Just What is a Robot Architecture Anyway?
Turing Equivalency versus Organizing Principles

Ronald C. Arkin
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

1 Introduction

Over the years, there has been seemingly endless debate on how robot software architectures differ from each other and how they resemble each other. Often points are made that some architectures can do one thing while another cannot, or that in fact they are equivalent. The question is posed "Just what does it mean when we say that an architecture is different in some respect from another or that they are in some ways equivalent?" An effort is made in this paper to answer that question.

2 Equivalency

Starting with the issue of equivalency first, the most oft heard refrain is that of Turing equivalency - i.e., a reductio ad absurdem argument. Certainly robot architectures in the extreme are all computable functions. But even among Turing machines (TM) we can see organizing principles that can facilitate computation (or proofs as it were). We have, for example, the basic TM with a single head and tape with the ability to move left or right one cell. Hopcroft and Ullman describe techniques for the construction of more sophisticated Turing Machines [12] which are equivalent in a theoretical sense yet have certain advantages for specific proofs. These include multi-tape TMs, two-way infinite tape TMs, nondeterministic Turing machines, multidimensional Turing machines, multihead Turing machines, off-line Turing machines, multistack machines, and counter machines. Seemingly a lot of different architectures for a simple concept.

It is no wonder then that a large number of software robotic architectures exist. It is my observation that the major distinction between these architectures is not that of computability, but rather that of efficiency: certain architectures are better suited for certain tasks. This is manifested by the underlying organizational principles embedded within the architecture itself. Similar arguments can be made for programming languages as well, e.g., assembler versus fortran versus lisp versus APL.

Several different criteria have emerged by which we can characterize these manifold approaches. The dimensions of architectural design choices involve decision factors such as:

- Analysis versus Synthesis
  This methodological difference relates to the underlying assumptions regarding just what intelligence is. In some instances intelligence is perceived as something that can be reduced to an atomic unit, which when appropriately organized and replicated can yield high-level intelligent action. In other approaches, abstract pieces of intelligent systems, often extracted from or motivated by biological counterparts, can be used to construct the requisite robotic performance.
- Top-down (knowledge-driven)/Bottom-up (data--driven) design
  This relates more closely to experimentation and discovery as a design driver versus a formal analysis and characterization of the requisite knowledge that a system needs to possess to manifest intelligent robotic performance. These differences perhaps parallel to a degree the "scruffy/neat" dichotomy in AI.
- Domain relevance versus domain-independence
  To some extent this characteristic captures the view that there either is or is not a single form of intelligence. Here the AI parallel is "weak versus strong" methods.

3 Organizing Principles

Two major classes of architectures can be noted that differ in their compositional method. The first assumes that a uniform specification when replicated (recursively or otherwise) is adequate for all tasks relevant to a robotic system. The second category chooses rather to combine multiple disparate strategies together for robot control. Each of these are discussed in turn.

3.1 Unified Field Theory Approaches

Strong claims regarding the adequacy of a particular formalism or mechanism as the basis for providing broad scale robot capacity have been made for several different architectures. It is interesting to note that these approaches differ significantly regarding the commitment made to the design of each one's particular fundamental unit of intelligence. Representative examples include the multi-level hierarchical approach espoused by Albus in his Theory of Intelligence [1], Meystel's recursive formalisms based on hierarchical organization principles [14], and Brooks' subsumption architecture [8]. It is worth observing that Brooks' approach is often viewed as diametrically opposed to the other work cited, but a commonality nonetheless exists in the assertion that a single construct when replicated is sufficient to ultimately reproduce real-time human-level intelligence.

3.2 Mix-and-match Approaches

Another approach to robotic architectures assumes that there are more than one architectural design unit in play, each of which is epistemologically distinct. Often these architectures include aspects of both deliberative and reactive control and are often broken into two, three, or more qualitatively different levels. Representational examples of these "accretional architectures" include Arkin's AuRA system [2], Gat's Atlantis [13], and Firby's RAPS [11], among others. Here no commitment is made to a particular processing paradigm that is to be considered uniform throughout intelligence. A partition of functionality is made based on a robotic agent's particular needs at a given time. These hybrid approaches may lack the simplistic elegance found in a uniform theoretical expression of intelligence, but the question is open as to whether a single control construct used through replication can adequately capture all aspects of real-time human-level performance. Unified field theories, in general, are notoriously hard to establish.

4 Finally, Niche Finding

McFarland's concept of ecological niches [15] is particularly relevant to the question of appropriate architectural design. This view promulgates that robotic systems must find their place within the world as competitors with other ecological counterparts (e.g., people). The issues regarding robotic system cost also serve as important driving factors. In order for robots to be commonplace they must find the ecological niches that allow them to survive and/or dominate their competitors, whether they be mechanical or biological. If indeed one can foresee multiple potential niches for robotic systems (just as there are multiple niches for biological systems) it is not hard to envision the utility of a wide range of architectural paradigms. This argues for the value of a diversity of architectural solutions. Granted not all current architectural approaches will survive, and indeed some may dominate the diversity of solutions, but ultimately it is just this ecological pressure that will serve as the fundamental architectural selection mechanism, not an academic's perspective on their elegance, simplicity, or utility.

References

[1] Albus, J.S., "Outline for a Theory of Intelligence", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 21, No. 3, pp. 473-509, May-June 1991.

[2] Arkin, R.C., "Integrating Behavioral, Perceptual, and World Knowledge in Reactive Navigation", *Robotics and Autonomous Systems*, 6 (1990), pp. 105-122.

[3] Arkin, R.C., "Neuroscience in Motion: The Application of Schema Theory to Mobile Robotics", chapter in *Visuomotor Coordination: Amphibians, Comparisons, Models, and Robots*, ed. P. Evert and M. Arbib, Plenum Publishing Co., pp. 649-672, 1989.

[4] Arkin, R.C., "The Impact of Cybernetics on the Design of a Mobile Robot System: A Case Study", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 6, Nov/Dec 1990, pp. 1245-1257.

[5] Arkin, R.C., "Modeling Neural Function at the Schema Level: Implications and Results for Robotic Control", chapter in *Biological Neural Networks in Invertebrate Neuroethology and Robotics*, ed. R. Beer, R. Ritzmann, and T. McKenna, Academic Press, pp. 383-410, 1992.

[6] Balch, T. and Arkin, R.C., "Communication in Reactive Multiagent Robotic Systems", *Autonomous Robots*, Vol. 1, No. 1, pp. 27-52, 1994.

[7] Beer, R., Chiel, H. and Sterling, L., "A Biological Perspective on Autonomous Agent Design", *Robotics and Autonomous Systems*, 6 (1990), pp. 169-186.

[8] Brooks, R, "A Robust Layered Control System for a Mobile Robot". *IEEE Jour. of Robotics and Automation*, 2(1):14-23, 1986.

[9] Brooks, R., "Elephants Don't Play Chess", *Robotics and Autonomous Systems*, 6 (1990), pp. 3-15.

[10] Clark, R.J., Arkin, R.C., and Ram, A., "Learning Momentum: On-line Performance Enhancement for Reactive Systems", *Proc. 1992 IEEE International Conference on Robotics and Automation*, Nice, France, May 1992, pp. 111-116.

[11] Firby, R.J., "Adaptive Execution in Complex Dynamic Worlds", *Ph.D. Dissertation*, YALEU/CSD/RR #672, Yale University, Jan. 1989.

[12] J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.

[13] Gat, E., "Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture for Controlling Real-World Mobile Robots", *Proc. AAAI-92*, pp. 809-815, San Jose, CA, July 1992..

[14] Meystel, A., "Theoretical Foundations of Decision Making Processes for Design and Control of Intelligent Mobile Autonomous Robots", Technical Report, Drexel University, 1986.

[15] McFarland, D. and Bosser, T., *Intelligent Behavior in Animals and Robots*, MIT Press, 1993.

[16] Ram, A., Arkin, R.C., Moorman, K., and Clark, R., "Case-based Reactive Navigation: A case-based method for on-line selection and adaptation of reactive control parameters in autonomous robotic systems", Technical Report GIT-CC-92/57, Georgia Tech. 1992.

[17] f Ram, A., Arkin, R., Boone, G., and Pearce, M., "Using Genetic Algorithms to Learn Reactive Control Parameters for Autonomous Robotic Navigation", *Journal of Adaptive Behavior*, Vol. 2, No. 3, pp. 277-305, 1994.

[18] Weems, C., Riseman, E., and Hanson, A., "A Report on the Results of the DARPA Integrated Image Understanding Benchmark Exercise", *Proc. 1989 DARPA Image Understanding Workshop*, pp. 165-192.

A Appendix: Workshop Questions

In light of the material presented in the body of this paper, it is clear that I cannot make sweeping generalizations regarding the questions posed by the organizers. No context is provided for the environment in which the target systems are to be working, hence the responses are necessarily underconstrained. Instead of taking a dogmatic stance independent of the target domain, I will rather address the central issues concerning the questions raised and focus on how our own architectural considerations have been driven by these environments, focusing on real world applications wherever possible. In particular, these applications have included:

1. Low-level nuclear waste surveillance
2. Office cleaning robots
3. Unmanned aerial vehicles (for both contest tasks and military scout scenarios)
4. Teams of military scout unmanned ground vehicles
5. Mobile manipulators in manufacturing environments
6. Research navigation platforms

The questions the organizers raised and my responses follow:

- *Coordination* - *How should the agent arbitrate/coordinate/cooperate its behaviors and actions? Is there a need for central behavior coordination?*

  In what will seem to be repeated often, this question is pointless when posed without context. If we are to take a unified field theory approach (Sec. 3.1), clearly a stand would be made that one method is sufficient. I would contend, however, that the underpinnings of the architecture are heavily dependent upon the target domain, as stated earlier. If an architectural designer views coordination mechanisms as a carpenter would view his toolbox, flexible efficient designs could be created that would fit a particular niche (e.g., structured or unstructured, significant a priori knowledge or no, etc.). Whether or not there is a need for central behavior coordination would depend upon the particular problem being addressed. Perhaps what is most important for progress in this area, is rather than attempting to use a single hammer for all jobs, is to instead develop guidelines, or even better, formal methods for domain analysis, thus facilitating the mapping of particular architectural features to a problem. As Brooks' alluded to in one of his papers [9] elephants are not made to play chess, nor should they be. Extending this argument, robots that are geared to conduct surveillance tasks in structured environments (e.g., low-level nuclear waste) should not be constrained architecturally to those that scout in dynamic unstructured environments (e.g., ARPA UGV Demo II program). The physical attributes of these agents differ (locomotion and perception) as do their surroundings. What possible rationale is there to force them to operate with identical architectures if they will not ever share the same tasks?

- *Representation - How much internal representation of knowledge and skills is needed? How should the agent organize and represent its internal knowledge and skills? Is more than one representational formalism needed?*

  My position, unlike pure reactivists, has been that when knowledge is sound and reliable it should be made available to the system. The real question is not how much knowledge should be used or how should it be organized, but rather *when* should world knowledge be used? This relates more closely to a shifting between deliberative and reactive modes of execution and is a central question in robotics. When does one deliberate and take advantage of *a priori* knowledge? On the other hand, when does one just use sensory stimuli and react, or "Just do it"? Our systems have taken advantage of both strategies when appropriate through the use of a hybrid system. In general, when domains are more heavily structured and temporally consistent the utility of a priori knowledge increases. This is not an inviolate statement however, as even under these circumstances the cost of maintaining accurate fidelity with the world is often unnecessary Ii.e., environmental complexity as an additional factor). On the other hand, where myopia can be a problem, representational knowledge, coupled with suitable problem-solving techniques, (e.g., path planning, case-based reasoning or other methods) can often pay huge dividends, enabling solutions that may otherwise be infeasible. The issue of numbers of representational formalism relates closely to the epistemology of what is being represented. If there are qualitative differences in world knowledge (i.e., it is epistemologically distinct) then different formalisms will be necessary. Behavioral representations versus spatial knowledge are distinct and serve as one such example. I leave the remainder of this argument to the epistemologists amongst us.

- *Structural - How should the computational capabilities of an agent be divided, structured and interconnected? What is the best decomposition/granularity of architectural components? What is gained by using a monolithic architecture versus a multi-level, distributed, or massively parallel architecture? Are embodied semantics important and how should they be implemented? How much does each level/component of an agent architecture have to know about the other levels/components?*

  It is a mistake, in my estimation, to answer this question, without regard to the system as a whole, i.e., the agent, its surroundings, and its intentions. The main body of this position paper addresses this issue so the reader is referred to that for my response to these questions.

- *Performance - What types of performance goals and metrics can realistically be used for agents operating in dynamic, uncertain, and even actively hostile environments? How can an architecture make guarantees about its performance with respect to the time-critical criteria for deciding what activities take place in each level/component of the architecture?*

  These are important questions. Performance metrics are crucial for widespread acceptance of the research that is proceeding in most of our laboratories today. Proof-of-concept may be adequate for many academicians (often including myself) but for effective technology transfer, accurate performance assessment at a minimum and ideally performance guarantees need to be established and quantified. For dynamic domains this is notoriously difficult as they themselves resist modeling and characterization. One possibility is the establishment of a series of benchmarks along which to measure architectures similar to the lines of what the ARPA vision community accomplished a few years back [18]. Benchmarks however are notorious for their bias and unreliability in real world scenarios. An unfortunate side effect of robotic technology is that robotic systems, for their implementation, require more stringent performance standards than humans. It is unlikely that a robotic automobile could be certified in the same manner that a sixteen year old human driver is. Standard metrics (such as we have used [6] (time for task completion, efficiency, total distance covered, etc.) are not likely to be convincing when robots are to be fielded alongside humans. Performance criteria are less crucial when they take people out of harm's way (e.g., military scenarios) rather than place them potentially in it (an automated freeway UGV) despite the fact that performance guarantees are likely easier to be made in the latter due to the inherent structure of the environment. Optimality is generally not a concern for these domains as is reasonableness and timely task completion. The economic effectiveness of these systems is certainly significant and I refer the reader to McFarland's discussion of ecological niche-finding [15] for what I view as a promising analysis of what it takes for successful robotic performance assessment.

- *Psychology - Why should we build agents that mimic anthropomorphic functionalities? How far can/should we draw metaphoric similarities to human/animal psychology? How much should memory organization depend on human/animal psychology?*

  I have often relied on ethological/psychological/neuroscientific studies of animals for guidance in constructing robotic systems [3; 4; 5]. The answer to this questions relies on the goals of the individual researcher. There are some (e.g., [7]) who are using robotics to explore animal models of behavior in an attempt to validate them. My work on the other hand is centrally motivated to build intelligent machines, and the animal literature serves only as inspiration, not constraints, into how we construct our robotic agents. It is has been often said that biological systems serve as the existence proof for intelligent agents. I further believe that there are common principles which are evidenced in biological systems that can be utilized in silicon-based robotic systems. It is these principles that we seek in our own work. As I invariably explain during talks to my colleagues in the psychological communities, I choose rather to exploit and contort their underlying theories and models to fit the robotics domain, rather than to validate their own theories of biological behavior. This correctly appears to be largely a one-sided transaction, but nonetheless it has served my research well.

- *Simulation - What, if any role can advanced simulation technology play? (etc.)*

  Simulation, unfortunately, is a necessary evil. I lament those who spend ever increasing amounts of time in enhancing simulation environments, when actual robotic experiments are feasible. The old excuse that robotic hardware is too expensive is now gone with the proliferation of low-cost platforms that are commercially available. In certain circumstances though, due to the brittleness of most robotic systems, the time required, to often exercise robot systems thousands upon thousands of times for learning exercises and the like, simulation is still a necessary fallback. We use it frequently in our laboratory, more often than I prefer. A simulator at a minimum should use the same control code that drives the actual robot being simulated. This, from my perspective is of far greater concern than the actual world modeling. For those that use simulation, a tried and true migration path should be in evidence from simulations to real robotic systems in order to establish credibility. Those who rely on simulations alone can often be deceived into believing a result that is inaccurate or infeasible. There certainly is a large body of anecdotal evidence (ours as well) supporting this.

- *Learning - How can a given architecture support learning? How can knowledge and skills be moved between different layers of an agent architecture?*

  The basic question needing an answer here is *what and when* does something need to be learned (the credit assignment and saliency problems)? We have conducted extensive research in learning using on-line behavioral modification [10], case-based reasoning [16], and genetic algorithms [17]. The question as posed by the organizers cannot be answered without task and domain-specific information. Further, there is no short answer to the reposed question, esp. when taken as a generalization. Perhaps an interesting topic for a follow-up workshop?