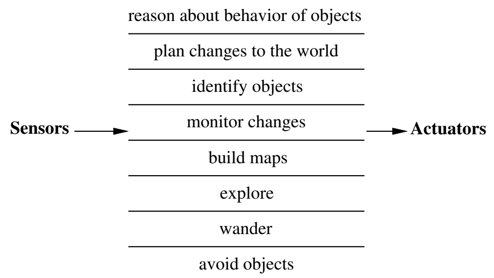

Figure 25.10 Design for a behavior-based mobile robot, showing the layered decomposition.

present, however, it is difficult to see how the design methodology can scale up to more complex tasks. Even for the task "Find an empty cup and bring it back to the starting location," the robot must form an internal representation that corresponds to a map in the classical architecture. Furthermore, the behavior-based approach requires the design of a new controller for each task, whereas classical robots are **taskable**: they can be assigned a goal and can carry out a plan to achieve it. In many ways, behavior-based designs resemble the ILA and LLA levels in Shakey, which are "nonclassical" components of an overall classical architecture.

Rosenschein's situated automata fall somewhere in between. The process of generating the automaton can be completely automated, so that if the the compiler is included in the robot's software, the robot can be given a task and can compile its own situated automaton. The automaton can then be executed in software. Comparing this approach with the explanation-based learning method used in SOAR and THEO, we see that the situated automaton is compiled prior to execution and must handle all possible situations that can arise, whereas the explanation-based learning approach generates efficient rules for handling the types of situations that actually arise during execution. In simple domains, the situated-automaton approach is feasible and probably more efficient, whereas in very complex domains, especially those with recursive structure, the complete automaton becomes too large or even infinite in size.

25.5 Configuration Spaces: A Framework for Analysis

Recall from Chapter 3 that the main element in analyzing a problem is the **state space**,[6] which describes all the possible configurations of the environment. In robotics, the environment includes the body of the robot itself. The main distinction between the problems discussed in Chapter 3 and those in robotics is that robotics usually involves *continuous* state spaces. Both the configuration of the robot's body and the locations of objects in physical space are defined by real-valued

[6] There is a source of confusion here because the term "state space" is used in the robotics and control literature to include the robot's state parameters and certain of their derivatives. Enough derivatives are included that the robot's motion can be predicted from its dynamics and control equations. For simplicity, we assume here that derivatives of state parameters are excluded.