Architectural Support for Routine Evolution

Charles Earl (EARL@CS.UCHICAGO.EDU)
R. James Firby (FIRBY@CS.UCHICAGO.EDU)
Department of Computer Science, The University of Chicago
Chicago, IL 60637

We are interested in designing autonomous agents that can continuously and intelligently adapt their behavior to the specifics of their environment through long term interaction in the environment.This paper presents a framework for *routine evolution* in the context of an plan execution system called RAMA for Routine Activity Maintenance and Analysis.

RAMA stores knowledge about what to do and what to expect in structures called *dynamics*. During execution, RAMA updates parameters associated with its expectations (e.g. duration, time of occurrence, frequency of occurrence). It uses this information to determine when unexpected events have occurred and when expectations have failed. It then modifies its dynamics so that such anomalies can be accounted for during subsequent execution.

Agre and Shrager refer to the process through which activity associated with recurrent goals becomes adapted to particular environments as *routine evolution* (Agre & Shrager, 1981) and posit that it is the basis of skill acquisition. An agent comes to an task for the first time with a rough plan and develops a detailed notion of what to expect as that activity unfolds. A theory of routine evolution must specify: 1) How an agent acquires expectations; 2) How are they refined through activity; and 3) What should happen when expectations are determined to be inaccurate? In this paper, we discuss how RAMA addresses these questions.

RAMA's knowledge consists of: 1) a library of process models, called *dynamics*, which describe what to do and what to expect as tasks are performed; 2) a library of anomaly types which RAMA uses to classify unexpected occurrences and expectation failures; 3) a library of dynamic transformations which describe how to modify dynamics in response to discovered anomalous events or discovered dependencies; and 4) a library of suggestive causal patterns, which describe which possible dependencies to look for to explain anomalous events. RAMA's interpreter retrieves appropriate dynamics in response to goals.

During execution RAMA updates expectations. The more accurate these estimates are, easier it is to spot when something unusual has happened. RAMA updates the following information for each dynamic it uses: 1) **Interval durations**, which are used to measure how long events last and the time between events; 2) **Likelihood estimates**, which measure how likely it is for an event to occur; and 3) **Completion rate estimates**, which estimate how likely it is that the dynamic will complete.

Unexpected events and expectation failures constitute anomalies. RAMA *recognizes* an unexpected event by comparing its observations of the world against the predictions of its active dynamics. If an observation is not predicted that observation is an unexpected event. Likewise, when an event is predicted to occur but not observed when expected, an expectation failure occurs.

When an anomaly occurs, RAMA classifies it as belonging to one of a set of anomaly types. These anomaly types overlap and extend the *nonmotivated* (i.e. not involving other intentional agents) anomaly types Leake (1991) defines in his thesis. The anomaly types are used as indeci that can trigger the activation of other dynamics or retrieve one or more simple transformations that say how to modify the dynamic to account for the anomaly during subsequent execution.

Finally, RAMA attempts to determine precursors to expectation failures or unexpected events by learning dependencies. This learning is guided by a notion of what events are critical enough to merit learning precursors for. For example, having the expected outcome of a dynamic not obtain by the expected completion time of the dynamic would constitute such an event. RAMA then uses dependency analysis (Cohen & Howe, 1995) to determine likely precursors of such a failure.

We have implemented the described architecture and have begun evaluation in a graphical simulator developed here at the University of Chicago. We are investigating how to tractably tune the dependency analysis.

References

Agre, P., & Shrager, J. (1993). Routine Evolution as the Microgenetic Basis of Skill Acquisition *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society* (pp. 510–516). Hillsdale, NJ: Lawrence Erlbaum Associates.

Howe, A., & Cohen, P. (1995). Understanding Planner Behavior. *Artificial Intelligence*,*6*, 287–317.

Leake, D. (1991). *Evaluating Explanations*. Doctoral dissertation, Department of Computer Science, Yale University, New Haven.