APOC - An Architecture Framework for the Analysis and Development of Complex Agents

Matthias Scheutz
Artificial Intelligence and Robotics Laboratory
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
MSCHEUTZ@CSE.ND.EDU

July 15, 2003

Todays most commonly held theories of mind are versions of "functionalism", i.e., the view that mental states such as "believing that p" or "desiring x" are construed as states of a functional architecture, which is eventually realized in physical states (in biological organisms these physical states are brain states). However, there are currently no formal accounts of what functional architectures are, or what they would look like for a complex, human-like agent. And even if there were such accounts in terms of causal networks of states, they would likely not be very helpful to AI researchers who intend to build complex agents based on such architectures. In addition, a series of problems would likely result from the hitherto ill-defined notions of "functional realization and implementation" [4, 5, 7]. Furthermore, even if it were possibly to relate high-level functional states directly to low-level physical states, this relation would very unlikely give us any insight into the inner workings of a complex system. For one, functional states are typically holistic, and their inner structure is essentially obscured. What's more, the causal mechanisms that bring about the instantiation of these states cannot be understood in terms of these states, even if causal relationships are viewed as state transitions among them. Therefore, an agent architecture framework is needed, in which architectures of complex agents can be specified in a way that (1) allows us to see their high-level functional organization, and (2) permits us to specify functional components that can be implemented in computers.

In this chapter, we will introduce such an *architecture framework* that we have developed over the last several years called APOC (for "Activating-Processing-Observing-Components") [3, 1, 2, 15]. APOC not only provides a unified framework for the specification of architectures of minds at different levels of abstraction, but as such it also provides an intermediary level of architectural specification between functionalist views (as favored by philosophers) and physical descriptions (as favored by neuroscientists) that connects these two levels in a systematic way and allows AI researcher to design agent architectures to be implemented on computers. To facilitate the design of complex agents, we have created a *APOC development environment* in JAVA, which is built on APOC, allows for the distributed implementation of architectures specified in APOC on multiple computers, and furthermore supports the interaction with simulated and robotic agents in a transparent way.

APOC consists of heterogeneous computational units (based on [6, 8]) called *components* that can be connected via four link types to form an agent architecture. The four link types cover important basic interaction types among components in agent architectures: the *activation link* (A-link) allows components to communicate with other components; the *observation link* (O-link) allows components to observe the state of other components; the *process control link* (P-link) enables components to influence the computation taking place in other components, and finally the *component link* (C-link) allows a component to instantiate other components and connect to them via the other three links.

Components can vary with respect to their complexity and the level of abstraction at which they are defined. They could be as simple as a connectionist unit or as complex as a full-fledged condition-action rule interpreter. APOC can be used as an analysis tool for the evaluation of architectures, since it can express any agent architectures in a unified way (e.g., cognitive architectures such as SOAR, ACT-R, and others, as well as behavior-based architectures such as subsumption, motor schemas, situated automata, etc.).

APOC also supports the idea that mental states (or concepts) can be defined in terms of architectural capacities of agent architectures [13, 10, 11, 9, 12]. Hence, component and link structures in APOC can be used to define minimal requirements for the presence of mental states: an architecture A is capable of instantiating a mental state S, if the architecture-based definition of S in terms of APOC is a substructure of A. Thus, APOC is a first attempt to provide a framework that not only allows for a detailed computational specification of architecture-based concepts, but also provides an immediate implementation of the specification, thereby contributing to the resolution of one of the most critical questions in the foundations of AI: how do we know that a given physical system instantiates a given mental state? Systematic investigations of APOC structures can then be used to develop an architecture-based taxonomy of possible cognitive and affective states [14].

APOC also introduces a novel idea that is essential for the study of computationally plausible theory of minds: the notion of *cost induced by an architecture*, which is defined in terms of the cost associated with *structures*, *processes*, and *actions on the architecture*. *Structural costs* are those that are incurred as a result of merely having a certain component or link instantiated. They can be thought of as maintenance costs that are associated with any work that needs to be done to keep a component or link up to date. *Process costs* are those associated with running processes; they include computational costs, and possibly the costs of I/O and other such operations. Typically process costs will be proportional to the complexity of the computation performed by the process. Finally, *action costs* are those associated with primitive operations on the architecture (such as instantiating a new component or link, or interrupting a process). Each action has a fixed cost, making the computation of action costs a simple matter of assessing the associated cost whenever the action is executed.

The notion of cost induced by an architecture is then inductively defined in terms of these three basic cost types. It is crucial for any theory of mind that takes the nature and constraints of real-world agents into account as complex minds have to intrinsically cope with resources restrictions. We conjecture that (efficient) resource management is one dimension along which to evaluate designs of complex agents (evolution certainly did).

Finally, using the notion of cost induced by an architecture, the notion of *performance-cost trade-off* $PCT(P,A,T,E)$ for an agent architecture A and a task T in an environment E can be defined as P/C, where P is the given performance measure for T and C is the cost of A for T in E. Mathematically, performance-cost trade-offs are orders, and can thus form the basis of the comparison of agent architectures: given an order $>_P$ defined on P, an architecture A is said to be *better* than an architecture B with respect to T, E, and P, if $PCT(P,A,T,E) >_P PCT(P,B,T,E)$. As part of this chapter we will show how APOC allows for a novel comparison of different agent architectures in terms of the cost induced by them. This will help in answering questions about evolutionary trajectories of architectures of biological organisms as evolution will always favor architectures with higher performance-cost trade-offs.

References

[1] V. Andronache and M. Scheutz. APOC - a framework for complex agents. In *Proceedings of AAAI Spring Symposium 2003*. AAAI Press, 2003.

[2] V. Andronache and M. Scheutz. Growing agents - an investigation of architectural mechanisms for the specification of 'developing' agent architectures. In *Proceedings of FLAIRS 2003*, pages 2–6. AAAI Press, 2003.

[3] Virgil Andronache and Matthias Scheutz. Contention scheduling: A viable action-selection mechanism for robotics? In Sumali Conlon, editor, *Proceedings of the Thirteenth Midwest Artificial Intelligence and Cognitive Science Conference, MAICS 2002*, pages 122–129, Chicago, Illinois, April 2002. AAAI Press.

[4] Matthias Scheutz. Implementation: Computationalism's weak spot. *Conceptus*, 31(79):229–239, 1998.

[5] Matthias Scheutz. When physical systems realize functions... *Minds and Machines*, 9(2):161–196, 1999.

[6] Matthias Scheutz. Behavioral states: Linking functional and physical descriptions. In John Barnden, editor, *Proceedings on the AISB'2000 Convention on Designing Human-like Minds*, pages 389–399. Society for the Study of Artificial Intelligence and the Simulation of Behaviour, 2000.

[7] Matthias Scheutz. Causal vs. computational complexity? *Minds and Machines*, 11((4)):534–566, 2001.

[8] Matthias Scheutz. Ethology and functionalism: Behavioral descriptions as the link between physical and functional descriptions. *Evolution and Cognition*, 7(2):164–171, 2001.

[9] A. Sloman. What sort of control system is able to have a personality. In R. Trappl and P. Petta, editors, *Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agents*, pages 166–208. Springer (Lecture Notes in AI), Berlin, 1997.

[10] A. Sloman. Architectural requirements for human-like agents both natural and artificial. (what sorts of machines can love?). In K. Dautenhahn, editor, *Human Cognition And Social Agent Technology*, Advances in Consciousness Research, pages 163–195. John Benjamins, Amsterdam, 2000.

[11] A. Sloman. Beyond shallow models of emotion. *Cognitive Processing: International Quarterly of Cognitive Science*, 2(1):177–198, 2001.

[12] A. Sloman. Varieties of affect and the cogaff architecture schema. In C.G. Johnson, editor, *Proceedings Symposium on Emotion, Cognition, and Affective Computing AISB'01 Convention*, pages 39–48, York, March 2001.

[13] A. Sloman. Architecture-based conceptions of mind. In *Proceedings 11th International Congress of Logic, Methodology and Philosophy of Science*, pages 397–421, Dordrecht, 2002. Kluwer. (Synthese Library Series).

[14] Aaron Sloman and Matthias Scheutz. A framework for comparing agent architectures. In *UK Workshop on Computational Intelligence*, 2002.

[15] APOC. http://www.nd.edu/˜airolab/apoc/.