TEACHING INFERENCING STRATEGIES

Richard G. Feifer
Michael G. Dyer
Margot Flowers

April 1986
CSD-860085

TEACHING INFERENCING STRATEGIES

Richard G. Feifer

Michael G. Dyer

Margot Flowers

Artificial Intelligence Laboratory
UCLA Computer Science Department

and

Center for the Study of Evaluation
UCLA Graduate School of Education

WORD COUNT: 3173

TOPIC AREA: Intelligent Tutoring

KEY WORDS: Inferencing, Graphical Environments, Teaching, Learner Models

ABSTRACT

Sherlock is an intelligent tutoring system designed to teach effective reasoning. Our work is based on the assumption that inferencing is an important prerequisite skill for using the higher level learning strategies being developed to aid in comprehension.

The current implementation is a discovery learning game in which the player manipulates facts and rules in a graphic format. The game uses its own inferencing mechanism to provide feedback.

The next steps will be the addition of an intelligent coach and the empirical validation of the game as an educational tool. We propose the use of Sherlock to study and improve human reasoning.

RATIONALE

Why is learning more difficult for some than for others? One working assumption has been that unsuccessful learners lack the prerequisite learning skills (Wittrock 1974; 1978). Teaching these learning skills, then, could make them successful learners (Wittrock, 1977). This approach has worked for some but not for others. For the latter group, their deficit may exist at an even higher level than that of the strategies being taught. As a result, they are incapable of applying these new learning strategies.

This claim is based on a belief that using these learning strategies is contingent on knowing how to reason effectively. To that end we have designed a computer simulation game intended to teach people how to think and reason in a clear and well-structured manner.

Sherlock consists of a graphic environment in which the player manipulates icons representing facts, beliefs, and rules, attempting to construct a "support graph" for a given statement.

BACKGROUND

Efforts to improve education have concentrated on improving instruction through instructor training, better media and curriculum development. With the cognitive movement has come an awareness of the learner as a central agent in the learning process. In this view "learner training" becomes at least as important as instructor training.

In this respect, important efforts have been in the area of developing cognitive learning strategies. This work involves the following steps:

1. Determine the underlying processes necessary for effective learning.

2. Develop explicit strategies which accomplish these processes.

3. Train the learner in the effective use of the strategies.

One aspect of learning, to which these three steps have been applied, is the integration of new knowledge into the learner's existing schema (Wittrock, 1977). As a result of the first step, learner generation has been identified as an important underlying process. Generation refers to the learner discovering or creating his/her own meaning in the to-be-learned material (Wittrock, in press). The second step has identified a number of strategies which might facilitate this process. These include: outlining, paraphrasing, summarizing, generating questions, and conceptual mapping (Doctrow, Wittrock & Marks, 1978; Anderson, 1979). The third step, training the learner to use these strategies, will be the focus of this project.

All of these techniques represent aids for structuring inferencing on the part of the learner. To use them the learner must first be capable of making and recognizing appropriate inferences. Sherlock is an approach to the theoretical and practical problems inherent in teaching inferencing.

PROBLEMS

Telling a learner that inferencing is a strategy for understanding is useless if he/she lacks the skills needed to make inferences. For learners in this category we must apply the three steps described above, but now to the task of inferencing:

1. Determine the processes underlying making inferences.
2. Develop explicit strategies which accomplish these processes.
3. Train the learner in the effective use of these strategies.

For the first step, an effective means must be found to observe inferencing strategies. Unfortunately most people cannot readily articulate the reasoning behind inferences they make. Therefore a means must be found to facilitate observation. Once the process can be observed, step two is to identify which strategies typify the sucessful inferencer. It might be observed, for example, that the sucessful inferencer works backwards from the goal, rather that forward from the facts. The final step is to developing the technology for teaching these strategies. This will necessitates providing the student an opportunity for practicing identified strategies with feedback. While traditional computer-assisted instruction can provide immediate feedback, its accuracy is limited by the range of responses the programmer was able to anticipate.

SHERLOCK

Sherlock is a computer program which incorporates techniques of aritificial intelligence (AI) to teach inferencing strategies in a game format. The program provides an environment for observing the processes of inferencing by adding a physical dimension to the task. This permits not only observation of the learner making inferences, but handles some of the cognitive overhead, making protocols feasible. By comparing the observations of successful and unsuccessful players of the game, strategies of more effective inferencers can be defined.

In its current version Sherlock trains inferencers by providing practice and immediate feedback. In addition to being told the sub-skills involved, it is important that the learner have an opportunity to practice the skills while receiving informative feedback (Bandura, 1968). A traditional strength of computer-assisted instruction has been its ability to provide immediate feedback. In order for the computer to provide feedback, however, it must first be able to evaluate the learner's performance. In conventional CAI this is done through some form of pattern matching. Because effective inferencing strategies require that the learner make a personal and creative reaction to the learning task, pattern matching is an inadequate tool.

The use of the present version of Sherlock to improve learning skills is based on several assumptions:

- The inferencing skills necessary to play the game are parallel to those needed in more traditional learning tasks.

- The game will motivate learners to practice making inferences.

- The practice with feedback which the game provides will improve the individual's ability to make inferences.

Sherlock presents the user with a screen containing a goal (an icon containing a belief, concept or idea to be supported) and ideas (icons representing facts, beliefs, domain specific rules, and general inference rules). Using a mouse the player can move an icon to another part of the screen, create a link between two icons, break a link, or ask for a hint.

When icons, which together will allow an inference, have been linked, an icon representing that inference is automatically generated. The player's goal is to build a support-graph (Toulmin, 1958; 1979; Flowers, McGuire & Birnbaum, 1982; Flowers, 1985) which represents a valid chain of reasoning supporting the goal icon.

```
                +------------+
                | John will  | Conclusion
                |    eat     |
                +------------+
                      ^
                      |                +--------------------+
             +------------+            | If X has food      |
             | Support    |<-----------| and X is hungry... |
             |   Box      |            | then X will eat    |
             +------------+            +--------------------+
                 ^          ^                 Rule
                /            \
     +------------+          +------------+
     | John is    |          | John has   |
     | hungry     |          |   food     |
     +------------+          +------------+
                    Facts
```

figure 1

SAMPLE SESSION

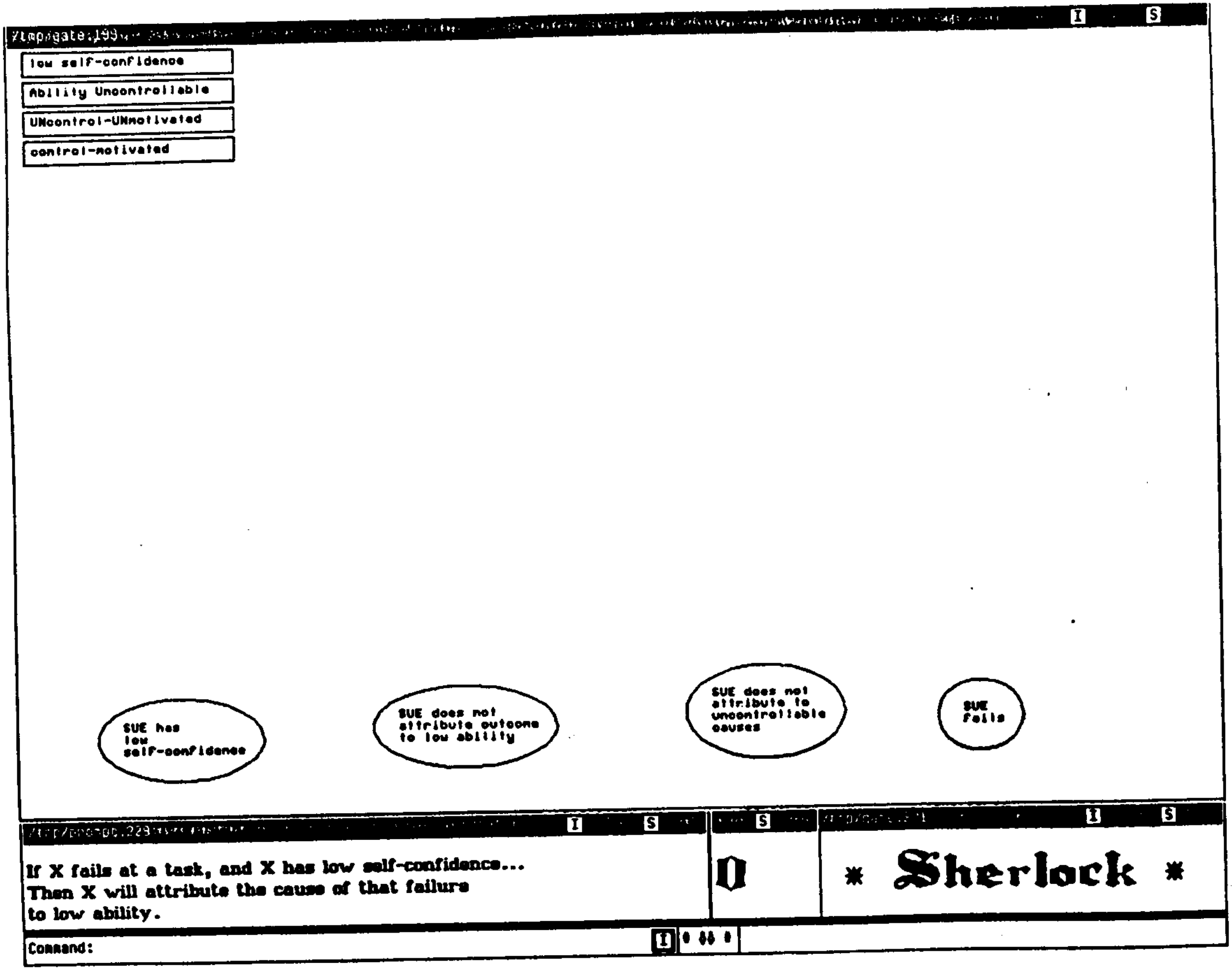
Screen 1

We are currently testing Sherlock on a subset of rules from the domain of attribution theory (Weiner, 1979; 1980). In the upper left-hand corner of the screen are the rules which are available to the learner. The full text of a rule can be printed in the lower left-hand window by pointing the mouse at the desired rule.

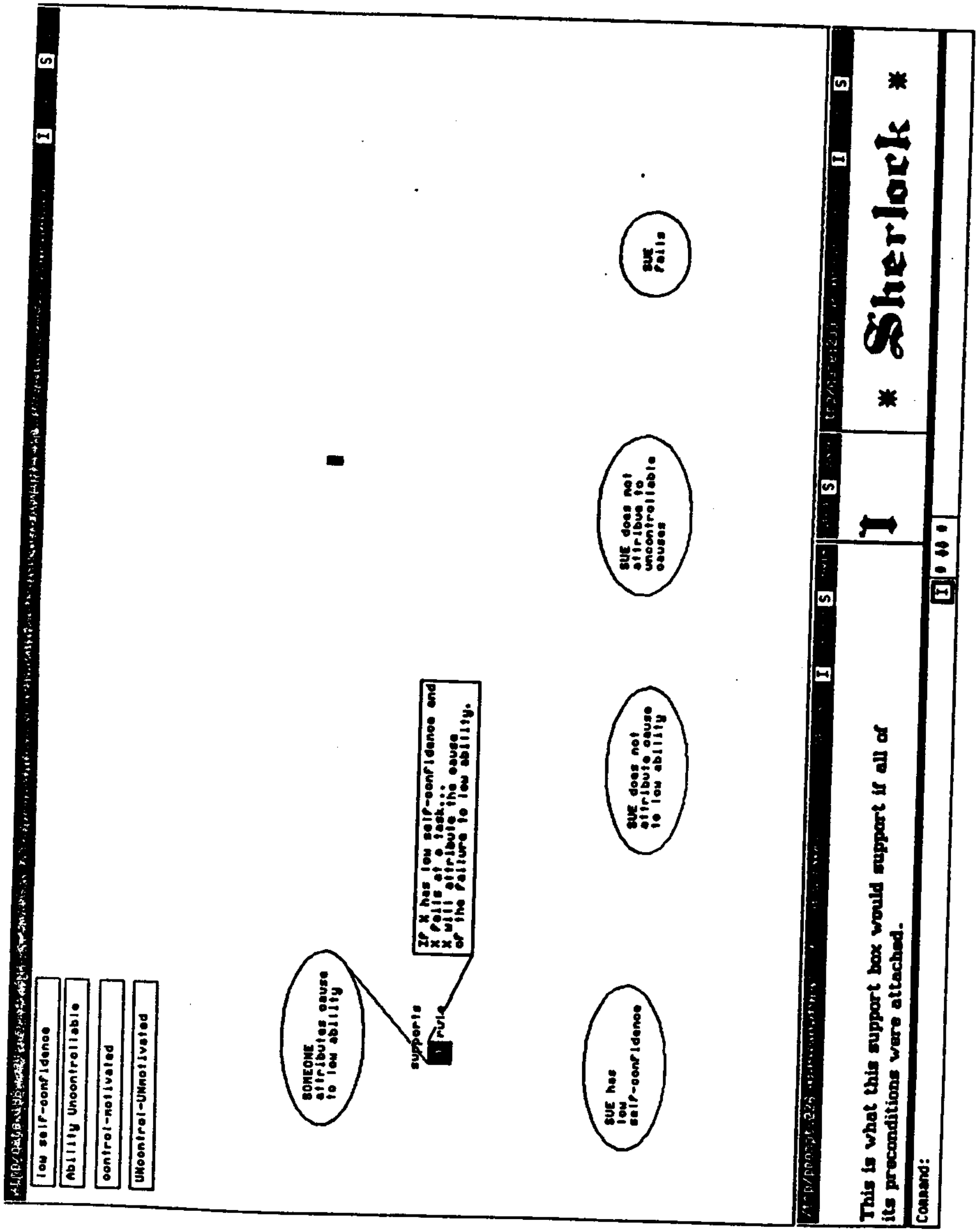
The ovals at the bottom of the screen are the facts which the learner starts with. The goal in this particular session is to determine whether or not Sue will be motivated according to attribution theory.

Screen 2

The player has decided to start with the rule "Low Ability." Using the mouse, a copy of this rule has been requested. The copy contains the full text of the rule, "If


low self-confidence
Ability Uncontrollable
UNcontrol-UNmotivated
control-motivated
SUE has low self-confidence
SUE does not attribute outcome to low ability
SUE does not attribute to uncontrollable causes
SUE fails
If X fails at a task, and X has low self-confidence...
Then X will attribute the cause of that failure
to low ability.
0
* Sherlock *
Command:

SCREEN 1


low self-confidence
Ability Uncontrollable
control-motivated
UNcontrol-UNmotivated
SOMEONE attributes cause to low ability
supports
rule
If X has low self-confidence and X fails at a task... X will attribute the cause of the failure to low ability.
SUE has low self-confidence
SUE does not attribute cause to low ability
SUE does not attribue to uncontrollable causes
SUE fails
This is what this support box would support if all of its preconditions were attached.
Command:
* Sherlock *

SCREEN 2

X has low self-confidence and X fails at a task, then X will attribute the cause of the failure to low ability." The copy is also automatically attached to a support box, so that the learner may begin building. It is the support boxes which actually make inferences, based on the rule and fact or facts which are connected to it.

In an earlier version of Sherlock, inferences were built directly from rules. Support boxes have been added for several reasons. First, it permits the use of the same rule as support in several different places. Conceptually more important, however, is that it will eventually enable us to manipulate a rule as the product of a belief (Flowers, McGuire & Birnbaum, 1982).

In this case the learner has asked the support box "What if?," or what inferences would be possible if the appropriate facts were connected. This has led to the support box generating a new fact icon, "SOMEONE attributes the cause low-ability." This icon can now be used as any other fact for building. The "What if?" capability give the player the option of starting a graph at any point, without having to build (given facts) from the ground up.

Screen 3

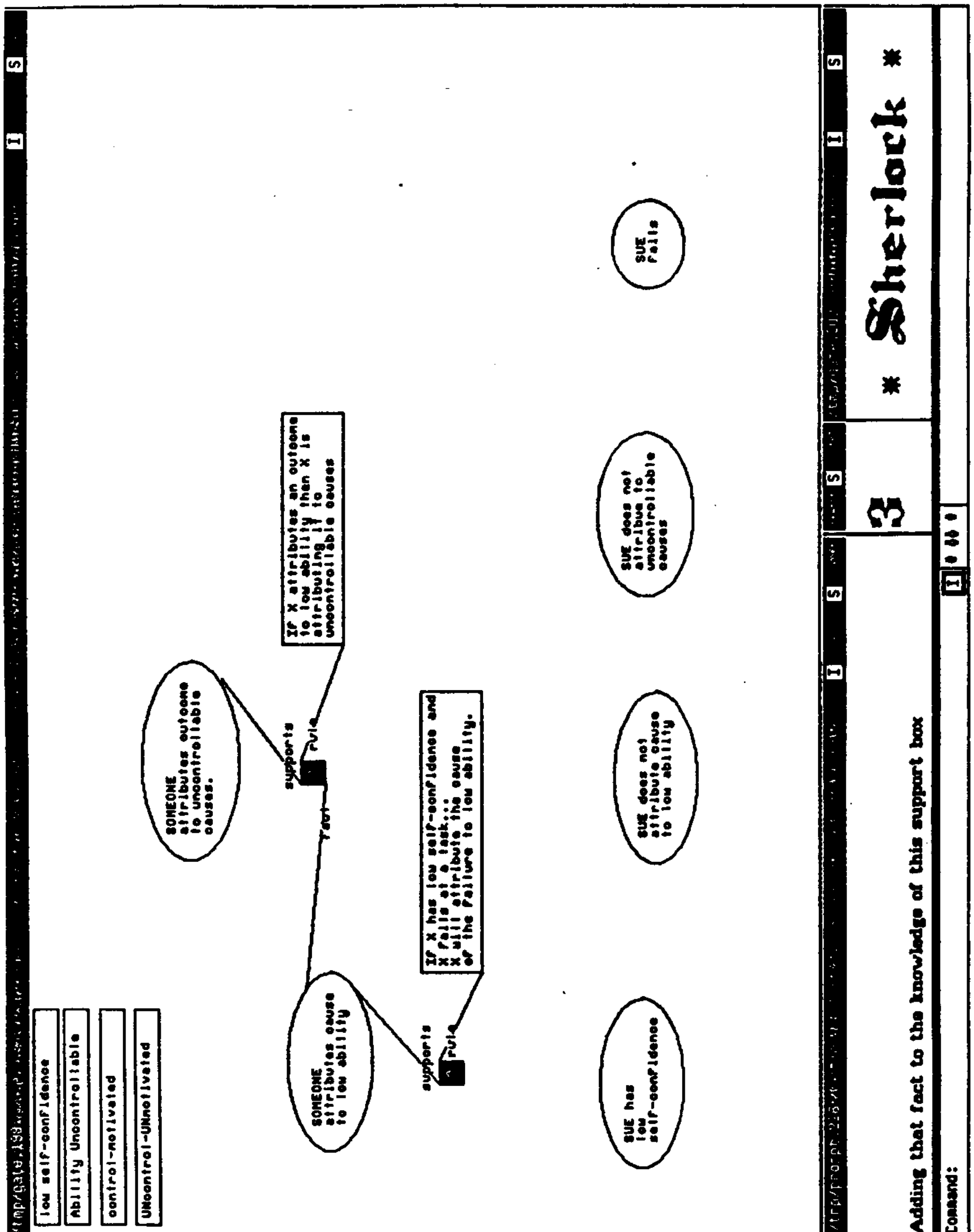
The player recognizes that this new fact is the antecedent for the rule, "Ability Uncontrollable." By connecting this fact and rule to a support box Sherlock is able to infer that, "SOMEONE attributes outcome to uncontrollable causes."

Screen 4

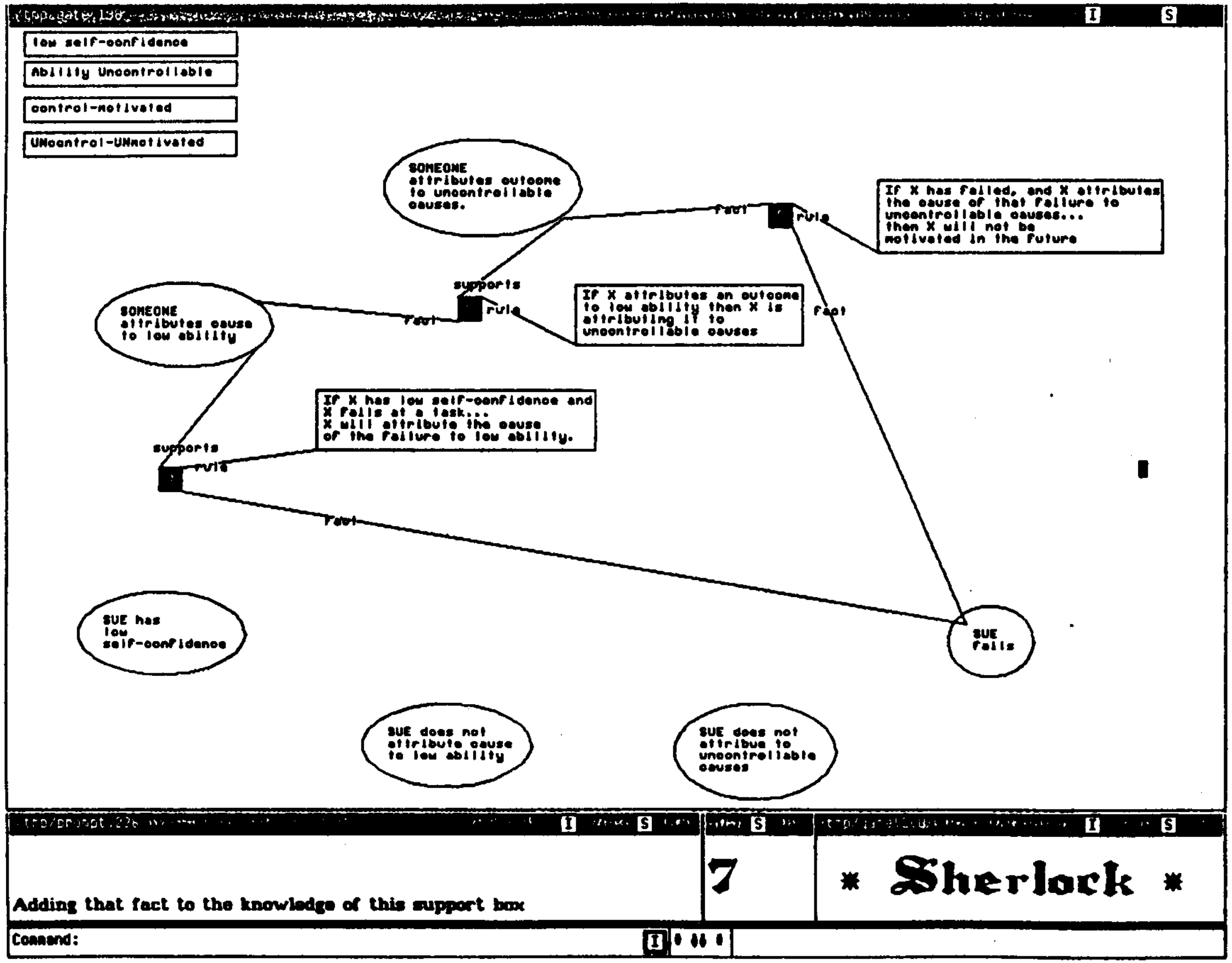
The learner now connects this new fact and the given fact that "SUE failed" to a support box which is connected to the rule, "Uncontrol-Unmotivated." Currently unresolved is the question of whether or not this should allow an inference. It is clear that attaching "SUE failed" and "FRED attributes outcome to uncontrollable causes" should not lead to an inference. This player, however, felt that the general SOMEONE should unify with a specific. Sherlock makes no inference at this point.

Screen 5

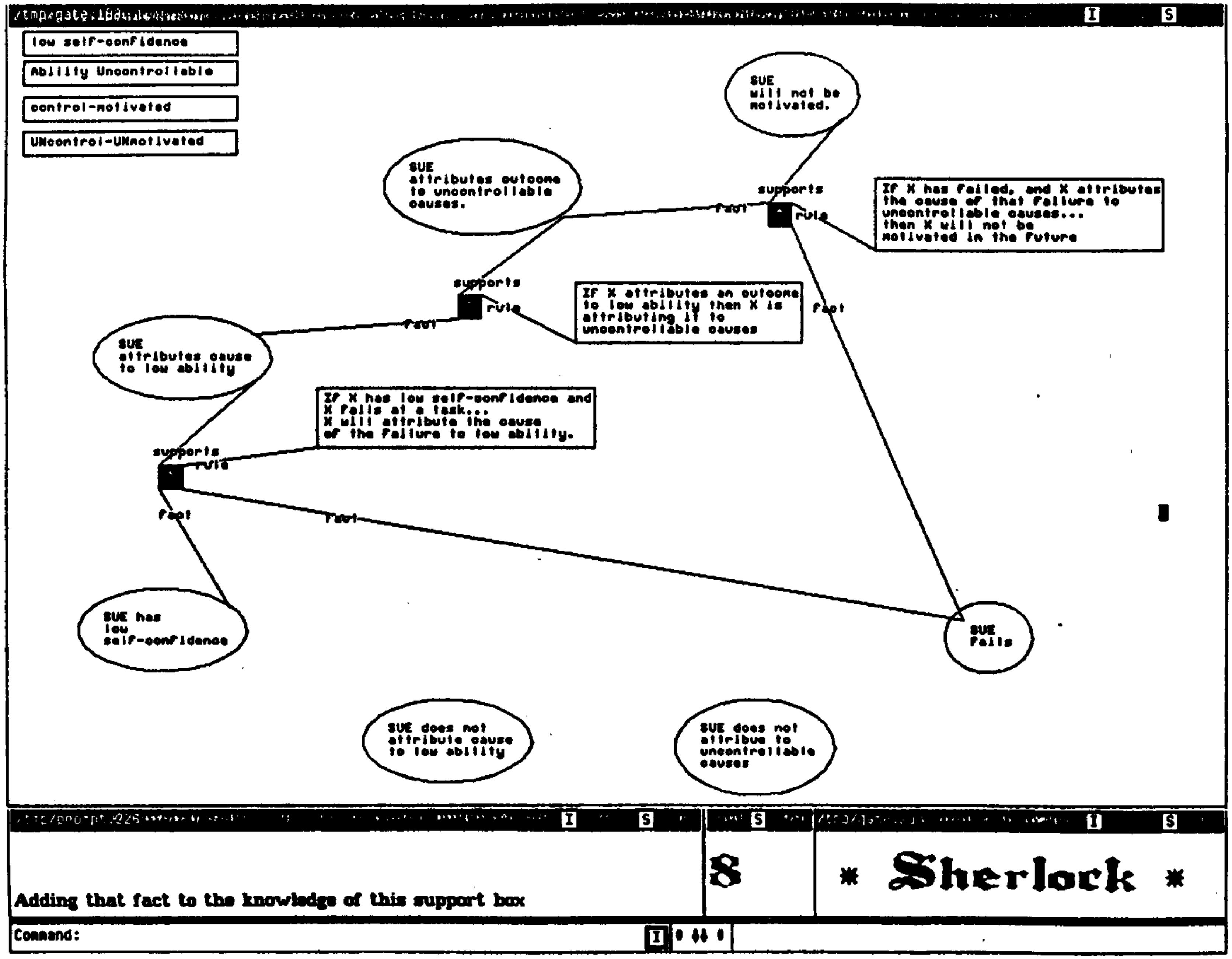
The learner now realizes that the antecedent for the original "Low Ability" rule can be satisfied with two of the given rules. When these are attached, Sherlock replaces the general fact about SOMEONE with a new fact about SUE. Since the old fact had been connected to another support box, the new fact replaces it as input to the support box. Sherlock


low self-confidence
Ability Uncontrollable
control-motivated
UNcontrol-UNmotivated
SOMEONE attributes cause to low ability
supports
rule
SOMEONE attributes outcome to uncontrollable causes.
supports
rule
fact
If X attributes an outcome to low ability then X is attributing it to uncontrollable causes
If X has low self-confidence and X fails at a task... X will attribute the cause of the failure to low ability.
SUE has low self-confidence
SUE does not attribute cause to low ability
SUE does not attribue to uncontrollable causes
SUE fails
3
* Sherlock *
Adding that fact to the knowledge of this support box
Command:

SCREEN 3



low self-confidence
Ability Uncontrollable
control-motivated
UNcontrol-UNmotivated
SOMEONE attributes outcome to uncontrollable causes.
If X has failed, and X attributes the cause of that failure to uncontrollable causes... then X will not be motivated in the future
fact
rule
supports
SOMEONE attributes cause to low ability
If X attributes an outcome to low ability then X is attributing it to uncontrollable causes
If X has low self-confidence and X fails at a task... X will attribute the cause of the failure to low ability.
SUE has low self-confidence
SUE fails
SUE does not attribute cause to low ability
SUE does not attribue to uncontrollable causes
Adding that fact to the knowledge of this support box
7
* Sherlock *
Command:


low self-confidence
Ability Uncontrollable
control-motivated
UNcontrol-UNmotivated
SUE will not be motivated.
SUE attributes outcome to uncontrollable causes.
supports
fact
rule
If X has failed, and X attributes the cause of that failure to uncontrollable causes... then X will not be motivated in the future
If X attributes an outcome to low ability then X is attributing it to uncontrollable causes
SUE attributes cause to low ability
If X has low self-confidence and X fails at a task... X will attribute the cause of the failure to low ability.
SUE has low self-confidence
SUE fails
SUE does not attribute cause to low ability
SUE does not attribue to uncontrollable causes
Adding that fact to the knowledge of this support box
* Sherlock *
Command:

SCREEN 5

then replaces the second SOMEONE with SUE. Once again, the old fact had been input to a support box, so the new fact replaces it as input. It is now possible for the third support box to infer that "SUE will not be motivated."

IMPLEMENTATION

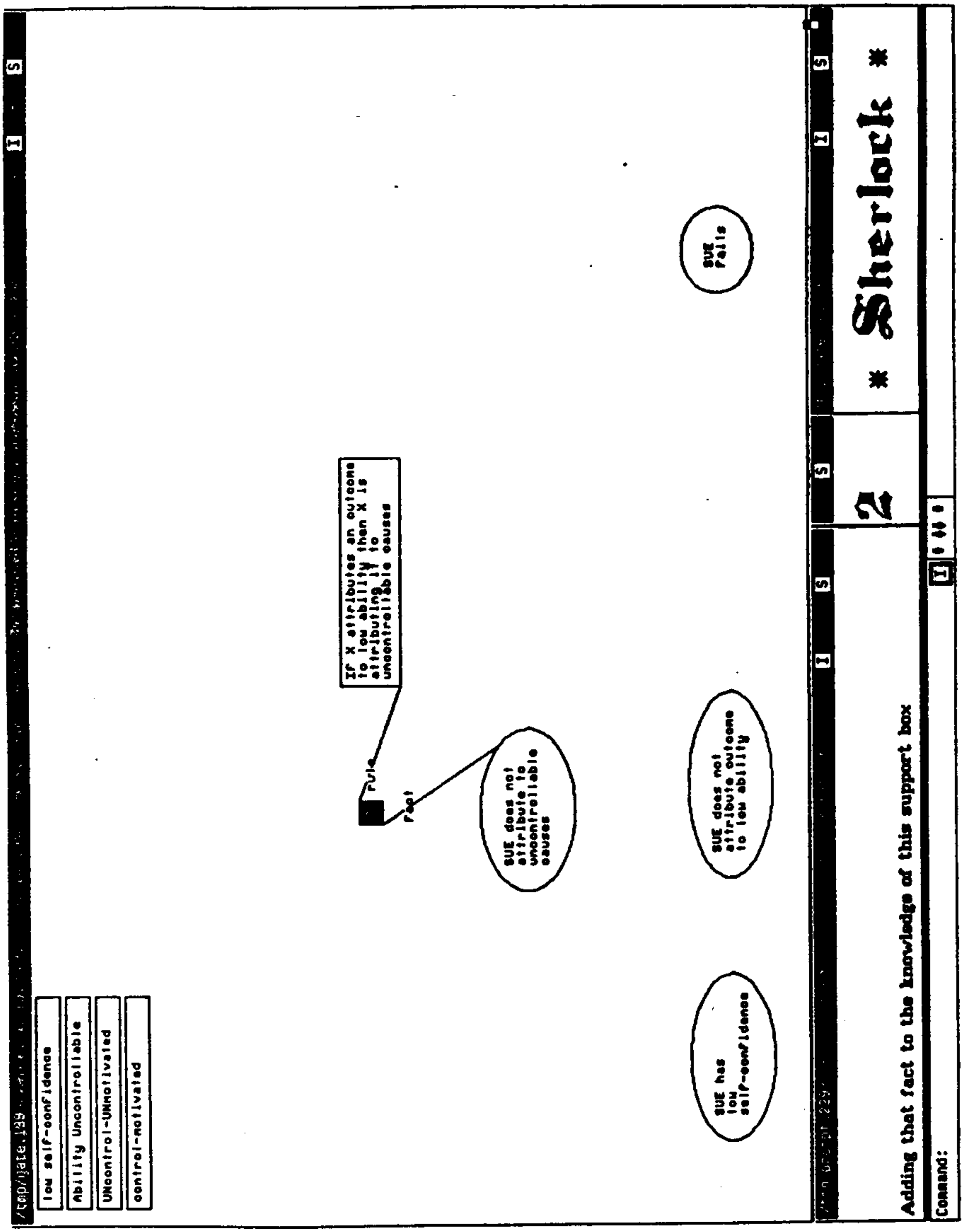
At the heart of Sherlock is a system for representing ideas and their relationships to each other. Each idea is an object, a programming entity which the rest of the program can interact with. For example, one object might represent the fact that "John is hungry." The object has slots, each of which can be filled with a specific kind of information. One slot in a fact object is the logical assertion which the object represents. Another slot contains the name of any objects which support it (ie: the inference connections which led to its creation).

A graphic interface, GATE (Mueller and Zernik, 1984), is then used to provide the learner with a visual display of the internal representation. The fact object just described would be represented by an icon on the screen which contains the text "John is hungry." Also if the supported-by slot had anything in it, the display will show a link between the John-is-hungry icon, and the icon which supports it. The interface can work in both directions, that is when the user makes a change on the graphic display, it is reflected in the internal representation.

The third leg of the program is an inferencer using Tlog (Read, 1984). Tlog provides PROLOG facilities within a LISP environment. The relevent facility here is the ability to generate any conclusions which can be derived from a given set of assertions.

In conventional CAI the correctness of a response is determined through some sort of matching against a list of acceptable answers which the programmer supplies. The player is thus limited to any answers which the author was able to anticipate. An inferencer, on the other hand, tries to emulate human reasoning in determining whether a response in Sherlock is correct. Theoretically, any response which reflects valid reasoning will be recognized by the inferencer.

Unfortunately most PROLOG implementations (including Tlog) will not recognize any valid human reasoning. For example, a powerful and under-used inference rule is Modus Tollens (Johnson-Laird & Wason, 1977). In screen 6 the learner is attempting such an inference, the inference must be extended to recognize this as valid and generate "SUE


/tmp/gate.139
low self-confidence
Ability Uncontrollable
UNcontrol-UNmotivated
control-motivated
rule
fact
If X attributes an outcome to low ability then X is attributing it to uncontrollable causes
SUE does not attribute to uncontrollable causes
SUE has low self-confidence
SUE does not attribute outcome to low ability
SUE fails
2
* Sherlock *
Adding that fact to the knowledge of this support box
Command:

SCREEN 6

does not attribute the cause to low-ability."

Not only does the use of an inferencer free the learner from trying to find the answer or answers which the programmer was seeking, it frees the programmer from specifying parameters for every possible juncture. By separating the process (the inferencer) from the domain, each can be manipulated without affecting the other. The game can be applied to new domains by substituting only domain knowledge.

Sherlock is a meta-game allowing a number of play options within the same environment. One group of variations involve presenting iconically to the player any two items of a facts-rule-inference set, and asking them to provide the third:

> What rule R would allow the player to draw conclusion C using facts F1..Fn?
> (ie: What rule would allow you to infer that John will eat from the facts that there is food and John is hungry?)
>
> What facts F1..Fn are required before the player can draw conclusion C using rule R?
> (ie: What facts would allow you to conclude that John will eat using the rule that if X is hungry and there is food, X will eat?)
>
> What does rule R let the player infer from facts F1..Fn?
> (ie: What can you infer from the facts that John is hungry and there is food, using the rule that if X is hungry and there is food, X will eat?)

COACHING

The next step in the development of Sherlock will be the addition of intelligent coaching capabilities. Examples of such an approach include West (Burton & Brown, 1976; 1979) and Wumpus (Goldstein, 1979). This approach provides the opportunity for learner-directed activity, with the coach providing guidance when needed.

To provide this guidance West and Wumpus maintain learner models containing a current skill mastery estimate. The coach decides when to intervene based on input from the learner model and its pedagogic heuristics. The major limitation of both programs is that they operate in very limited worlds, as circumscribed by the games to which they were applied. These constraints were necessary to avoid the combinatoric problems in representing a more natural world.

The designers of the above programs limited the role of the coach, evidently to maintain the game spirit (Burton & Brown, 1979). The coach never asked the learner why he made a particular move, because it would interrupt play. Since the theme of our game is reasoning the coach can ask why without violating the spirit. Similarly, the output of the West coach was not developed beyond an acknowledgement of an appropriate time and topic for tutoring. We hope to combine the work done in cognitive modelling with work in the area of teaching meta-cognitive strategies (Wittrock, 1974).

The coach in Sherlock will look for inference bugs and inference strategies. The inferencer will be used to provide evidence regarding bugs which seem to exist in the learner's attempts to build a graph. While protocol analysis will later be used to develop a taxonomy of bugs and strategies, we have begun by speculating a small subset of each for testing. The subset of bugs includes;

- Denying the Antecedent. Assuming that because the antecedent of a rule is false that the consequent must also be false.

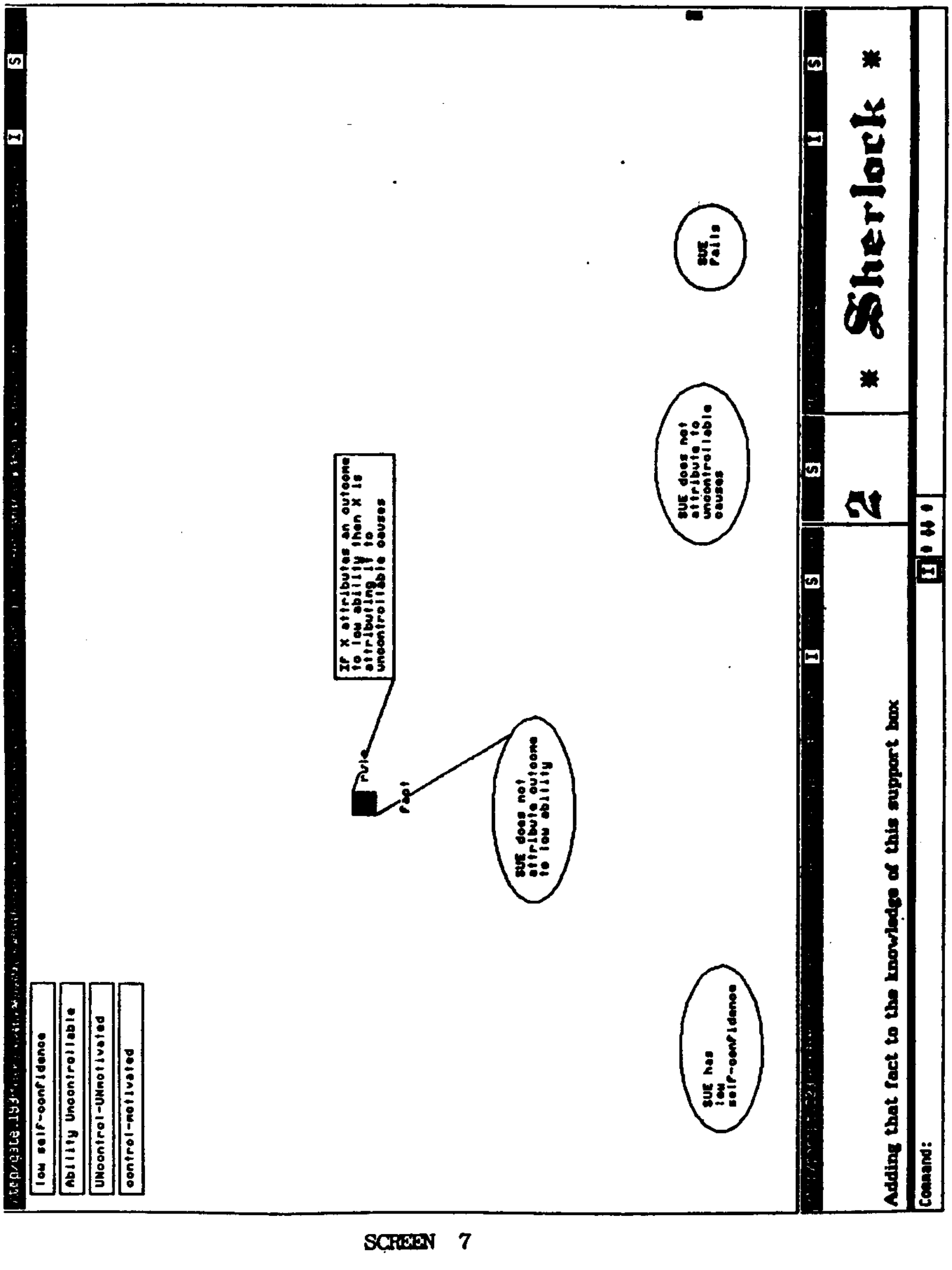
- Affirming the Consequent. Assuming that because the consequent of a rule is true that the antecedent must also be true.

- Biased Assumption. Assuming that something which "sounds right" is true.

Assume in Screen 7, for example, that the learner has just indicated that he/she believes that he/she has enough connected to the support box to make an inference. Sherlock determines that an inference is not possible at this point, so it turns to a set of inference bug recognizers to determine if one of them could be responsible for the misunderstanding.

In this case it is determined that the inference bug, "Denying the antecedent," would have led to an inference. If this is determined to be the most likely bug, the information would be supplied to the student model.

Similarly the coach would watch for evidence of reasoning strategies which are being used. The inferencer will be extended to determine specific strategies which would be fruitful at a given time. As in West, the coach would use an intersection between missing strategies and currently useful strategies to determine the content and


low self-confidence
Ability Uncontrollable
UNcontrol-UNmotivated
control-motivated
SUE has low self-confidence
SUE does not attribute outcome to low ability
Fact
Rule
IF X attributes an outcome to low ability then X is attributing it to uncontrollable causes
SUE does not attribute to uncontrollable causes
SUE Fails
2
* Sherlock *
Adding that fact to the knowledge of this support box
Command:

SCREEN 7

timing for coaching.

Our current subset of strategies includes;

- o Forward Reasoning. Use the given facts to create any possible inferences, and build up from there.

- o Backward Reasoning. Find a rule which would lead to the goal and support it through facts created using "What if?"

- o Macro Chains. Find rules whose consequents are the antecedents for other rules and link them using "What if?"

- o Reasoning by Reduction. Find a rule which would disprove the goal and find a contradiction.

EMPIRICAL VALIDATION

To establish the validity of using Sherlock to improve an individuals ability to make inferences, we are designing a series of empirical studies.

In the first phase the game will be refined, and the feasibility of the design explored. Early versions of the game will be field tested with high school students. Several domains will be used at this point with the goal of optimizing motivation and age-level appropriateness. On the basis of this field test, revisions will be made to the game environment and selections made as to the most effective variations for instructional purposes. There will also be an attempt to identify the strategies used by the more successful players of the game. These can then be incorporated into the game's hinting mechanism.

The second phase will involve testing the inferencing ability of student who have played the game against a control group. Tests for measuring the ability to make inferences will be chosen after a careful review of existing instruments. Considered will be instruments specifically labeled as inferencing tasks (ie: Inference Section of the California Test of Mental Maturity) as well as reading tests which separate inferencing from other aspects of comprehension.

In addition to the pre and post test measures, the game program will be designed to keep statistics on the ongoing success of each player. Statistical analysis within the experimental group of correlation between success at the game and pre-test scores will indicate the overlap between skills needed for each task. Analysis of the correlation between exposure to the game and post-test scores will provide support for the use of the game in its present form to improve inferencing ability.

SUMMARY

Providing the less able learner with missing strategies involves more than identifying the relevant strategies. Prerequisite skills for the strategies must also be identified. Assuming inferencing to be such a skill, a means must be found to provide immediate and effective feedback for the wide range of inferences a student might make.

Sherlock has provided an answer to this problem. Further it is providing a tool for better understanding the inferencing process and specific reasoning strategies used by more successful learners. Incorporating this knowledge into an intelligent coach, Sherlock will have the potential to go beyond providing practice with feedback and actually train individuals to be better reasoners.

REFERENCES

Anderson, T.H. Study Skills and Learning Strategies. In O'Neil, H.F. and Spielberger,C.D. (eds.). Cognitive and Affective Learning Strategies. New York, New York: Academic Press, 1979.

Burton,R. and Brown, J.S. A Tutoring and Student Modelling Paradigm for Gaming Environments. In Coleman, R and Lorton,P.Jr. (eds.). Computer Science and Education. ACM SIGCSE Bulletin, 1976, 8(1), 236-246.

Burton,R. and Brown, J.S. An Investigation of Computer Coaching for Informal learning Activities. International Journal of Man-Machine Studies, 1979, 11, 51-77.

Doctorow, M.J., Wittrock,M.C., and Marks,C.B. Generative Process in Reading Comprehension. Journal of Educational Psychology. 1978, 70, 109-118.

Dyer, M.G. and Flowers,M. Really Arguing With Your Computer in Natural Language. National Computer Conference: Las Vegas, Nevada, 1984.

Flowers,M., McGuire,R. and Birnbaum,L. Adversary Arguments and the Logic of Personal Attacks. In W.G. Lehnert and Ringle (eds.). Strategies for Natural Language Processing. Hillsdale, N.J.: Lawrence Erlbaum Assoicates, 1982.

Flowers,M. Memory-Based Reasoning: A Computer Model of Human Reasoning. Ph.D. Dissertation, Yale University, forthcoming.

Goldstein, I.P. The Genetic Graph: A Representation for the Evolution of Procedural Knowledge. International Journal of Man-Machine Studies, 1979, 11, 51-77.

Johnson-Laird, P.N. and Wason, P.C. Thinking: Readings in Cognitive Science. Cambridge, England: Cambridge University Press., 1977.

Mueller, E. and Zernik, U. GATE Reference Manual, UCLA AI Lab, Tools Note 6, 1984.

Reed , W. Introduction to Tlog. UCLA AI Lab, Tools Note 4, 1984.

Toulmin, S.E. An Introduction to Reasoning. New York, New York: Macmillan Publishing Co., 1979.

Toulmin, S.E. The Uses of Argument. Cambridge, Mass.: Cambridge University Press, 1958.

Weiner, B. A Theory of Motivation for Some Classroom Experiences. Journal of Educational Psychology, 1979, 71, 3-25.

Weiner, B. Human Motivation. New York: Holt, Reinhart & Winston, 1980.

Wittrock,M.C. A Generative Model of Reading Comprehension. Journal of Reading Behavior, 1977, 9, 109-112.

Wittrock,M.C. The Cognitive Movement in Instruction. Educational Psychologist, 1978, 13, 15-29.

Wittrock, M.C. Generative Reading Comprehension. Ginn Occasional Reports, in press.

Wittrock,M.C. Learning as a Generative Process. Educational Psychologist, 1974, 11, 87-95.