EFH-Soar: Modeling Education in Highly Interactive Microworlds [*]

Cristina Conati
Intelligent Systems Program, University of Pittsburgh
Pittsburgh, PA, 15260, USA

Jill Fain Lehman
School of Computer Science, Carnegie Mellon University
Pittsburgh, PA 15213-3890 USA

Abstract

The goal of our work is to produce a process model account of education in microworlds based on Soar, a theory of cognition and learning. In the context of a microworld that supports the exploration of qualitative electrostatics, we present operational models of both skilled and student interaction. In addition, we describe an episodic memory mechanism encoded in the student model that gives insight into the processes involved in learning from incorrect behavior.

1 Introduction

In the field of computer-aided instruction, highly interactive microworlds have gained importance as educational tools aimed at supporting learning by exploration[7]. In contrast to more traditional educational strategies that "teach" the target knowledge to the student, learning by exploration focuses on stimulating the student's initiative in gaining knowledge about the domain[10, 11]. Highly interactive microworlds provide simulations that allow the student to experience the nature of some subject-matter domain through active exploration of the model's laws and behaviors.

Claims for the efficacy of interactive education have naturally led to studies evaluating those claims (for example, [3, 4, 5, 13]). For the most part, the format of such studies have fallen into the standard psychological experimental paradigm: a treatment group is exposed to an alternative instructional method (for example,

---

[*] This research was sponsored by the Markle Foundation. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the Markle Foundation.

computer-aided drill) while a control group is taught under normal classroom conditions. Performance of the two groups on a post-test measure of learning and transfer is then compared for evidence of a treatment effect. Because this method of evaluation examines only aggregate behavior and does not model the learning process itself, the results are usually difficult to interpret. When a treatment effect is found, it is unclear which aspects of the instructional method were responsible for the effect, i.e., which aspects enabled the superior learning or transfer. Similarly, when treatment effects are absent, it is unclear which aspects are at fault.

In contrast, we adopt a theory-based, cognitive approach to studying the question of instructional efficacy, believing that to evaluate a method of education we must understand the nature of the learning process itself. In essence, we propose to understand student learning behavior in highly interactive environments in terms of Soar, a specific theory of human cognition.

This paper describes the first stage of the EFH-Soar project, consisting of the formalization of the knowledge necessary to interact with a highly interactive environment within a model of a skilled player. Preliminary results obtained from the implementation of a student model are also presented.

2 The Subject-matter Domain

Our subject-matter domain is simple discrete electrostatics (hereafter, simply electrostatics). It consists of the electrostatics of discrete charged particles, all having the same mass. The relevant quantitative knowledge, linear attraction/repulsion of forces, Coulomb's Law and $F = ma$, is generally covered in about eight weeks of instruction in an elementary physics course.

Our interactive microworld is Electric Field Hockey (hereafter, simply Hockey) [2]. With respect to electrostatics, Hockey involves determining the trajectory of a unit-charge particle (the puck) from a given initial position, around a given set of obstacles, to a fixed final position (the net), by placing a number of additional unit-charge particles (Figure 1). The motion of the puck along its path is shown during the drawing of the trajectory. In addition, the puck's velocity is recorded statically by the spacing of the dots that represent the path followed.

The student has a limited number of options in controlling exploration in Hockey: specifically, the dot representation of velocity can be replaced with a vector representation of acceleration, and any of seven levels of increasingly difficult play may be chosen. Increasing difficulty is achieved by varying the initial state of play along four dimensions: the number and configuration of obstacles, the starting position of the puck with respect to the boundaries of the playing field and the obstacles, the existence of additional, unmoveable charged particles in the field, and the number of charged particles available to the player to maneuver the puck into the net.

As a basis for building the operational models we are interested in, a nearly complete video record of two students' education in the domain has been collected, culminating in sessions in which the students interact with Hockey as a class assignment. The purpose of the curriculum was to teach students a purely

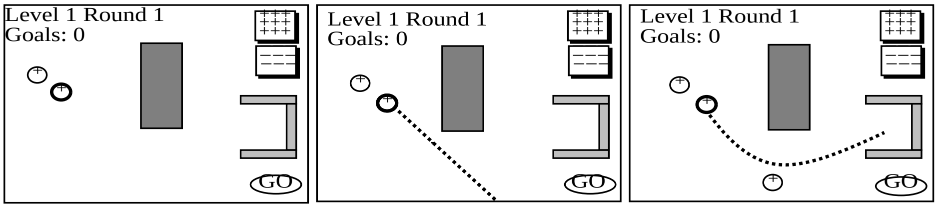


Figure 1: Electric Field Hockey: an example of interaction at Level 1.

qualitative model of electrical and magnetic phenomena as a theoretical basis for later, standard quantitative material [12]. In the course, students worked in groups of two through a notebook of desk-top experiments to be performed and questions to be answered. Our video record is of one such team. To date, we have concentrated on studying the video related to the performance of one of the two students (hereafter, Student 2) in her interaction with Hockey.

3 The Soar Cognitive Architecture

Soar [6, 8] belongs to a family of cognitive theories that share important features both in terms of psychological mechanisms and methods of use (see, e.g., [1, 9, 14]). These theories characterize human cognition as goal-directed problem solving.

As a particular member of this family, Soar can be described as a system that formulates tasks in terms of problem spaces, operators, and states. Problem solving proceeds in a sequence of *decision cycles.* Each decision cycle accumulates knowledge from a long-term, production-based *recognition memory* by allowing all the productions whose conditions match working memory elements in the current state to fire in parallel. The knowledge that is added to the state represents preferences concerning the next step to take. Once quiescence is reached (no more productions fire) a fixed *decision procedure* examines those preferences in order to choose a new problem space, operator, or state. If there is enough knowledge in long term memory to make the decision procedure's choice unequivocal, the preferred next step is taken and the next decision cycle entered.

If, on the other hand, Soar does not know how to proceed in a problem space because there is not enough knowledge to suggest a next step, or there is conflicting knowledge suggesting more than one step, an *impasse* has occurred. In response to an impasse, Soar creates a subgoal and a new problem space in which to acquire the missing knowledge (an impasse within the new space will have the same effect, i.e. Soar creates its own goal-subgoal hierarchy automatically as a result of being unable to proceed). Once an impasse has been resolved by problem solving in the subspace, the Soar learning mechanism (*chunking*) captures the result of problem solving in new productions. Chunking acts as knowledge compilation device and is automatic rather than deliberate, being invoked whenever an impasse is resolved. It has been used successfully to characterize learning in many tasks and domains, although it has not been previously applied to the types of educational environments that are the focus of EFH-Soar.

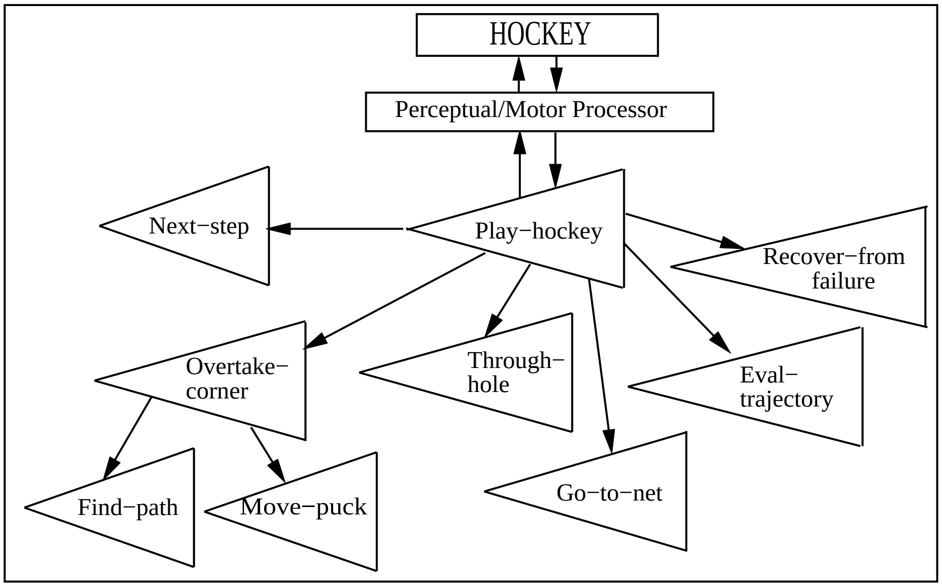


Figure 2: EFH-Soar problem spaces.

Figure 2 shows a portion of the problem spaces and operators that make up the current model. The model possesses the basic functionality for interacting with Hockey as a piece of external software, for re-encoding Hockey's spatial representation in symbolic terms, for reasoning about the placement of charges to create a trajectory that manuevers around or through an obstacle, for interpreting and evaluating a trajectory in terms of the forces that determine it, and for modifying the position of previously placed charges on the basis of the resulting trajectory.

We view the model in Figure 2 as a highly skilled Hockey player. At the lowest three levels of the game, it has the knowledge to play perfectly given the limits of its perceptual representation.[1] Constructing the system as a highly skilled player helps us understand what is necessary to have Soar play Hockey and provides a knowledge-level description of the student who has acquired the subject-matter domain. Thus, much of this system will form the basis of the student models. Indeed, our preliminary analysis of Student 2's behavior shows that the skilled model is a good fit at Level 1 of the game. At Level 3, however, the skilled

---

[1] Like humans, the model uses a symbolic spatial representation which has a grain size much larger than the actual number of unique locations in the field represented inside Hockey. In other words, neither we nor the model can guide the mouse to place a charge at a particular pixel location. Thus, EFH-Soar's intended location corresponds to a region inside the microworld. As a result, positioning of a charge can be correct at the higher level of granularity but slightly off at the lower level.

model performs correctly, while Student 2 does not. The discrepancy allows us to pinpoint both places where a model of Student 2 must differ from the skilled model and instances in Student 2's behavior where learning may occur.

4 EFH-Soar: The Skilled Model

The problem space description shown in Figure 2 gives a static view of the system at one level of detail. In this section we expand this view to include EFH-Soar's main knowledge structures and dynamic behavior.

4.1 Spatial Cognition and Representation

The essential feedback from Hockey takes the form of continuous trajectories of the puck as it moves under the influence of the fixed charges. How does the student perceive this continuous curve, such that it can be reasoned about by the symbolic processes that make up the student's knowledge? An analogous problem exists for output: how does the student's symbolic reasoning result in the placement of a charge in a continuous field? Both of these problems are simply specific (somewhat specialized) instances of the general, deep issue of moving between the continuous and quantitative on the one hand and the symbolic on the other.

In working with Hockey, we have developed a strong hypothesis about the human cognitive representation of the spatial world and how it relates to perception. Its most important property is that it is a highly approximate, qualitative representation that depends on continuous re-perception of the actual external world (the source of high-quality knowledge) to update and correct the low-quality internal representation. Thus, although the internal representation that cognition works with is approximate, it does not degrade. This hypothesis is coupled with many additional (unimplemented) mechanisms for how the symbol representation increases its discrimination, how perception maps the symbolic representation into the spatial world, and so on. EFH-Soar contains an initial realization of the hypothesis that extends prior work in this area [15].

Hockey is implemented in an object-oriented programming language developed at the Center for Design of Educational Computing at Carnegie Mellon University [2, 12]. For each object on the screen, Hockey encodes the coordinates of its position and its type. The objects are the puck, obstacles, net, charges placed on the screen, two boxes containing the available charges, and control buttons. In addition, the trajectory that displays the motion of the puck consists of points placed at equal time intervals, so that their spacing reflects the puck's speed. The program encodes up to 200 pairs of x-y coordinates in a trajectory.

The Perceptual-Motor Processor (PMP, see Figure 2) accomplishes the translation between Hockey's quantitative representation of the screen and the qualitative internal representation that is EFH-Soar's knowledge of the external world. This internal representation is EFH-Soar's *spatial model.* It encodes relative positions among objects whose descriptions vary in level of detail as a function of focus of attention. Specifically, the spatial model consists of:

- A structure, called the *spatial mapping*, with the following attributes:
    - *linear scale*: the scaling factor that filters all distances returned by the visual processor to be within the range of the system's internal resolution. The scaling value in the current model is 10.
    - *angular resolution*: the level of resolution for direction available in the internal representation. Currently the system descriminates 32 angles.[2]
    - *center of focus*: encodes the object that is the center of the focused area.
    - *focused objects*: a list of other objects in the focused area.
- The list of objects present on the screen. The objects explicitly represented in the spatial model are those encoded by the Hockey program plus the corners of the net and obstacles. If an object is not in focus, the only information explicitly available to the system is the object's type. If a shift of attention brings it into the focused area, its other attributes are added to the spatial model.
- Descriptions of the relative positions of objects within the focused area. These descriptions are encoded in two different ways: *spatial relations* define relative position in terms of the attributes left, right, below and above; *displacements* define relative position in terms of the distance and the relative direction between two objects.

The system performs visual operations by issuing visual commands to the PMP. Visual commands are used to perceive new situations and to perform shifts of attention to different areas of the screen.

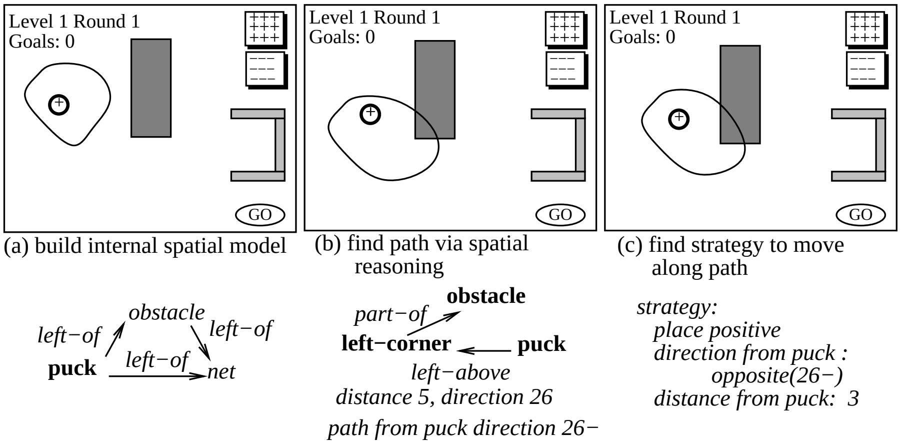
4.2 An Example of Problem-solving Behavior

The problem spaces in Figure 2 organize the knowledge EFH has to play Hockey. We consider a portion of that knowledge in tracing the system's behavior through the example in Figure 3. For convenience to the reader, the area in focus is circled in each picture, and focused objects appear in boldface in the state representation.

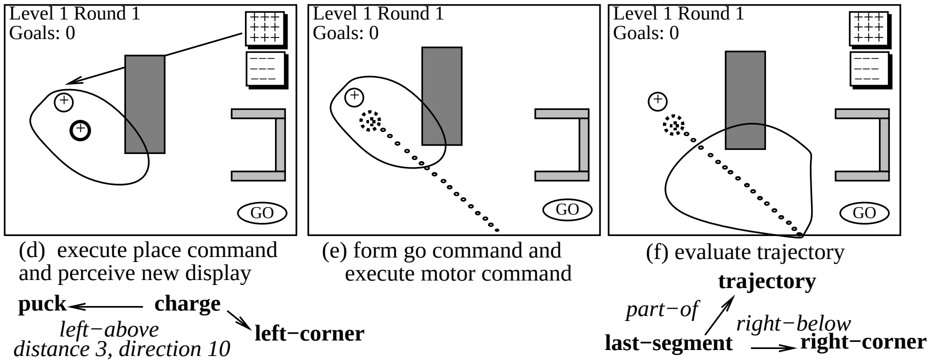
The first operation EFH-Soar performs in playing a new game is to perform a visual operation to acquire the spatial model of the current screen, with the puck as the center of focus (Figure 3a). Once the spatial model has been established, the system uses the *Next-step* problem space to propose a subtask for sending the puck into the net. The system, like our students, chooses subtasks by working its way left to right across the screen. Four subtasks are currently defined. *Overtake-corner* is used when the puck must clear a simple obstacle before reaching the net. *Through-hole* is used when the puck must go through an opening between

---

[2] The numbers in the spatial model have been established empirically by observing the system in interaction with Hockey. We believe that, in theory, a fixed resolution size is wrong and should be replaced with a mechanism for varying resolution depending on the grain size of the spatial reasoning performed. Modeling this variability is one of the project's future goals.



(a) build internal spatial model (b) find path via spatial reasoning (c) find strategy to move along path



(d) execute place command and perceive new display (e) form go command and execute motor command (f) evaluate trajectory

Figure 3: An example of EFH-Soar's problem-solving loop.

complex obstacles (see Figure 4). *Go-to-net* is used when no obstacle remains between the puck and the net. *Recover-from-failure* is used after an action fails to achieve the goal of the subtask.

The problem spaces that implement the first three subtasks have a common structure. They map the current spatial model into a motor action that performs the current subtask. Defining this motor action consists of two main phases:

- The spatial reasoning phase defines the trajectory that the puck must follow to accomplish the current subtask. Spatial reasoning is performed by operators in and subspaces of the *Find-path* problem space. To clear the obstacle in Figure 3, for example, the system defines a path to clear one of the obstacle's left corners by first performing a shift of focus to bring the obstacle into the focused area. Then, to decide which way to clear the obstacle, it brings into focus the obstacle's two left corners. Analyzing the direction between the puck and each of the two corners, the system chooses which corner to use to clear the obstacle, selecting the one that defines the smallest angle. The final path is defined as the direction between the puck and the chosen corner, slightly decremented to allow the puck to clear the edge (Figure 3b).

- The domain reasoning phase uses the *Move-puck* problem space to find a strategy that moves the puck along the desired path via game actions. It is in this space that the system decides whether to use an attractive or a repellent force, how many charges to place, and where to place them. Since we assume that everyone can perform the spatial reasoning to define a path for the puck to follow, it is in the Move-puck space that we expect to find the differences between a skilled and novice player. Only a person with the necessary physics and game knowledge can take into consideration all the factors relevant to arriving at a good move. In the skilled model, the rules in the *Move-puck* space encode knowledge about repulsion and attraction, superposition, and the effect of distance. In the example in Figure 3, these rules suggest both the strategy of placing a repelling charge close to the puck to move it along the desired path (Figure 3c) and a corresponding motor action.

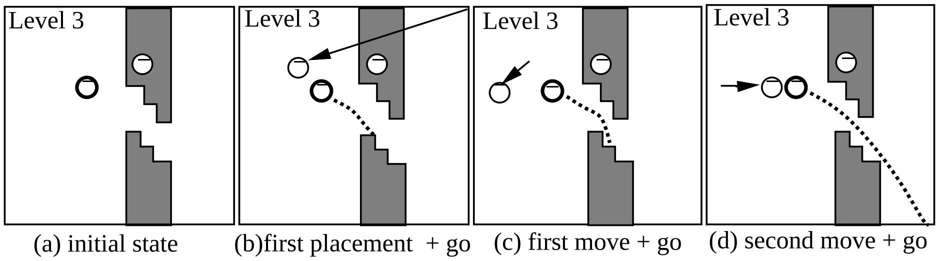
Every motor action proposed by a subtask is translated by the PMP into a Hockey command using the linear-scale and angular resolution contained in the spatial mapping. Once the action has been performed, the PMP perceives the new screen and modifies EFH-Soar's spatial model. As a default, the result of an action becomes the center of focus and all the objects previously focused on remain so (Figure 3d).

The system is now ready to click on the GO button, displaying the trajectory of the puck subject to the electrostatic force created by the placed charge (Figure 3e). The PMP encodes a qualitative representation of the trajectory in the spatial model which the system then focuses on using the *Evaluate-trajectory* problem space. The trajectory is evaluated in terms of the subparts defined by the positions of the objects relevant to the current subtask. In our example (Figure 3f), since the last portion of the trajectory overtook the chosen corner, the performed action is recognized as successful.

Having achieved the subgoal, the system again uses the *Next-step* problem space to define the next subtask to accomplish. Since there are no other obstacles between the puck and the net in our example, EFH-Soar chooses to try sending the puck directly into the net and the *Go-to-net* subtask is proposed. By repeating the phases of finding a path to accomplish the current subtask and a corresponding strategy to achieve the path, the skilled model eventually sends the puck into the net.

5 EFH-Soar: The Student Model

Given the skilled model described above, we have begun to build a student model for one of our two videotaped subjects. At Level 1, Student 2's behavior matches that of the skilled model.



(a) initial state (b)first placement + go (c) first move + go (d) second move + go

Figure 4: An example of learning from incorrect behavior.

The performances of the skilled model and the student begin to diverge at Level 3, when Student 2 faces the situation shown in Figure 4a. Her first action is the placement of a repelling charge that is inadequate to overcome the effects of the charge glued to the obstacle (Figure 4b). Her next move (Figure 4c) seems to be an attempt to compensate for the direction of the force produced by the glued charge but is still too far away to produce the desired outcome. Finally, in (Figure 4d), the forces combine correctly to achieve her subgoal of pushing the puck through the hole. When she faces an analogous situation at Level 4 of the game, Student 2 does not repeat this series of actions. Instead, she places the initial repelling charge directly at the position analogous to the one in Figure 4d. This episode is an example of learning from previous error. Although the capability to gain experience from past failures is a basic component of human learning, it tends to be problematic for machine learning systems that, like Soar, learn while doing. The difficulty is not so much in noticing, post hoc, that a choice was wrong, as it is in unlearning what was learned on the way to making that choice. In the remainder of this section we outline a general mechanism we have developed to allow EFH-Soar to model learning from incorrect behavior.

5.1 Learning from Incorrect Behavior

To model the student's initial incorrect behavior (Figure 4b), we eliminate from the skilled model the knowledge that takes into consideration the influence of close charges when computing a new placement. We call this modified model EFH2. The first time EFH2 faces the situation in Figure 4a, the model, like the student, ignores the effects of the glued particle, placing a charge aligned with a path through the opening. It chooses to place the charge at a distance from the puck that was adequate to achieve its goals at Level 1. EFH2 then clicks on the GO button and evaluates the resulting trajectory. Since it detects a failure, problem solving continues in the *Recover-from-failure* space where an action that adjusts the direction of the placed charge is selected (Figure 4c). Another GO reveals the second failure, and the *Recover-from-failure* space proposes an action that brings the charge closer to the puck (Figure 4d), allowing the puck to move through the opening. Chunks are built throughout the problem solving episode just described. When faced with a situation similar to Figure 4a in the future, these chunks will be triggered and EFH2, unlike Student 2, will repeat the sequence of proposing the first wrong placement, evaluating it, performing the direction adjustment, evaluating the second failure, and proposing the final, correct movement.

5.2 Overcoming Incorrect Behavior Through Assimilation and Recall

Our preliminary solution to this problem lies in giving EFH2 a simple episodic memory that is created and used by processes that allow the system to reconstruct past problem solving in order to avoid repeating mistakes. *Assimilation*, *recognition*, and *recall* are the three processes that must be coordinated. Although the details of the implementation are beyond the scope of this paper, it is critical to note that these mechanisms do not substitute for or supplant chunking; rather, they arise from chunking over additional types of problem solving in the model, as described below.

EFH's simple episodic memory is built up through the process of assimilation. Specifically, when a subtask proposes an action, the system uses an impasse into an *Assimilate* problem space to notice the features of the current situation. A situation is defined by the objects in focus, the current subgoal and the proposed action. The *recognition chunks* that result from the resolution of an Assimilation impasse will recognize the current situation as one that has been seen before, if it is encountered again in future problem solving. Assimilation is also performed after the evaluation of each action. In this case the situation comprises the outcome of the action (success or failure).

How do recognition chunks allow EFH2 to modify its behavior and learn to overcome its previous incorrect actions? The second time the system is in the state in Figure 4a, the chunk that proposes the first placement will fire, allowing the recognition chunks to detect the state and the proposed action as a situation it has seen before. This recognition causes an impasse into the *Recall* problem space where the system tries to avoid performing an unsuccessful move by recalling, *before* it acts, what the outcome of the action will be. In order to recall what happened the last time the proposed action was performed in the same state, the system must "imagine" the result of the action. Imagining means simply that a new charge and its spatial attributes are added to the spatial model without actually being present on the Hockey screen. If we annotate this state with "success" and a recognition chunk fires, we know that the placement was successful in the past. If no recognition chunk fires, we can annotate the state with different failure types; the failure type that triggers a recognition chunk represents the previous outcome of the current action in the current situation.

Once a failure has been recalled, EFH2 must reconstruct its prior problem solving to find the actions that led to success. To accomplish this, the system uses the *Recover-from-failure* space to reason about the imagined state, triggering the chunks that suggest the movement of the charge in Figure 4c. The recall/recognition process continues by imagining the movement of the charged particle, recalling the second failure, recalling the adjustment of the distance and the resulting success. Since the sequence of actions eventually leads to a success, the system resolves the impasse into the Recall space by substituting for the proposed (but unperformed) place action a new action that places the charge in the final position in Figure 4d. As the impasse resolves, chunking creates a new piece of knowledge that maps the original situation (Figure 4a) directly into this new action. Thus, the second time the model encounters the situation in 4a it must recall its prior mistakes to overcome its initial incorrect learning; but the third time it encounters 4a, it behaves skillfully, acting appropriately without problem solving.

6 Conclusions and Future work

The goal of the work presented here is to produce a process model account of education in microworlds. The skilled model helps us to understand what is necessary to have Soar play Hockey and gives a knowledge-level description of the student who has acquired the subject-matter domain. Future work will be directed at developing a model of the processes involved in the transition from naive to skilled player. This model should make clear what (if any) improvement in physics knowledge occurs during the transition. Our observation of student players indicates that it may be possible for a student to become skilled by solving problems in game terms, i.e. without significantly improving her physics knowledge. For this reason, the distinction between physics knowledge and game knowledge is fundamental to predicting the educational effectiveness of the microworld. It is part of our future work to formalize this distinction in EFH-Soar.

A first result in the development of a student model has been the realization of a mechanism for learning from incorrect behavior. Yet, the processes of assimilation, recognition, and recall discussed in the previous section are problematic because they are capable of reconstructing chains of memories of arbitrary length and involving the imagination of an arbitrary number of changes to the spatial model. This flaw in the mechanism is the result of two simplifying assumptions: first, that assimilation captures all and only the necessary details of the situation and second, that assimilation, recognition, and recall are automatic and unmediated by other processes.

The modified form of the mechanism we envision relaxes these assumptions. In it, each episodic memory is constructed by a potentially inaccurate assimilation process that can be automatic or mediated by further problem solving. If assimilation is no longer guaranteed to capture the complete and correct details of the situation, overspecific episodes may result in the breakdown of the recognition and recall processes. This should constrain the recall capabilities of the system with respect to both the length of the problem solving sequence and to the relevance of the performed actions.

References

[1] Anderson, J. R., *Rules of the Mind*, 1993, Lawrence Erlbaum Associates, Hillsdale, New Jersey.

[2] Chabay, R. and Sherwood, B., *The Electricity Project and the cT Programming Language*, 1989, Center for Design of Educational Computing, Carnegie Mellon University, Technical Report No. 89-10.

[3] Clements, D. H., *Effects of Logo and CAI environments on cognition and creativity*, 1986, Journal of Educational Psychology, 78, 309-318

[4] Clements, D. H.,*Metacomponential development in a Logo programming environment*, 1990, Journal of Educational Psychology, 82, 141-149.

[5] Klahr, D. and Carver, S. M., *Cognitive objectives in a Logo debugging curriculum: instruction, learning, and transfer*, 1988, Cognitive Psychology, 20, 362-404.

[6] Laird, J. E., Newell, A., and Rosenbloom, P. S., *Soar: An architecture for general intelligence*, 1987, Artificial Intelligence, 33, 1-64.

[7] Lawler, R., *Designing computer based microworlds*, 1987, New Horizons in Educational Technology, Yazdani, M., (eds), Ablex Publishing, Norwood, NJ,

[8] Newell, A., *Unified Theories of Cognition*, 1990 Harvard University Press, Cambridge, Massachusetts.

[9] Polson, P. G. and Kieras, D. E., A quantitative model of the learning and performance of text editing knowledge, 1985, in CHI '85 Proceedings.

[10] Schank, R., C., and Farrel, R. *Creativity in education: a standard for computer-based teaching*, 1988, Machine-Mediated Learning, 2, 175-194.

[11] Schaube, L., Glaser, R., Raghavan, K., and Reiner, M., *Causal models and experimentation strategies in scientific reasoning*, 1991, The Journal of the Learning Sciences, 2, 201-138

[12] Sherwood, B. and Chabay, R., *Electrical Interactions and the Atomic Structure of Matter: Adding Qualitative Reasoning to a Calculus-Based Electricity and Magnetism Course*, 1991, Proceedings of the NATO Advanced Research Workshop on Learning Electricity or Electronics with Advanced Educational Technology, Springer-Verlag, Berlin, Germany.

[13] Simon, T., *Claims for Logo; what should we believe and why?*, 1987, Computers, Cognition, and Development. Rutkowska, J. C. and Crook, C. (eds), Wiley and Sons.

[14] VanLehn, K., *Mind Bugs: The Origins of Procedural Misconceptions*, MIT Press, Cambridge, Massachusetts.

[15] Ward, B., ET-Soar: *Toward an ITS for Theory-based Representations, Ph.D. dissertation, 1991, School of Computer Science, Carnegie Mellon University.*