Agent Modeling in Expert Critiquing Systems

Michael C. Fu and Caroline C. Hayes
Department of Computer Science
University of Illinois at Urbana-Champaign
1514 Beckman Institute
405 N. Mathews, Urbana, IL 61801
mfu@cs.uiuc.edu, hayes@cs.uiuc.edu

Abstract

Expert critiquing systems are a type of human-computer collaborative system in which a computer agent presents reasoned opinions about a human agent's problem-solving process for a given task. The challenge in such systems is to provide timely critiques relevant to the user's focus of attention. A problem with many expert critiquing systems is that their critiques are not always timely or relevant; consequently such systems interfere with problem-solving rather than provide assistance. The problem arises in part from insufficient representations of the human's problem-solving processes. In this paper, we discuss the flexible use of an agent model based on a task-decomposition hierarchy of human experts in a critiquing system called SEDAR. The model differs from previous research efforts in three ways: 1) the structure of the model, 2) the function the model performs in the expert critiquing system, and 3) the influence of the model on communication between the computer and human agents. A prototype of SEDAR was implemented for the flat and low-slope roof design domain. The results of early testing on the prototype show that SEDAR assists users effectively and reduces the error rate.

Introduction

Expert critiquing systems are a type of human-computer collaborative system in which a computer agent helps a human agent generate a solution for a given problem by providing timely and relevant critiques about the human's problem-solving process or product. Advice irrelevant to the human agent's focus of attention or advice presented at inappropriate times during the problem-solving process may distract the human and ultimately decrease the overall quality of the solution. As is noted in (Fischer et al 1993), "...the challenge in building critiquing systems is not simply to provide feedback: the challenge is to say the right thing at the right time." In complex domains, the human agent's focus of attention may shift from one solution subpart to another during the problem-solving process. This may be due to dependencies between the solution subparts or the individual user's problem-solving strategy. However, most expert critiquing systems generally do not attempt to model the focus of attention of the human agent as it changes during problem-solving. As a result, critiques generated by these systems are not always timely and relevant to the human agent.

In this paper, we describe the agent model of SEDAR (the Support Environment for Design And Review), which is used to track the human agent's focus of attention flexibly as it changes during the problem-solving process and to direct the content and timing of critiques from the computer agent. The model consists of an augmented task-decomposition hierarchy developed from protocol analyses of human experts in a problem domain. This "task-based" model differs greatly from past research on agent models in expert critiquing systems in three areas: 1) the structure of the model, 2) the function of the model within the expert critiquing system, and 3) the influence of the model on communication between the human and computer agents. Describing the structure of an agent model involves specifying the composition and organization of the model. Describing the function of an agent model involves specifying how it addresses the problems outlined above: constraining the content of the critiques, and improving reactivity to the user's changing needs. Besides the informational content of the critiques, the computer agent must also communicate the critiques to the human agent effectively. Human-computer communication issues influenced by an agent model include the timing of critiquing strategies and the ordering of information presented to the human agent.

The three issues, function, structure, and influence on communication, are discussed in the context of previous work in the field of human-computer collaborative systems: expert critiquing systems, plan-recognition systems, and intelligent tutoring systems. We then describe our approach to modeling the human agent in SEDAR and contrast it to previous work along the three issues. A prototype version of SEDAR was created for the flat and low-slope roof layout domain,

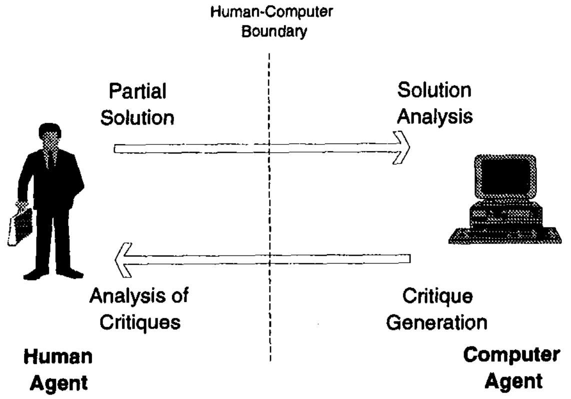

Figure 1: Human and Computer Agents in Expert Critiquing Systems

and an informal evaluation showed that the system is able to help experienced designers reduce the number of errors in their roof layouts.

Agents and Agent Modeling in Expert Critiquing Systems

In expert critiquing systems, the human agent and the computer agent collaborate to achieve a higher-quality solution than could be obtained by either alone. The role of the human agent in an expert critiquing system is to direct the problem-solving activities of the collaborative system and to select and implement critiques generated by the computer agent. The role of the computer agent in an expert critiquing system is to critique the human agent's partial solution and problem-solving process. The critiques serve not only to point out specific problems with the solution, but also to prompt additional reflection by the human. Solution synthesis by the human agent may be tightly interwoven with critique generation by the computer agent, resulting in a continual cyclical flow of information across the human-computer boundary (Figure 1). Each cycle of solution generation, analysis, critique generation, and critique analysis is called a critiquing episode.

Related Work

In this section, previous work involving agent models is summarized. Three classes of systems discussed in the literature that often employ agent models are expert critiquing systems [(Fu 1995), (Mastaglio 1990), (Fischer et al 1993)], intelligent tutoring systems, and plan-recognition systems (Suchman 1987). Unlike expert critiquing systems, intelligent tutoring systems are intended for novices in a domain; the emphasis is on educating the human user rather than improving the quality of solutions. Plan-recognition systems use plan-based agent models. User actions are employed to select a plausible user plan from a library of plans, which is then used to guide the system's behavior. Each type of system will be examined in terms of the three issues of agent structure, function, and influence on communication.

Expert Critiquing Systems

Mastaglio's LISP-CRITIC (Mastaglio 1990) uses a domain and user model to tailor its explanations in terms of LISP programming concepts and functions underlying the critiquing rules of the system's knowledge base. The modeling component consists of four major parts: 1) a representation of what the user knows about LISP concepts and functions, 2) a dialog history of explanations that the user has been exposed to, 3) a preference record of LISP functions and deactivated rules, and 4) a code analysis module that provides statistical data of acceptance and rejection of critiques in the past. Besides the primary function of tailoring explanations, the modeling component is used to determine what subset of the overall rule base to fire for each individual programmer. The model influences communication between human and computer by providing explanations that the user understands.

Fischer's HYDRA (Fischer et al 1993), which critiques kitchen layouts, uses a different model of the human agent. Instead of modeling the human's knowledge, as in LISP-CRITIC, HYDRA elicits and maintains a record of the user's overall goals (or functional specifications) for the kitchen layout. Examples of functional specifications for a kitchen layout would include the size of the family, the right- or left-handedness of the cook, how often cooking is performed in the kitchen, etc. HYDRA uses its "specific" critics to detect inconsistencies between the evolving kitchen layout and the functional specifications. The use of the agent model allows HYDRA to provide critiques relevant to the user's overall goals for the kitchen.

A significant problem with the above agent models is that they do not address the issue of changing focus during the problem-solving process. In complex domains, the user may solve parts of the problem at a time or interleave the synthesis of different problem subparts. Ideally the computer agent should be able to adapt its critiques automatically to this changing context. The model of the user's domain knowledge in LISP-CRITIC allows for the tailoring of explanations for each critiquing episode, but it does not provide a means to adapt the set of critiquing knowledge automatically to the human agent's changing focus of attention. Similarly, while the goal model of HYDRA helps to provide critiques relevant the user's overall goals for the project, it does not provide a framework to adapt to changing user focus during the design process.

Plan Recognition Systems

Plan recognition systems are another type of human-computer collaborative system in which a model of the human user in the computer agent strongly influences system behavior. These systems map observed user behavior onto a set of plans stored in a plan library; the inferred plan represents the system's belief of the user's plans and intentions. The plan is then used to direct the behavior of the system. For example, Suchman (Suchman 1987) describes several experiments involving a plan recognition system for assisting inexperienced users of photocopying machines. Initially, the system asks the user a series of questions about the user's original documents and desired copies. This statement of intent is used to select a single plan from a library of plans. The plan is then presented to the user as a set of procedural instructions. As the user follows the plan, the user changes the state of the photocopying machine in predictable ways; the changes to system state constitute a trace of the user's actions. As Suchman notes, "The design assumption is that by detecting certain of the user's actions, the system can follow her course in the procedure and provide instructions as needed along the way." Significant problems with this approach arise when the system misinterprets the user's actions, when a single plan is insufficient to describe the scope of the user's problem-solving behavior, and when plans are used to predict the future behavior of the human agent. In several experiments involving this plan recognition system and novice users, Suchman found that the inflexibility of plan usage in the system led to numerous severe human-computer communication lapses after only a few incorrect actions were performed by the users.

Intelligent Tutoring Systems

Like expert critiquing systems, intelligent tutoring systems use agent models which influence system behavior greatly. In general, agent models in intelligent tutoring systems are termed as student models to reflect the teacher-student relationship between the computer and human. A typical student model is similar in structure and function to LISP-CRITIC's user model, and is used to track the student's current understanding of the domain. Besides influencing the content of the system's explanations, the model is also used to evaluate the student's specific needs and to help the system's instructional module prepare appropriate individual instruction.

Agent Modeling in SEDAR

The agent model of SEDAR uses an augmented task-decomposition hierarchy to represent the problem-solving process of human experts for a given task. By flexibly tracking the user's changing focus of attention during the problem-solving process, the system is able to generate relevant critiques at critical decision points without constraining the user to a particular solution path, as in the case of the plan-recognition system reported by Suchman. Unlike the agent models based on a user's knowledge of domain facts (e.g. LISP-CRITIC and intelligent tutoring systems), the task-based model of SEDAR represents experts' problem-solving processes for a domain. This process model is flexible enough to model the behavior of many users, not just one user.

Structurally, the model is most similar to the task-decomposition models used in automated design systems [(Brown & Chandrasekaren 1986), (Mittal & Dym 1986)]. Functionally, the model is used to represent the system's beliefs about the problem-solving state of the human user. The model accomplishes this by constraining the set of critiquing knowledge applied in each critiquing episode. The use of the model influences the manner of communication between the human and computer; it determines the time at which various critiquing strategies are used and the order in which information is presented to the user.

Throughout this section, our discussions will be illustrated with examples from the domain of flat and low-slope roof layout for commercial and industrial buildings.

Structure

The task-based model represents the problem-solving process of experienced humans in the problem domain. The process is represented as a decomposition hierarchy of tasks that may be encountered during problem-solving. Besides the task-subtask relationship, the model is augmented with semantic links describing ordering relationships and potential interferences between the tasks. Unlike intelligent tutoring systems, which are intended to interact with mainly inexperienced users, SEDAR is intended to interact with humans with prior problem-solving experience in the domain. Thus the composition and organization of the model reflects this level of domain competence.

Figure 2 shows a portion of the task-based model for the flat and low-slope roof layout domain. The rectangular boxes in the example represent tasks that the user may perform. The task at the left, Roof-Layout, is the most abstract task and represents the overall task of roof layout. It is decomposed into subtasks, Roof-Component-Layout, Equipment-Layout, and Footprint-Layout. These subtasks are further decomposed into their constituent subtasks. The leaf tasks on the far right of the figure (e.g. Drain-Layout, Walkway-Layout, Air-Handler-Layout, etc.) represent the layout of a specific type of roof object or subsystem. The heavy dotted lines, or part-of links, represent the task-subtask decomposition. Before-task links, shown as single-arrowhead solid lines, are possible ordering relationships between tasks, and are drawn from observations of human expert behavior. An example of this is the before-task ordering between Equipment-Layout and Roof-Component-Layout; we observed that humans tend to lay out large, heavy mechanical equipment before other roof components. Interferes-with links, shown as double-arrowhead dashed lines, represent potential interferences among tasks at the same level of abstraction in the problem-solving process.

During system use, the tasks in the model are activated to represent the user's place in the design process and current focus of attention. Each task in the model one of three possible activation states: inactive, active, or focus. In Figure 2, inactive tasks have no shadow, active tasks have a light gray shadow, and focus tasks have a dark gray shadow. Focus tasks are believed to be within the direct focus of the user's most recent actions. A task becomes a focus task under two conditions: 1) the trigger for the task is activated by a user action, or 2) the task is an ancestor of a focus task. For the flat and low-slope roof domain each task has a set of design objects (e.g. drains, air-handling-units, etc.) that serve as its trigger. When the user places a drain on the design, the Drain-Layout task becomes a focus task because one of its triggers is a drain design object. The ancestors of Drain-Layout (Drainage-System-Layout, Membrane-Layout, and Roof-Component-Layout) are also focus tasks. Active tasks are non-focus tasks that: 1) are directly related to a focus task by an interferes-with relation, or 2) were focus tasks previously. These tasks represent relevant considerations outside the current user focus. Finally, inactive tasks are those that have not been addressed yet by the user.

Structurally, the task-based model may be compared to task-decomposition hierarchies found in automated design systems like PRIDE (Mittal & Dym 1986) and AIRCYL (Brown & Chandrasekaren 1986). In the AIRCYL architecture, a hierarchically organized community of design agents represent the hierarchical structure of the artifact (an air cylinder). Each design agent has a repertoire of design plans to accomplish its tasks at its level of abstraction in the hierarchy. The AIRCYL architecture does not deal with interferences between tasks; constraint checking occurs only within the plan steps within each task, and design activity occurs only along the task-subtask relationships; finally, interleaving of the design of different artifact subparts is not supported. Mittal, Dym, and Morarja discuss the hierarchical representation of "design goals" in the PRIDE system for designing paper handling systems within copiers. Each goal in PRIDE represents the design of a small set of parameters that describe some part of the artifact being designed. In SEDAR, the tasks of the agent model are not constrained to represent solely parameterized design, and thus are more general than the design goals of PRIDE.

Function

The agent model serves to influence the behavior of SEDAR in three significant ways. First, the model is used to constrain the knowledge applied during each critiquing episode. This helps to reduce the run-time complexity from the application of critiquing knowledge, and does not constrain the user to a particular ordering of task completion. Second, the model allows the system to be reactive to changes in the users' focus of attention. Finally, user feedback regarding the appropriateness of the advice offered during critiquing episodes is captured in a limited way by providing a means for the user to deactivate individual tasks in the agent model.

Each task in the agent model is associated with a set of relevant critiquing knowledge. For each critiquing episode, only critiquing knowledge associated with focus and active tasks are applied. This approach has two major advantages. First, only critiques relevant to the user's focus of attention are generated; this at least partially addresses the problem in advice-giving systems of saying the right thing at the right time. As the user's focus of attention shifts, the set of focus and active tasks in the agent model and hence the set of knowledge applied during the critiquing episode also change to match the focus shifts. Besides providing critiques directly relevant to the user's current focus of attention, the system also provides critiques relevant to tasks that the user should consider because of potential interferences. Second, since only a limited subset of the critiquing knowledge is applied for each critiquing episode instead of the entire set, run-time complexity due to the application of knowledge should be reduced. One of the goals of a future evaluation of SEDAR will be to determine the amount of reduction of run-time complexity. The technique described here is a much stronger approach to constraining the set of knowledge applied during each critiquing episode than that taken by LISP-CRITIC. It is a more general approach than the rule application method of HYDRA, which applies only rules directly related to a placed design object.

One of the key problems with early plan recognition systems was their inflexibility in adapting to a user's changing focus of attention. In SEDAR, we address this problem by using the task-based model to track the user's problem-solving process rather than to constrain it to a particular sequence of tasks. This is important in expert critiquing systems because the locus of control of the problem-solving process is with the human user. The model is updated whenever new information (knowledge about user actions) becomes available, and the effects of the update on the behavior of the system are immediate.

Besides being adaptive to the user, SEDAR also provides a user adaptation capability for certain aspects of the system. This capability is used to capture user feedback on the appropriateness of the advice offered during critiquing episodes. First, the user may "turn off" tasks in the agent model so that they are not included in the inferencing performed in subsequent critiquing episodes. A similar capability is provided for individual rules in the critiquing knowledge base and individual critiques. This feature allows the user to

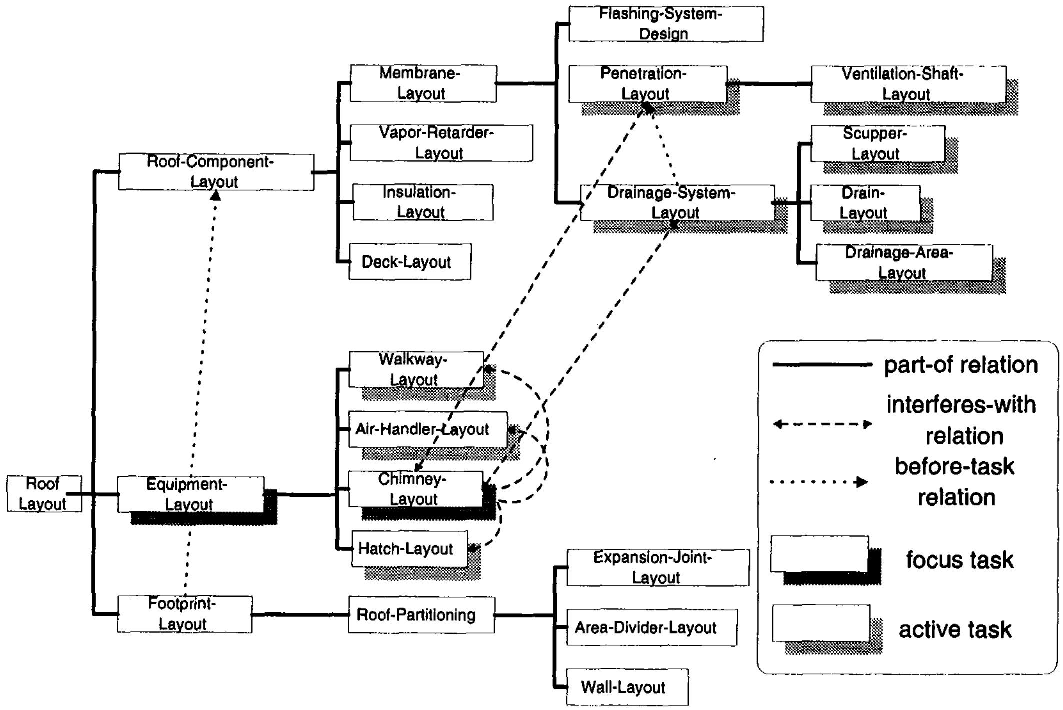

Figure 2: A Portion of the Task-Based Model for Flat and Low-Slope Roof Layout

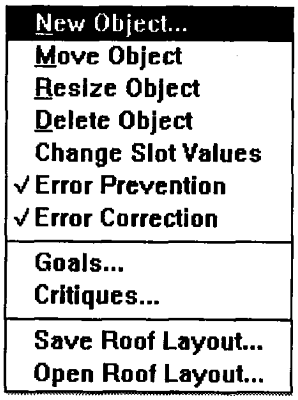

Figure 3: The SEDAR Action Menu

alter the informational content of the critiques generated by the system. Second, each of the various critic agents in the system may be deactivated. Thus the user also has a limited ability to adapt the system to provide specific types of situated advice.

Interagent Communication

Communication between the human and computer agents in expert critiquing systems should embody the four principles of coherent, cooperative dialog described by Grice (Grice 1975): quantity, quality, relation, and manner. A correct quantity of information must be passed between the two participants. Too little or too much information may distract and confuse the receiving agent. Quality refers to the truthfulness of the information being passed. Relation, or relevance, means that the information should be topical within the context of the conversation. Finally, we must consider the manner in which communication occurs. In previous sections, we have discussed how the agent model of SEDAR addresses the issues of quantity and relation; by interpreting the user's focus of attention using the task-based model, only critiques relevant to the user's focus of attention are presented for each critiquing episode. The issue of quality is implicitly dealt with in human-computer collaborative systems; the human agent assumes that the computer agent is providing truthful information. In this section, the manner of communication and its relation to the agent model is discussed. Specifically, the use of the agent model in SEDAR affects the timing of critique presentation and the order in which information is displayed to the user.

Providing critiques at inappropriate times is distracting to the human user. Like an unexpected interjection during conversation, poor timing of critiques serves to interrupt the thought processes of the recipient. (Silverman 1992) provides a conceptual framework for three different timings of critiques: before-task, during-task, and after-task critiques. Before-task

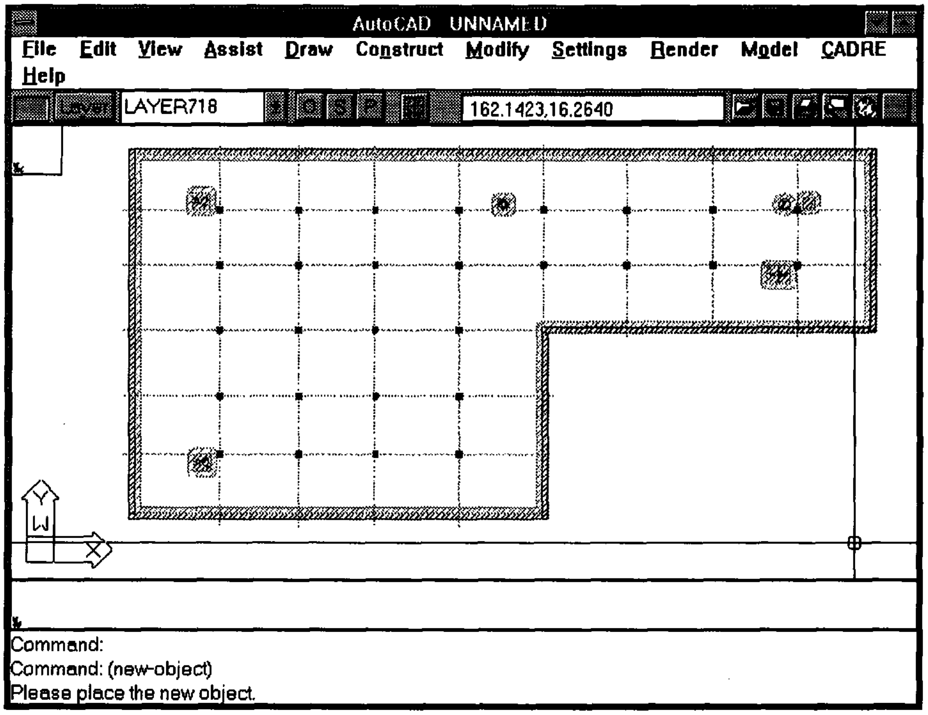

Figure 4: An Example of the Error Prevention Critiquing Strategy

critiques are used to preempt commonly recurring errors before the task is executed by the human agent. If the user may be safely interrupted, then during-task critiques help to eliminate errors while the context of the situation is fresh in the human agent's memory. After-task critics detect errors after the task is completed. These types of critiques may be implemented using the task-based model of SEDAR. For example, suppose that the user decides to add an object (a masonry chimney) to the roof layout by selecting the "New Object..." entry on the SEDAR action menu (Figure 3). The selection of the masonry chimney object triggers an update of the tasks in the agent model. After the agent model is updated, the before-task critic (called the error prevention critic) then computes "off-limits" areas (shown as shaded areas around objects) on the existing design, and displays them directly on the roof drawing (Figure 4). The purpose of this display is to warn the human agent of potentially poor design decisions before the actual placement of the object. In an alternative mode of interaction, the human user may use the task-based model to direct the activity of an after-task critic by specifying a particular solution component from the task-based model for analysis. In Figure 5, the user has selected the Equipment-Layout task, which causes the after-task critic to check the layout of all equipment in the roof field. After the critiques are generated, the user may examine each of the graphical/ textual critiques in turn.

Relevant information presented in an unordered way may also adversely affect the user's understanding of the advice given by the system. SEDAR uses the state of the agent model to order the information presented to the user; critiques resulting from rules associated with focus tasks are presented before those from rules associated with active tasks. Figure 6 shows an example of a critique of a chimney illegally placed within the minimum specified distance (one foot) from another chimney. While the system also has advice regarding the placement of a roof access mechanism, the critique resulting from the focus Chimney-Layout task is given first to the user.

Prototype Evaluation and Discussion

The prototype of SEDAR for flat and low-slope roof layout was evaluated in two experiments. The first experiment was a system usability evaluation, which rated the performance of SEDAR along various usability issues. While the full results of this experiment are reported elsewhere (Fu 1995), one outcome was an informal verification that the functional decomposition of roof subsystems of the task-based model was appropriate and that the system was usable.

The second experiment measured the prototype system's error reduction effectiveness. Two roof layout tasks were presented to five system evaluators at different levels of expertise. Two of the evaluators were experts, having extensive backgrounds in architectural

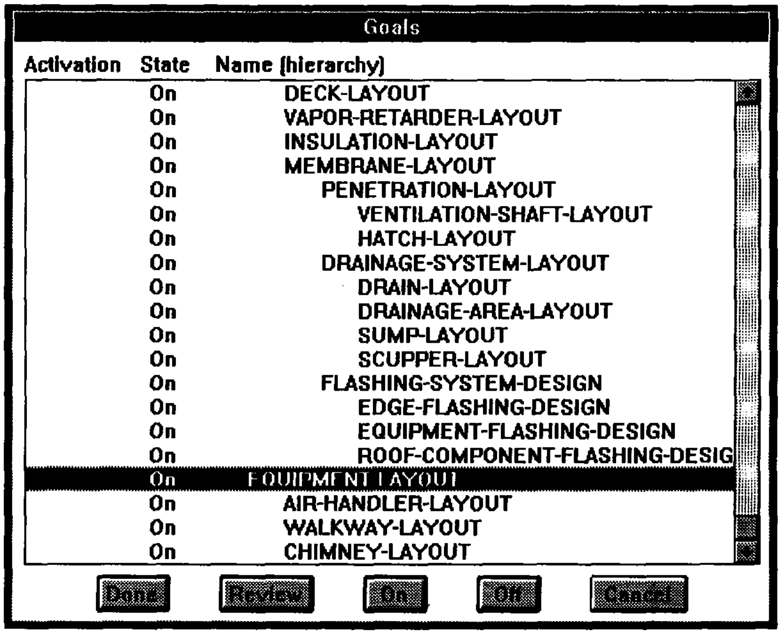

Figure 5: Specifying a Particular Solution Component to Critique

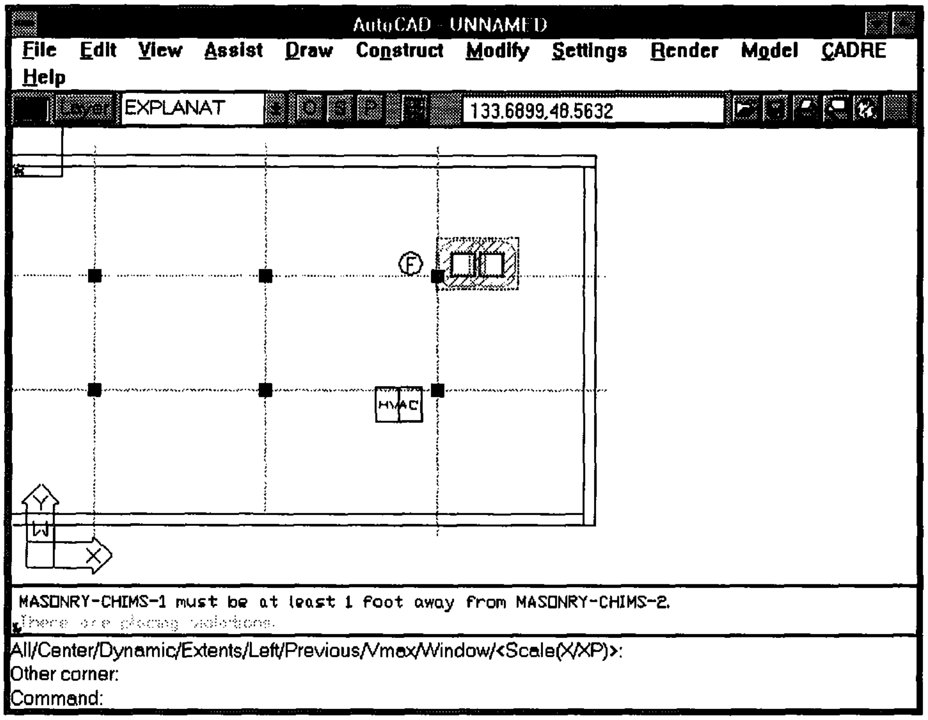

Figure 6: Example of a Critique for a Misplaced Chimney

design. Two evaluators had backgrounds in architectural design but were not practicing architects, and thus were considered intermediates. Finally, the last evaluator was a novice, having no previous architectural design experience. Each evaluator was asked to complete one roof layout without any critic agents activated (no advice offered) and one with all of the critic agents activated (all possible advice offered). The experiment showed that SEDAR helps to reduce the number of errors made by expert and intermediate level designers, but the novice. The first expert made 15 errors, and SEDAR's critics identified 14 of them which the designer then corrected. The second expert made eight errors and SEDAR's critics pointed out seven of them. The system was also able to help the two evaluators of intermediate experience, resulting in similar error reduction rates. One issue that became clear when working with the intermediate designers was that they required additional explanations (supplied by the experimenter) of concepts in roof design to clarify the critiques and suggestions generated by the system. This issue was especially important in the case of the novice evaluator; there were too many gaps in the user's domain knowledge for SEDAR to provide effective support. The novice simply did not understand SEDAR's critiquing well enough for them to be of use.

During the evaluation of the SEDAR prototype, designers commented on additional services that they would have liked for SEDAR to provide. The most commonly requested service was the capability to detect suboptimal arrangements of objects. A significant question, for future work, is how the task-based agent model of SEDAR can be used to influence suggestion generation.

Another suggestion from the evaluators was to add a learning component to SEDAR. The primary desire was for a means of adding knowledge to the domain-specific critiquing knowledge base. We are currently examining methods for effective knowledge acquisition which will enable users to each modify or customize the knowledge base to their individual needs.

Conclusions

In human-computer collaborative problem-solving systems, computer models of the human agent may be employed to improve the interaction between the computer and human agents. Previous work on models for expert critiquing systems has focused on using models of the human agent's factual knowledge of the domain and overall goals for the solution, in order to direct critiquing activities of the system. SEDAR uses a model of the problem-solving process rather than a model of domain knowledge or solution goals. We call this a task-based agent model because it is an augmented task-decomposition hierarchy reflecting the problem-solving process of human experts in the problem domain. This model is used to track the user's focus of attention during problem-solving and determine the subset of critiquing knowledge required for each critiquing episode. As the user works on different parts of the solution, the model is updated to reflect the user's changing focus, and the subset of knowledge applied is changed to match this new critiquing focus. The model is used in a flexible way because it is used to track rather than to direct the user's problem-solving process. The model also influences the manner of communication between the human and computer agents; its framework may be used to direct the timing of various critics and to order the information presented to the user. The results from an informal evaluation of a system prototype for the flat and low-slope roof layout domain are promising, and have illustrated the system's capability to reduce the number of errors made by experienced roof designers.

References

Brown, D., and Chandrasekaren, B. 1986. "Knowledge and control for a mechanical design expert system". *IEEE Computer* 19(7):92–100.

East et al, E. 1995. The Reviewer's Assistant System: System Design Analysis and Description. Technical report, U.S. Army Construction Engineering Research Laboratory.

Fischer et al, G. 1993. "Embedding critics in design environments". *The Knowledge Engineering Review* 8(4):285–307.

Fu, M. 1995. Using a Task-Based Model of Design for Flexible Control in an Expert Critiquing System. Master's thesis, University of Illinois at Urbana-Champaign.

Grice, H. 1975. "Logic and conversation". In *Syntax and Semantics 3: Speech Acts.* New York: Academic Press.

Mastaglio, T. 1990. "User modelling in computer-based critics". In *Proceedings of the 23rd Hawaii International Conference on System Sciences, Volume 3: Decision Support and Knowledge-Based Systems,* 403–411. Los Alamitos: IEEE Computer Society Press.

Mittal, S., and Dym, C. 1986. "PRIDE: an expert system for the design of paper handling systems". *IEEE Computer* 19(7):102–114.

Silverman, B. 1992. *Critiquing Human Error: A Knowledge-Based Human-Computer Collaboration Approach.* New York: Academic Press.

Suchman, L. 1987. *Plans and Situated Actions: The Problem of Human-Machine Communication.* New York: Cambridge Unversity Press.

Zhou et al, H. 1989. "CLEER: An AI system developed to assist equipment arrangements on warships". *Naval Engineers Journal* 101(3):12–137.