nitrogen (N), and pH, as well as intermediate values of the water conductivity (the lightning) and oxygen in the water (O), the users solution is NORMAL. In figure 8 [3] the user is currently providing the measurement of the indicator pH. The different intervals serve as a presentation of the available alternatives. It could be noted that the intervals of values of the indicator pH, corresponding to the different classifications of the lake-status are marked with the classification's specific colour. Hence, the user is provided with an influencer, albeit no explicit control is displayed since no value is being favoured by the system. Explanations of the domain context such as the method of measurement (see figure 9) are reachable through the book-icon (see the green book in figure 8 [3]), enabling the user to acquire a richer context of the problem-solving knowledge, hence facilitating deep learning (Steels, 1990).

The user of our example session answers that the measurement of pH is 7.5. This input is analysed by the system. Here, the system finds a presumptive goal object type in the rule `Euthrophic:- pH >7, Phosphorus < 20.` Now, the value of the user's input is displayed in the trace-space (see figure 8 [4]) and instantly gets coloured according to the indicated classification. It is an indication to the user of the system's interpretation of his/her current strategy, i.e. that the value indicates to the system that the status of the lake is eutrophic. Moreover, the colour-coded display serves as a request to the user to approve or reject the made interpretation. This provides the user with correct and timely critique during the problem-solving and also allows the user to alter his input data to reflect the desired result.

5.3.2 Phase II: Identify & Compare Alternatives

The user can reject the interpretation through changing the input value or accept the interpretation by continuing his/her work. A more elaborate explanation could be activated if the user wants to further investigate why the system has inferred an alternative interpretation or if any functionally equivalent options exist. This critiquing feature can be activated according to the user's choice for reviewing the solution, and thus, the user's input data is used to generate the critique at a point when the user is ready to handle an analysis of the whole solution. In the case of our example object theory (see section 5.2.1), the critic could make the user aware of an existing functionally equivalent object of `pH`, namely `AcidityLow`, which is a combination of pH and Alkalinity.

5.3.3 Phase III: Construct Critique

After examining and inputting values of all indicators the user deems necessary and sufficient for the classification, he/she analyses the input data and suggests a solution (see figure 8 [5]). As stated, in this case, the user has suggested NORMAL. Then the system provides its solution (see figure 8 [6]), as well as the used indicators and values displaying to the user how the particular indicator is used in the reasoning as well as its relative importance in the actual conclusion. In our example, the system's diagnose differs from the user's.

The generated critique in our Analyse More example is provided through the visual comparison between the user's solution and the system's solution. Hence, the deficiencies of the solution can be categorised as either irrelevant data or omitted data. The irrelevant data only exists in the user's solution and the omitted only exists in the system's solution (see figure 8 [4]). Compared to the error model of Silverman (1992 a), the first category is irrelevant knowledge A and the latter, overlooked and missing knowledge C and D. As in the case of phase II, a more elaborate explanation could be activated by the user. This explanation could contain reasons to why or why not a certain value is to be preferred in the user's actual diagnosis. Moreover, thanks to the existing explanation feature of the system, the user has access to deep as well as surface knowledge justifying and explaining the provided critique in a way and level chosen by the user him-/herself.

5.3.4 Phase IV Improve

In the case of Analyse More, the system's solution serves as a firsthand improvement suggestion. However, during the user's problem-solving, the colour-codes used to identify the user's intended meaning have an additional functionality, since the colour-codes can be used to provide critique of the user's intermediate decisions. The different colours indicate to the user the importance and consequences of his/her current decision. This provides critique in a timing judged most appropriate since it is not administered beforehand and influencing the user, neither after when the user has his/her whole strategy set, but immediately after the user's current cognitive process is definitive and has been made explicit. The user has the opportunity to alter his/her decision, due to the critique displayed by the system through the colour-codes, indicating the consequences of the chosen strategy or input. This provides an interactive critique generation, which focuses the critique on the user's most recent doings, and still is able to provide critique on the whole solution whenever the user wishes.

Conclusively, by incorporating a critiquing feature based on the proposed critiquing methodology of CORRECTUS IICI, the additional functionality provided is thought to enable improved diagnoses as well as facilitated learning of the chemical physical domain of the system. Since the user interpretation is unique to the individual user and his/her current interpretation of the given situation (Edman & Hamfelt, 1997), a dynamic, logically sound and adapted critique in direct correlation to the informal theory and object theory relation is provided by the meta-theory. Hence, no outside disturbances could distort the interpretation of the user's intentions. This supports the user's problem-solving in a positive way and provides support and feedback dynamically from the synergy of the problem domain and the user's current interpretation of the domain and the problem-solving structures. Hence, the critique and problem-solving support is neither dependent on the system- or student model designer, nor on any external interpreter mechanism.

6 Conclusion, discussion and further work

The thesis is concluded in this chapter. The first section contains remarks on the proposed critiquing methodology and its capabilities of aiding the transfer of problem-solving knowledge. The following section holds a discussion on the integrated critiquing approach utilised in the methodology. Finally, the author's suggestions on further work are made.

6.1 Conclusion

The aim of this thesis has been to explore the critiquing approach and its capability of supporting the transfer of human problem-solving knowledge. The focus of the elaboration lay on the user's own potential to aid and guide the user diagnostic feature of the system.

The area of human cognition and learning has been addressed and an extensive elaboration on the critiquing approach has been presented in order to provide the reader with the foundation of the proposed critiquing methodology and the following discussion.

6.1.1 The user-system interaction

To enable an easier way of providing adequate and individually adapted user interaction than the use of a student model allows, the proposed critiquing methodology employs the user's, i.e. the human, capabilities of interpreting data. It is the user who controls the perspective and interpretation of the cooperative solution. The view of the user has been as an equal partner to the system. Not as in many cases, a bystander to the system.

The primary task of the system in the proposed user-system interaction is to provide a structured framework that supports the user's problem-solving. In this setting, the user's current comprehension of the problem-solving knowledge is presented to the user as the corresponding formal utterance and its coherency to the domain theory.

6.1.2 Supporting and optimising the user's problem-solving

The proposed critiquing methodology intends to, in a cooperative and interactive way, present to the user the formal outcome and the consequences corresponding to the user's current strategy. Incrementally and iteratively, the methodology helps the user to understand and improve the quality and implications of his/her solution in handling the problem. In other words, the methodology supports the user in understanding which information to include, keep and exclude in order to optimise the problem-solving strategy.

The integration of the critiquing methodology into an existing system-architecture allows the critic's expressive power to increase. This also increases the adaptation of the user support since specific pieces or parts of problem-solving strategies can be criticised, while still preserving the soundness and the general structures of the domain.

Thus, by the use in a knowledge-based environment, like the system-architecture for hypermedia, the proposed critiquing methodology will enable the generation of sound, dynamic and appropriate feedback to support the user's learning of problem-solving skills from the knowledge base used in the educational environment. By utilising the knowledge base of the supporting environment, the same expressivity as in the work environment itself is naturally ensured. Hence, the soundness of the generated critique and its explanations are dependent on the consistency of the knowledge base of the environment. This is attractive from the maintenance perspective.

The critique is dynamically generated, on the user's explicit intentions and for the specific problem at hand. By using a minimalist explanation strategy, which through the interaction with the user incrementally unfolds different layers and perspectives of the task in question, the adaptation of the presented critique can be made in a way decided by the user. This helps the user understand and optimise his/her own problem-solving strategy. Moreover, since this is achieved through the user's own exploration of the domain, the final optimised solution is not only strongly anchored in the user's understanding. Actually, it *is* the formal utterance of the user's own mental model of the problem, problem-solving strategies and the resulting solution.

Conclusively, this approach is thought to provide focus on the user's most recent and detailed need during his/her interaction with the system. Still, within an interpretation of the domain coherent with the knowledge formalised in the domain theory or its interpretation. Hence, the generated critique could remain coherent with the problem domain, while enabling a more dynamic user-oriented learning of deep as well as surface structures of the problem-solving knowledge and domain structures.

In other words, by utilising the strength of the critiquing approach and the meta-theory's ability to ensure the consistency of the domain theory, the proposed critiquing methodology establishes a positive addition to the field of supporting and optimising the user's learning of problem solving knowledge, and possibly to the user-system interaction perspective in a broader sense.

6.2 Discussion

This thesis set out to address the need of providing an adequate and easy way of automating the essence of educational interaction. The approach explored should allow user-activity, user-oriented support and include the user in the adaptation of the user support.

With the critiquing approach, the user is supported in his own exploration of the problem domain (Andresen, 1991). Hence, the user is active and able to create more innovative problem-solving solutions together with the system, without being limited by the expert strategies that initially was formalised in the system. The user creates his/her own understanding, and thus is motivated and informed in all steps of the problem-solving without needing skills in logic or computer programming.

The capability of the critiquing approach of criticising details of the user's goals has been viewed as one of its weaknesses. Fischer and his associates states that "Supporting users in their own doing means that details of user goals are often not available to the system, limiting the specificity of the critique the system can provide."(Fischer et al., 1990 p 346). By employing the view of the user as an equal partner in the user-system cooperation, this shortcoming is addressed since the system is able to access a richer understanding of the user's goal by interacting with the user.

Albeit, the interactive approach is difficult since people are not always good at explaining their own mental processes. Hence, it could be argued that there might be a danger in relying on the user. What will happen if the user interpretation is incorrect? Still, by employing a coherency-check of the user's interpretations and the solutions, the critique and problem-solving are constrained to be coherent to the domain. In other words, the system checks that the user's interpretation is logically correct according to the existing knowledge. Thus, the development of the meta-theory gets more complex, but the soundness of the generated solutions should be protected.

Moreover, it could be argued that the system cannot actually see or evaluate that the user has acquired the knowledge, since no tests are performed. Despite the attraction of a system truly able of interpreting the user and understanding his/her cognitive growth, the complexity of modelling the understanding of the user, currently makes this feature unfeasible. Other more straightforward approaches need to be employed. The words of Ulrika Leimar (see section 2.3) provides an implication on finding a way of supporting the user's learning of problem-solving knowledge, implying that giving the user the means to learn is to adapt the support to the user's own cognitive processes. By using unobtrusive presentation, the user him-/herself decides which level and amount of explanation to address, thus, restraining the practical limitations of the system not being able to evaluate the user's current comprehension of the domain.

Due to the complexity of formalising the user diagnostic feature, as well as the support from the constructivistic view, the utilisation of the user in adapting and interpreting the solution and critique seems appropriate. The result of the critique, the cognitive growth, takes place in the mind of the user. The user's answers to the questions, i.e. problem-solving capability, then become the indication on the learning degree. The learning process need not to be explained to happen. "The student learning status can be viewed in terms of knowledge acquired and his answers or solutions to problems can be seen as the result of a reasoning process"(Aiello et al., 1991). Therefore, the system does not have to understand or hypothesise on why the student made his/her mistakes, but merely to provide the user with the means of developing optimised self-explanations to aid the learning or problem-solving. Albeit demanding and relying on the user's activity and motivation, the critiquing methodology provides a way of utilising the user-system synergy in supporting learning while having the user's own solution as focus. Thus, the system and user together optimise the user's problem-solving.

Finally, it could be argued that this approach will decrease the transparency and intuitiveness of the system since different users will get different critiques and different approaches. Since these solutions is thought to provide each user with the most optimised understanding, the proposed perspective of the critiquing approach will provide the same functional result to each user. It will optimise his/her problem-solving ability without discarding any user of a lower or higher level than the system or student model designer has thought of when designing the system or student model. The limits, as well as the important demands of clarity, real world correspondence and usefulness of the knowledge used, are set by the domain and the meta-theory's ability of interpreting and incorporating informal theory and user interpretation in the object theory.

Conclusively, the critiquing approach seems to hold capabilities of providing an adequate and easy way of automating the essence of the educational interaction, while employing the user as an equal partner in providing adapted support to the user in order to optimise his/her learning of problem-solving knowledge.

6.3 Further work

In order to criticise the user's solution properly, a critic implementing the proposed critiquing methodology needs the ability to reason about knowledge and reasoning. Providing a formalisation based on first predicate order logic would enable a direct transformability of the proposed methodology into programming languages based on logic (Aiello et al., 1991). Hence, it would allow a desired ease of integrating the critiquing methodology with the meta-theory of existing knowledge-based systems.

Moreover, the proposed framework is here presented in the context of enabling transfer of problem-solving knowledge from the system to the user, i.e. the end-user level of critiquing. Still, the author believes that the proposed methodology incorporated into a framework holding end-user modifiability can be used to not only provide a way of maintenance of knowledge bases, but also be explored in the expert-level as a support in the construction of knowledge-based systems (see e g Håkansson, Widmark & Edman, 2000). By enabling the system to store the knowledge gained from the user, in a way, the system could be "trained" by its user, emphasising the answers the system receives on its analysing and comparative questions. Furthermore, by storing the knowledge provided by the expert or experts and providing questions on the interpretation of found differences, the system designer will be able to optimise the system strategies and provide transparency within the domain. To provide such a helpful assistance to the intricate task of knowledge acquisition would possibly facilitate the development and usage of knowledge based systems for a broader area of utilisation.

Perhaps such a framework will provide an easy, user-oriented and yet adequate way of supporting the transfer of human problem-solving knowledge.

7 References

Aiello, L. Carlucci., Cialdea, M. & Nardi, D (1991) A Meta Level Abstract Description of Diagnosis in intelligent tutoring systems, *Proceedings of the Sixth International PEG Conference*, pp 437, Rapallo (Genova), Italy.

Anderson, J. R. (1993) *Rules of the Mind*, Lawrence Erlbaum, Hillsdale, N.J., USA.

Andresen, B. B. (1991) Knowledge based learning environments - Theories and practice, *Proceedings of the Sixth International PEG Conference*, pp 41-48, Rapallo (Genova), Italy.

Burton, R.R & Brown, J.S. (1982) An Investigation of Computer Coaching for Informal Learning Activities. In Intelligent Tutoring Systems, D.H. Sleeman, J.S. Brown, Eds., Academic Press, London – New York, 1982, pp 79-98. (Referenced in Fischer, G., Lemke, A. C., Mastaglio, T. (1990) Using Critics to Empower Users, *Proceedings of the Conference on Human Factors in Computing Systems (CHI'90).)*

Cawsey, A., Galiers, J., Reece, S., Sparck Jones, K. (1992) The role of explanations in collaborative problem solving, *Proceedings of the ECAI-92 Workshop W15 Improving the Use of Knowledge-Based Systems with Explanations*, pp 97—106, Vienna, Austria.

Clancey, W. J. (1979) Use of MYCIN´s rules for tutoring, *International Journal of Man Machine Studies* Vol. 11, pp 25-49, (Edited version in Buchanan & Shortliffe (1994) *Rule-based expert systems*, pp. 464-489*,* Addison-Wesley).

Clancey, W. J. (1983) Extensions to rules for explanation and tutoring. Artificial Intelligence Vol. 20, pp 215-251. (Edited version in Buchanan & Shortliffe (1994) *Rule-based expert systems,* pp. 531-568. Addison-Wesley).

Davis, R (1976) Applications of Meta-level Knowledge to the Construction, Maintenance, and Use of Large Knowledge Bases. PhD thesis, Stanford University, 1976 (Referenced in Moore, J D., Swartout, W. (1988) *Explanation in expert systems: A survey*, Information Science Report ISI/RR-88-228, University of California, USA.)

Durkin, J. (1994), *Expert Systems, design and development.* Macmillan Publishing Company, Prentice-Hall.

Edman A. (2001) *Combining Knowledge Systems and Hypermedia for User Co-operation and Learning*, Tryck & Medier, Uppsala.

Edman, A. & Hamfelt, A. (1997) A Basis for a System Development Methodology for User Cooperative Systems, *Proceedings of International and Interdisciplinary Conference on Modelling and Using Context, CONTEXT 97, Rio de Janeiro, Brasilia, 1997.*

Edman, A. & Hamfelt, A. (1999) A System Architecture for knowledge based hypermedia. *International Journal of Human-Computer Studies,* 51, pp 1007-1036.

Edman, A., Lindman, Å. & Sundling, L. (1993) Design Issues Concerning Explanations in an Educational Expert System – A Case Study, *Proceedings of the Seventh PEG Conference*, PEG-93, pp 108-115. Edinburgh, Scotland.

Edman, A. & Mayiwar, N. (2003) A Knowledge-Based Hypermedia Architecture Supporting Different Intelligences and Learning Styles, *Proceedings of the Eleventh International PEG Conference, PEG 2003,* (in print), St. Petersburg, Russia.

Eriksson Lundström, J. (2003) *Facing a knowledge-based hypermedia system – Visualising context-based explanations in the user interface,* Thesis for the Degree of Bachelor of Science, Department of Information Science, Uppsala University.

Fischer, G., Lemke, A. C., Mastaglio, T. & Morch, A. I. (1991) The Role of Critiquing in Cooperative Problem Solving, In *ACM Transactions on Information Systems* 9/3, pp 123-151.

Fischer, G., Lemke, A. C., Mastaglio, T. (1990) Using Critics to Empower Users, *Proceedings of the Conference on Human Factors in Computing Systems (CHI'90).*

Forbus, K (1984) An Interactive Laboratory for Teaching Control System Concepts. Report 5511, BBN, Cambridge, MA, 1984 (Referenced in Fischer, G., Lemke, A. C., Mastaglio, T. (1990) Using Critics to Empower Users, *Proceedings of the Conference on Human Factors in Computing Systems (CHI'90).)*

Gardner, H. (1983) Frames of Mind. Basic Books, New York, USA (referenced in Edman, A. & Mayiwar, N. (2003) *A Knowledge-Based Hypermedia Architecture Supporting Different Intelligences and Learning Styles.)*

Gobet, F. & Wood, D. (1999) Expertise, models of learning and computer-based tutoring, *Computers & Education 33, pp189-207.*

Guerlain, S. A., Smith, P. J., Heinz Obradovich, J, Rudmann, S, Strohm, P., Smith, J. W, Svirbely, J. & Sachs, L (1999) Interactive Critiquing as a Form of Decision Support - An Empirical Evaluation, In *Human Factors*, Vol. 41, No. 1, pp 72-89.

Hamfelt, A. (1992) *Metalogic Representation of Multilayered Knowledge*, Ph D Thesis in Computing Science 15, Uppsala University. (Referenced in Edman, A. & Hamfelt, A. (1997) A Basis for a System Development Methodology for User Cooperative Systems, *Proceedings of International and Interdisciplinary Conference on Modelling and Using Context, CONTEXT 97, Rio de Janeiro, Brasilia, 1997.)*

Harrius, J. (1993) *An Architecture and a Knowledge Representation Model for Expert Critiquing Systems*, Thesis for the Degree of Licentiate of Philosophy, Department of Computer and Information Science, Linköping University.

Håkansson, A., Widmark, A. & Edman, A. (2000) The KANAL-system, an intelligent rule editor supporting the knowledge acquisition process. *Proceedings of Swedish AI Society Workshop, SAIS-00.* Linköping, Sweden.

Kraus, L. A., Reed, W. M, Fitzgerald, G. E. (2001) The effects of learning style and hypermedia prior experience on behavioural disorders knowledge and time on task: a case-based hypermedia environment, *Computers in Human Behaviour* 17 pp 125-140.

Langlotz, C. & Shortliffe, E. (1983) Adapting a Consultation System to Critique User Plans, *International Journal of Man-Machine studies*, Vol 19, pp. 479-496.

Liautaud, B. & Hammond, M. (2000) *e-Business Intelligence: Turning Information into Knowledge into Profit*, McGraw-Hill. pp 3-13.

Lucardie, L. (1994) *Functional Object-Types as a Foundation of Complex Knowledge-Based Systems.* Thesis for the Degree of Doctor of Philosophy, Computer Science, Technical University of Eindhoven, pp. 73-111 (Reprinted in *Knowledge Management* (2003) Lucardie, L. & Zelin, J., Uppsala University)

Löwgren, J. (1993) *Human-computer interaction - What every system developer should* know, Studentlitteratur, Lund

Marton, F., Dahlgren, L. O., Svensson, L. & Säljö, R. (1999) *Inlärning och omvärldsuppfattning (Learning and understanding of the surrounding world),* 2nd edition, Prisma, Stockholm.

Means, B. (1994) *Technology and educational reform*: the reality behind the promise. Jossey-Bass Inc. Publishers, San Francisco, CA, USA (referenced in Kraus, L. A., Reed, W. M, Fitzgerald, G. E. (2001) The effects of learning style and hypermedia prior experience on behavioural disorders knowledge and time on task: a case-based hypermedia environment, *Computers in Human Behaviour* 17 pp 125-140.)

Miller, P. (1984) A Critiquing Approach to Expert Computer Advice: ATTENDING, Pitman Publishing Ltd, London, England.

Moore, J D., Swartout, W. (1988) *Explanation in expert systems: A survey*, Information Science Report ISI/RR-88-228, University of California, USA.

Newell, A. (1982) The Knowledge Level, *Artificial Intelligence*, 18, pp 87-127.

Puppe, F. (1999) Meta Knowledge for Extending Diagnostic Consultation to Critiquing Systems, Institute for Artificial Intelligence and Applied Informatics, Wurzburg University.

Silverman, B. G. (1992 a) *Critiquing Human Error – a Knowledge Based Human-Computer Collaboration Approach,* Academic Press, Washington D.C., USA.

Silverman, B. G. (1992 b) Survey of Expert Critiquing Systems: Practical and Theoretical Frontiers, *Communications of the ACM*, Vol 35, No.4.

Steels, L. (1987). The Deepening of Expert Systems. In: Steels, L. and Wielinga, B. (eds) *AI Communications*, vol. 1, nr. 1, pp. 9-17, Amsterdam: IOS Press.

Steels, L. (1990). Components of Expertise. *AI Magazine*, vol.11, nr. 2, pp. 28-49, Menlo Park, CA: The AAAI Press.

Swartout, W. R. & Smoliar, S.W. (1987) On making expert systems more like experts, *Expert Systems*, Vol 4, No 3. pp 196-206.

Waern, Y., Hägglund, S. & Rankin, I. (1988) Expert Critiquing Systems – psychological and computational aspects of generating comments, *Preprints of fourth European Conference on Cognitive Ergonomics, Cambridge, England.*

Wenger, E. (1987) *Artificial Intelligence and Tutoring Systems, Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kaufman, Los Altos, CA, USA.