Deriving and Using Abstract Representation in Behavior-Based Systems

Monica N. Nicolescu and Maja J. Matarić

Computer Science Department University of Southern California 941 West 37th Place, Mailcode 0781 Los Angeles, CA 90089-0781 monica|mataric@cs.usc.edu, http://www-robotics.usc.edu/~agents/projects/abn.html

We present a representation that addresses two current limitations of the behavior-based systems (BBS) (Matarić 1992), (Arkin 1998): the lack of abstract representation within behaviors (which makes them hard to use in complex, sequential problems) and the need for behavior redesign even for tasks that use subsets of the same behavior set. We introduce the concept of behavior networks, based on the abstract behaviors representation described below.

We distinguish the following two types of behavior preconditions: world preconditions (activate the behaviors based on the state of the environment) and sequential preconditions (task-dependent conditions, often postconditions of other existing behaviors). In standard BBS behaviors, both types of preconditions are tested together, thus hard-coding a particular solution. The key step in adapting specialized behaviors to more general use is in the separation of the execution conditions from the outputs or actions, which allows for a more general set of activation conditions. The pairing of a behavior's conditions and its effects, without the specification of its inner workings, constitutes an *abstract behavior*. Intuitively, this is simply an explicit specification of the behavior's execution conditions (i.e., preconditions) and its effects (i.e., postconditions). The result is an abstract and general operator much like those used in classical deliberative systems (Fikes & Nilsson 1971). The behaviors that do the work that achieves the specified effects under the given conditions are called *primitive behaviors*, and may involve one or an entire collection of sequential or concurrently executing behaviors, as is typical for BBS. Behavior networks are a means of specifying strategies or general "plans" in a way that merges the advantages of both abstract representations and behavior-based systems. The nodes in the networks are abstract behaviors, and the links between them represent precondition and postcondition dependencies. The task plan or strategy is represented as a network of such behaviors.

We have implemented the proposed concepts on a physical mobile robot (Pioneer 2-DX) given an object delivery task in an enclosed, 2-section environment. The robot successfully finds a box, which may be in either section, goes with it through the door and pushes it to the delivery point. The solution makes use of two behavior networks and captures the important aspects of the proposed concepts: abstract representation, behavior reuse, behavior networks and the importance of relying on real embedded behaviors.

As a next goal, we seek to automate the behavior network generation and to use the representation to address humanrobot interaction. The abstract representation should allow us to employ simple communication mechanisms which would enable the robots to benefit from the human and also learn from and share their acquired knowledge and experiences.

The abstract behavior representation we are proposing combines the advantages of deliberative, STRIPS-like architectures (Fikes & Nilsson 1971), and those of BBS' capability to operate in dynamically changing environments. However, it is important to note that we are not describing a hybrid architecture. Our work is related to the approaches of Kaelbling & Rosenschein (1990) (the situated automata model) and of Lyons & Arbib (1989) who developed a robot schema model of computation for sensory-based robot programming. However, their implementations do not allow generalization and reuse of the compiled high level circuitry and respectively the robot schemas to multiple tasks. Maes (1990) describes an action selection mechanism for a situated agent, based on spreading activation within a network created dynamically from a given behavior repertoire. A key difference is that the network nodes, at least in the demonstrated examples, consist of STRIPS-like high-level operators much more abstract than those we employ.

References

- Arkin, R. C. (1998), *Behavior-Based Robotics*, MIT Press, CA.
- Fikes, R. E. & Nilsson, N. J. (1971), 'STRIPS: A new approach to the application of theorem proving to problem solving', *Artificial Intelligence* **2**, 189–208.
- Kaelbling, L. P. & Rosenschein, S. J. (1990), 'Action and planning in embedded agents', *Robotics and Autonomous Systems*(1&2), *June 1990* **6**, 35–48.
- Lyons, D. M. & Arbib, M. A. (1989), 'A formal model of computation for sensory-based robotics', *IEEE Transactions on Robotics and Automation* **5**(3), 280–293.
- Maes, P. (1990), 'Situated Agents Can Have Goals', *Journal* for Robotics and Autonomous Systems **6**(3), 49–70.
- Matarić, M. J. (1992), 'Integration of Representation Into Goal-Driven Behavior-Based Robots', *IEEE Transactions* on Robotics and Automation **8**(3), 304–312.

Copyright © 2000, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.