Learning and Instruction in the New Electric Elves Architecture

Written by
Jim Blythe, Jihie Kim, Yolanda Gil, Craig Knoblock, David Pynadath, Wei-Min-Shen.

Technical Point of Contact: Yolanda Gil (gil@isi.edu)

USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292

(310) 822-1511

Date of preparation: September 12, 2002

Innovative Claims

Cognitive systems embedded in human organizations could significantly enhance human performance by carrying out tasks in a maximally helpful and minimally intrusive manner. Because human organizations are highly dynamic environments, cognitive systems need to incorporate a fundamental ability to *adapt to changes in the environment where they operate and as a result improve their behavior over time*. The dynamic nature of the environment is due to a number of factors, including the varied and evolving nature of people's context and preferences, unexpected failures during the execution of the tasks, and the diversity of unanticipated combinations of resources (human and systems) required to perform or support complex tasks. The issue is exacerbated when the human organization is supported over a continuous and very long period of time where the environment evolves both slowly and abruptly due to different organizational changes.

Our claim is that it will take a combination of autonomous learning and interactive instruction techniques to make cognitive systems adequately adaptive. Autonomous learning will enable the creation of patterns of desirable behavior by generalizing from observations of the user and the system's performance over time. Interactive instruction will enable users (or other cognitive systems) to specify directly high-level guidance, additional constraints to take into account, or novel factors that should influence the system's decisions. Some of this guidance may be simply too hard to learn from few examples, given the well-known information-theoretic limitations of learning algorithms. In addition, some of the guidance may be prompted by significant changes of context that users would naturally expect the system to be able to assimilate up front instead of waiting for the system to learn it through performance failures.

We propose to explore learning and instruction approaches focusing on four key areas:

1. **autonomous learning from observation**, such as plan recognition to infer user's goals from observation in order to learn improved strategies for decision making.
2. **interactive instruction dialogues** that reason about what knowledge the system needs to acquire and what are the best strategies to acquire it from users.
3. **combining learning and interactive instruction**, such as elicitation of key features from users through selected examples combined with efficient on-line learning algorithms.
4. **learning during execution** to address execution/performance failures, such as predictive action models to compare the system's expectations with its observations.

Autonomous learning and interactive instruction are complementary techniques, but exploring and exploiting their synergies will be a key aspect of the work proposed. For example, a learning component may analyze its current knowledge to determine what missing pieces may be most useful to acquire according to some utility measure and pass that information to the interactive tool that may know best how to select, formulate, and organize appropriate questions to the user and organize them into a coherent dialogue. Conversely, queries to a user may represent generalized examples that help boost learning and as a result the system may be able to answer by itself questions that the interactive instruction tool had included in its planned upcoming dialogue.

We plan to conduct this investigation in the context of the new cognitive architecture proposed in another white paper for the ISI Electric Elves project.

Technical Rationale and Approach

The operation of a human organization involves dozens of critical everyday tasks to ensure coherence in organizational activities, to monitor the status of such activities, to obtain information relevant to the organization, to keep everyone in the organization informed, and so on. These activities are often well-suited for software agents, which can devote significant resources to perform these tasks, thus reducing the burden on humans. Such software agents enable organizations to act coherently, to attain their mission goals robustly, to react to crises swiftly, and to adapt to events dynamically.

Towards this vision, we previously developed an agent-based system called Electric Elves [Chalupsky *et al.* 2001]. We have successfully deployed the Elves in both an office and travel environment. The travel application is in use today and provides a tremendously useful set of services for a traveler, in making travel plans and monitoring their execution. In the process of building and deploying this successful system, we have learned a great deal about both its strengths and limitations. The current state of technology is agents that are carefully designed to work together, are capable only of responding to failures for which they were designed, cannot always explain their behavior, often require specialized interfaces, and require significant effort to develop.

In a separate white paper submitted to this same BAA #02-21, we have described in detail a new cognitive architecture that addresses these limitations. This architecture would support "intelligent agents" that: i) robustly accomplish their tasks, responding appropriately to failures; ii) communicate flexibly with humans and software agents; iii) explain their behavior both on success or failure; iv) dynamically compose new agents and behaviors from existing agents; v) rapidly build personalized agents without manual programming; vi) learn from their past experience. The architecture will result in a new Electric Elves framework and will support a wider variety of agents as well as a wider range of tasks such as battlefield awareness, network management, and disaster response.

In this white paper we are proposing research focused on a key portion of this architecture: to extend it with a combination of autonomous learning and interactive instruction capabilities that will enable the system to adapt in its highly dynamic environment and improve its performance over time. We start by describing in more detail each of the four key areas outlined in the previous section.

1) Autonomous Learning from Observation

Although we will design our system architecture to provide robust execution of the users' desired tasks, it is equally critical for the architecture to adapt to the heterogeneity and dynamics of the users, agents, and tasks over the system's prolonged lifetime. The system can obtain some of this knowledge by autonomously learning from its experiences in executing these tasks and in observing the behavior of the system and the users. This will enable the system to self-repair shortcomings in its knowledge through its normal operation and without placing an additional burden on the user.

The challenge in this autonomous learning task is twofold: (1) the architecture must determine what knowledge it must acquire, and (2) it must exploit the experience it has already available to fill in this missing knowledge as much as possible. We have had preliminary success with regard to the latter in the existing Electric Elves system. For instance, one key challenge is that users may have different requirements regarding the autonomy of the integration architecture to make decisions on their behalf. To avoid hand-tuning such autonomy for each human (or agent), we designed the architecture to automatically adapt their autonomy based on the ongoing history of their execution of user tasks [Tambe *et al.* 2000]. However, our existing learning methods have focused on learning very specific pieces of knowledge and have used only a very limited amount of the agents' available history. We plan to explore novel plan recognition (and, more specifically, user modeling) techniques that provide a general framework for continually extracting general knowledge from the agents' overall experiences. In the proposed work, we will exploit our previous work on plan recognition [Pynadath & Wellman 2000] and extend our framework's capability (unique in the field) for on-line translation of a continuous stream of observations into an autonomously learned decision-theoretic model of an agent's or user's preferences and behavior.

Furthermore, there is no existing work that addresses plan recognition in the context of our first learning issue of the identification of what knowledge the system needs. We propose to design the performance components of our agents to be aware of their overall learning goals to quantify the value of obtaining particular pieces of currently missing knowledge. In other words, the performance component could model its own behavior, with and without a piece of knowledge, and then measure its expected performance under both conditions with respect to its goals. The resulting utility measure for missing information can then be used to guide both the plan-recognition process used by the automated learning component as well as the interactive instruction component.

2) Interactive Instruction Dialogues

Current approaches to interactive instruction include interactive knowledge editors that have been shown effective to acquire complex knowledge (e.g., DARPA's RKF challenge problem evaluations). The techniques used include graphical and structured editors, diagnosing errors and helping users to fix them, and using existing knowledge to generate guidance [Blythe *et al.* 2001]. Alternative approaches to learn from observation, such as programming by demonstration, have only been successful at acquiring knowledge for relatively simple and well-confined tasks. Although interactive knowledge acquisition tools have been shown effective, users are still solely responsible for the process, in terms of deciding when, what, how, and how much to teach the system. We propose to explore a new approach to interactive instruction where the system has declarative representations of learning goals and strategies that will enable it to track its progress in acquiring and assimilating instruction. This would enable the system to capture meta-knowledge about the acquisition process and to exploit this meta-knowledge to guide further interactions with the user. In previous work, we have identified useful principles in the tutoring and learning literature that would be useful in interactive instruction in order to turn the system into a better student [Kim & Gil 2002].

3) Incorporating Interaction into Learning Algorithms

We argued above that both autonomous learning and interactive instruction are necessary for a cognitive system to provide life-long personalized assistance to a user. At the simplest level, autonomous learning can be made more efficient with key information from the user to point out which of many possible generalizations of examples best reflect what the user would desire the system to know. Closing the loop between these capabilities, so that information provided by the user can be complemented through learning techniques which may in turn suggest follow-up questions, will be key for reflective adaptation to changes in the environment. Doing this may require efficient incremental learning algorithms that respond quickly enough for interaction with the user. Another important function of learning algorithms could be to provoke the user to realize what knowledge the system is missing by seeking a comparison between carefully-chosen examples that are most informative. In past work, we began to explore these issues and developed a novel interactive learning approach to acquiring user preferences for flight planning using a decision-theoretic approach [Blythe 2002]. The system rapidly updated a feasible set of simple user preference functions, and provided feedback on the new feasible set in terms of the allowed solutions in order to elicit further preferences. We plan to explore how to design this kind of interactive learning for more complex types of knowledge and with other learning approaches. We will also explore how to design systems that can reflect over the learning goals and awareness annotations described above and bias the learning algorithms to explore certain generalizations that may result in the system resolving its own learning goals (i.e., answering its own questions instead of burdening the user).

4) Learning During Execution

While the above three areas address acquisition of new knowledge as a deliberative activity, learning plays an important role at performance time when there are failures during execution of a task. Beyond recovering from such failures and accomplished the task at hand, cognitive systems should have the ability to self-repair their knowledge so as to be able to avoid similar failures in future situations if at all possible. In previous work, we have successfully applied a prediction-based learning approach [Shen 1995] to our award-winning autonomous teams of learning robots [Shen 1999]. In this approach, a system reflects upon its knowledge to make predictions about the outcome of its actions and compares those predictions with observed consequences. The knowledge is then revised to correct the system's predictions. We plan to extend this approach to support many more agents and more complex coordination tasks as we envision for the new Elves architecture. We would also like to integrate this approach with the planning and self-repair capabilities that are being proposed for the new Elves architecture in another white paper for this same BAA.

In addressing these four key areas, we anticipate a number of common issues that will arise and that we now briefly discuss.

Organizing and Exploiting Accumulated Experience

To support both learning and interactive instruction in a persistent agent environment, the system will need to organize and annotate the history and experience of the system's performance as well as the evolution of its knowledge as it grows throughout its extended lifetime. This will result in an experience collection component that will raise a number of important research issues that we plan to explore. First, only selected parts of the accumulated experience will be relevant to specific agents, which are likely to have learning capabilities that will take input in the specific ontologies that the agent understands. Second, history needs to be segmented in order to reflect significant requirement changes due to the dynamics of its environment. Early experiences with a calendar management system that learned from watching a user schedule meetings showed that the system had no capability to reflect upon changes resulting from the beginning of a new semester or the undertaking of new obligations by the user. Finally, the system will need to capture some notion of decay of experiences that may be old and pertain to requirements that are no longer desirable due to changes in the system's environment.

Cross-Generalization of Experience

Another key issue is cross-generalization of experience in terms of identifying the scope of relevance for the agents' experiences with their users. Should the agent apply knowledge gleaned from a particular experience to only the specific task at hand or to all of its dealings with that user? Or should it generalize further and apply this new knowledge over other users as well, perhaps allowing an agent to learn about social norms? Generalization allows an agent to use a minimum of history to quickly tune its behavior over a wide variety of tasks and users. However, the agent must also be careful not to overgeneralize and thus mis-apply experience to learn an incorrect model of the user. The correct level of generalization will vary greatly across all of the tasks and users present. Therefore, the architecture will need a mechanism that can identify the extent to which it can apply different historical data and that can flexibly respond to data with different degrees of relevance.

Awareness Annotations Regarding Competence and Confidence

Effective cognitive systems currently lack awareness of what they know and have learned to date, as well as about what they do not know about yet. We propose to explore how awareness annotations to a body of knowledge that reflect the competence and confidence of the system based on frequency of use and resulting performance. These annotations would help improve robustness at performance time by influencing critical decisions based on the confidence level that the system has on the knowledge brought to bear. Furthermore, they would enable the system to determine what kinds of inputs from the user would improve their body of knowledge on both counts (confidence and competence). An important novel feature here is the focus on keeping track of what is known, not just on what is not known as is done in current acquisition interfaces.

Self-Explanation

Self-explanation capabilities are important for the learning components of the system. It is important that users understand what the system is doing and why, so as to enable them to provide the most helpful feedback and input possible, as well as to simply increase the users' degree of confidence in the agents' capabilities. To achieve such understanding, it is critical that the learning components of the system provide explanations of their behavior. These explanations will make the system's newly learned knowledge transparent to the user, as well as providing justifications for why the system reasons and behaves as it does. Agents should also be able to explain their behavior to other agents in terms of knowledge they use to make decisions. In addition, this capability will enable the interactive instruction component to engage the user in extending autonomously learned knowledge, since the learned knowledge will be expressed in terms understandable and explainable to the user.

The work proposed under this white paper will involve: (1) exploring each of the four areas outlined above, (2) understanding how they fit together, how they interact, and how to exploit synergies, (3) how to integrate learning and instruction capabilities within the new Electric Elves architecture as described in a separate submitted white paper, and (4) how the learning capabilities should influence the architecture's design.