Cognitive Architectures and the Construction of Intelligent Agents

Pat Langley
Computational Learning Laboratory
Center for the Study of Language and Information
Stanford University, Stanford, CA 94305 USA

A central aim of artificial intelligence and cognitive science is the construction of *intelligent agents*, which we can define as software artifacts that exhibit intelligent behavior in complex domains over extended periods of time. Intelligent agents can display different levels of functionality, but the long-term goal is to meet or exceed the broad range of capabilities that are found in human behavior.

Although the construction of intelligent agents requires the design and creation of computer software, there are quite different paradigms for responding to this challenge. One framework involves the application of principles and techniques from software engineering, which are used regularly in developing traditional large-scale software systems. Another alternative comes from the multi-agent systems movement, which has become popular in some AI applications. A third paradigm, on which we will focus here, embeds the agent within an established *cognitive architecture*.

As typically defined, a cognitive architecture has a number of distinctive characteristics that support the creation of intelligent agents. The most basic is that such an architecture comes with a programming formalism in which to encode knowledge and an associated interpreter. Thus, the developer has access to a programming language, and often a programming environment, to utilize in building intelligent systems. For example, architectures like Soar (Laird et al., 1987) and ACT-R (Anderson, 1993) specify a syntax for production rules and declarative memory elements, along with an interpreter to run these programs. Alternative frameworks like Icarus (Choi et al., in press) provide different structures but still provide a clear formalism.

More important is the fact that cognitive architectures incorporate strong assumptions about the representation of knowledge and the processes that operate on them. As a result, their associated programming languages have higher-level constructs than traditional formalisms, which lets one generate more compact code for many problems. For example, the typical interpreter for a cognitive architecture includes pattern-matching capabilities, which reduces greatly the code needed to produce interesting behavior. They may also incorporate default mechanisms for search control, which can reduce the amount of code further.

Another key factor is that cognitive architectures assume a *modular* representation of knowledge. Combined with additional constraints, such as a commitment to problem-space search, this supports the incremental construction of intelligent agents by addition of knowledge elements that are composed automatically with the existing knowledge base. Traditional programming paradigms allow introduction of new subroutines, but cognitive architectures let one add content at a much finer granularity.

This fine-grained modularity also offers the potential for agents to acquire their knowledge from experience. In fact, many cognitive architectures include learning mechanisms designed to support this process, although they differ substantially in their details. Successful results in machine learning have depended invariably on strong assumptions about the representation of knowledge, including a modular formalism and an associated interpreter. Since learning seems a desirable capability for intelligent agents, this is another reason that cognitive architectures support their effective construction.

In closing, we should note that developing intelligent agents for complex domains requires the *integration* of different types of knowledge and different capabilities. To many, this suggests the use of software engineering methods, which focus on specifying interfaces between distinct modules. In contrast, the paradigm of cognitive architectures provides a *unified* approach in which a common set of representations and mechanisms reduce the need for such careful crafting. Thus, it provides an important and viable alternative for the construction of large-scale intelligent systems.

References

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Lawrence Erlbaum.

Choi, D., Kaufman, M., Langley, P., Nejati, N., & Shapiro, D. (in press). An architecture for persistent reactive behavior. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*. New York: ACM Press.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, *33*, 1–64.