take into account more information, to plan ahead, to handle situations in which no immediate response is available, and to produce better responses tailored specifically for the current situation. Compilation processes such as explanation-based learning (Chapter 21) continually convert declarative information at the deliberative level into more efficient representations, eventually reaching the reflex level (Figure 27.1). Architectures such as SOAR (Laird *et al.*, 1987) and THEO (Mitchell, 1990) have exactly this structure. Every time they solve a problem by explicit deliberation, they save away a generalized version of the solution for use by the reflex component.

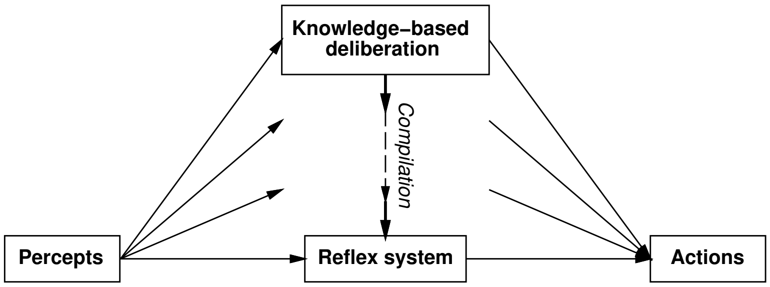

Figure 27.1 Compilation serves to convert deliberative decision making into more efficient, reflexive mechanisms.

As we saw in Part V, uncertainty is an inevitable problem in the real world. We also saw how uncertain knowledge is one possible response to the fact that exactly correct rules in realistic environments are usually very complex, and hence unusable. Unfortunately, there are clear gaps in our understanding of how to incorporate uncertain reasoning into general-purpose agent architectures. First, very little work has been done on compilation of uncertain reasoning and decision making. Second, we need more expressive languages for uncertain knowledge—a first-order probabilistic logic. Third, we need much better mechanisms for planning under uncertainty. Current algorithms, such as policy iteration (Chapter 17), are really more analogous to the simple search algorithms of Part II than to the powerful planning methods of Part IV. The latter methods incorporate partial ordering, goal directedness, and abstraction in order to handle very complex domains. *AI is only just beginning to come to grips with the problem of integrating techniques from the logical and probabilistic worlds* (Hanks *et al.*, 1994).

Agents in real environments also need ways to control their own deliberations. They must be able to cease deliberation when action is demanded, and they must be able to use the available time for deliberation to execute the most profitable computations. For example, a taxi-driving agent that sees an accident ahead should decide either to brake or to take avoiding action in a split second rather than in half an hour. It should also spend that split second thinking about the most important questions, such as whether the lanes to the left and right are clear and whether there is a large truck close behind, rather than worrying about wear and tear on the tires or where to pick up the next passenger. These issues are usually studied under the heading of **real-time AI**. As AI systems move into more complex domains, all problems will become real-time because the agent will never have long enough to solve the decision problem exactly (see also Section 27.2). This issue came up in our discussion of game-playing systems in Chapter 5, where

REAL-TIME AI