Requirements & Designs: Asking Scientific Questions About Architectures

Nick Hawes, Aaron Sloman and Jeremy Wyatt
School of Computer Science
University of Birmingham
{n.a.hawes,a.sloman,j.l.wyatt}@cs.bham.ac.uk

Abstract

This paper discusses our views on the future of the field of cognitive architectures, and how the scientific questions that define it should be addressed. We also report on a set of requirements, and a related architecture design, that we are currently investigating as part of the CoSy project.

1 What Are Architectures?

The first problem we face as researchers in the field of cognitive architectures is defining exactly what we are studying. This is important because the term "architecture" is so widely used in modern technological fields. An agent's cognitive architecture defines the information-processing components within the "mind" of the agent, and how these components are structured in relation to each other. Also, there is a close link between architectures and the mechanisms and representations used within them (where representations can be of many kinds with many functions). Langley and Laird (2002) describe a cognitive architecture as including "those aspects of a cognitive agent that are constant over time and across different application domains". We extend this to explicitly allow architectures to change over time, either by changing connection patterns, or altering the components present. Excluding such changes from the study of architectures may prevent the discussion of the development of architectures for altricial information-processing systems (Sloman and Chappell, 2005).

2 Related Work

Historically, most research into cognitive architectures has been based around specific architectures such as ACT-R, SOAR, and ICARUS (for a summary see (Langley and Laird, 2002)). A lot of work has been devoted to developing iterations of, and extensions to, these architectures, but very little work has been done to compare them, either to each other, or to other possible design options for cognitive architectures. In other words, little work has been done on the general science of designing and building cognitive systems. Anderson and Lebiere (2003) have recently attempted to address this by comparing two different architectures for human cognition on a set of requirements.

3 Architectures & Science

To advance the science of cognitive systems we need two related things: clear, testable questions to ask, and a methodology for asking these questions. The methodology we support is one of studying the space of possible *niches* and *designs* for architectures, rather than single, isolated, designs (Sloman, 1998b). Within such a framework, scientific questions can be asked about how a range of architecture designs relate to sets of requirements, and the manner in which particular designs satisfy particular niches. Without reference to the requirements they were designed to satisfy, architectures can only be evaluated in a conceptual vacuum.

The scientific questions we choose to ask about the space of possible architecture designs should ideally provide information on general capabilities of architectures given a set of requirements. This information may not be particularly useful if it is just a laundry list of instructions for developing a particular architecture for a particular application domain. It will be more useful if we can characterise the space of design options related to a set of requirements, so that future designers can be aware of how the choices they make will affect the overall behaviour of an agent. The questions asked about architectures can be motivated by many sources of information, including competing architecture designs intended for similar niches.

In order for questions about architectures, and their answers, to be interpreted in the same way

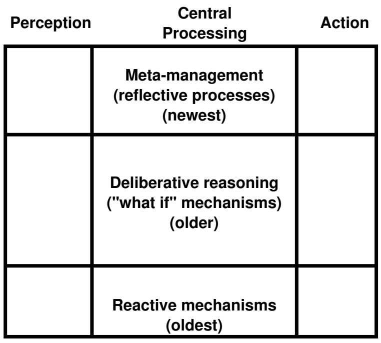

Figure 1: The CogAff Architecture Schema.

by researchers across many disciplines, we need to establish a common vocabulary for the design of information-processing architectures. As a step towards this, we use the CogAff schema, depicted in Figure 1, as an incomplete first draft of an ontology for comparing architectures. (Sloman, 2001). The schema is intended to support broad, two-dimensional, design- and implementation-neutral characterisations of architectural components, based on information-processing style and purpose. If an architecture is described using the schema, then it becomes easier to compare it directly to other architectures described in this way. This will allow differing architectures to be compared along similar lines, even if they initially appear to have little in common.

4 A Minimal Scenario

For our current research as part of the CoSy project[1], we are working from requirements for a pre-linguistic robot that has basic manipulative abilities, and is able to explore both its world and its own functionality. At a later date we will extend this to add requirements for linguistic abilities. We are approaching the problem in this way because we believe that a foundation of action competence is necessary to provide semantics for language. These requirements come from the CoSy *PlayMate scenario*, in which a robot and a human interact with a tabletop of objects to perform various tasks[2].

In our initial work on this scenario we will focus on the requirements related to the architectural elements necessary to support the integration of simple manipulative abilities with a visual system that supports the recognition of basic physical affordances from 3D structure. We see this as the absolute minimum system for the start of an exploration of PlayMate-like issues in an implemented system [3].

Our requirements analysis has led to the design of a prototype architecture which we believe will satisfy the niche they specify. Space restrictions do not permit a full description of the architecture, but in brief the architecture features multiple concurrently active components, including: a motive generator; information stores for currently active motives, general concepts, and instances of the general concepts; a general-purpose deliberative system; a fast global alarm system; a plan execution system; management and meta-management components; a spreading activation substrate; and closely coupled vision and manipulation sub-architectures.

The high-level design for this architecture is presented in Figure 2, and is in part inspired by our previous work on information-processing architectures (e.g. (Beaudoin, 1994; Sloman, 1998a; Hawes, 2004)). Although this design clearly separates functionality into components, these components will be tightly integrated at various levels of abstraction. For example, to enable visual servoing for manipulation (e.g. (Kragic and Christensen, 2003)), visual and proprioceptive perception of the movement of the robot's arm in space must be closely coupled with the instructions sent to the arm's movement controller.

The information-processing behaviour of the architecture is driven by motives, which are generated in response to environmental or informational events. We will allow humans to generate environmental events using a pointing device. The agent will interpret the gestures made with this device as direct indications of desired future states, rather than intentional acts (thus temporarily side-stepping some of the problems of situated human-robot interaction). Generated motives will be added to a collection of current motives, and further reasoning may be necessary if conflicts occur between motives. The deliberative system will produce action plans from motives, and these plans will be turned into arm commands by the plan execution system. This process will be observed at a high level by a meta-management system, and at a low level by an alarm system. The meta-management system may reconfigure the agent's processing strategies if the situation requires it (e.g. by altering the priorities associated with motives). The global alarm system will provide fast changes in behaviour to handle sudden, or particularly critical, situations.

[1] See http://www.cognitivesystems.org for more information.

[2] More information about the PlayMate is available at http://www.cs.bham.ac.uk/research/projects/cosy/pm.html.

[3] Our work on requirements from the PlayMate scenario is presented roughly at http://snipurl.com/cosy_playmate.

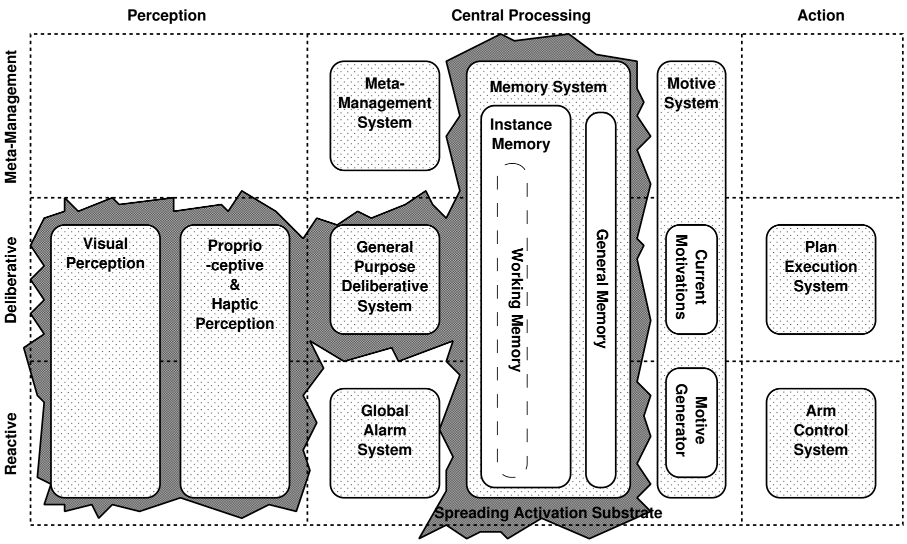

Figure 2: The Proposed Architecture.

Although a spreading activation substrate is featured in the design, we are not currently committed to its inclusion in the final system. Instead we are interested in the kinds of behaviour that such a design choice will facilitate. Information across the architecture may need to be connected to related information, and such an approach may allow the agent to exploit the structure of such connections by spreading activation, which may be based on co-occurrence, recency or saliency. We are also interested in investigating how to combine distributed approaches to representing and processing information with more localised approaches, and what design options this provides. Such a combination of processing approaches can be seen in the work on the MicroPsi agent architecture (Bach, 2005).

5 Architecture Evaluation

The question of whether, and why, our proposed architecture is appropriate for an agent with PlayMate-like requirements is quite hard to formulate in a way that is directly answerable. Instead, we must use our requirements analysis, and our previous experiences of designing architectures for such agents, to derive precise questions and suggest testable hypotheses from this. The following paragraphs present specific questions we could ask about the architecture, and many other architectures.

How can information exchange between architectural components be controlled, and what trade-offs are apparent? For example, should information from visual perception be pushed into a central repository, or should task appropriate information be pulled from vision when necessary, or should this vary depending on the system's information state, goals, the performance characteristics of subsystems, etc.?

What are the relative merits of symbolic and sub-symbolic (e.g. spreading activation) processing methods when applied to collating information across the entire architecture? The proposed spreading activation substrate could interact with various processes and ontologies, and record how information is manipulated. Alternatively, this could be implemented as a central process that must be notified by other processes when certain operations occur. These different approaches could be compared on their proficiency at managing large volumes of multi-modal information, their ability to identify changes of context across the architecture, the difficulty of integrating them with other processes, or the ease with which they facilitate other operations (such as attentional control).

To what degree should the architecture encapsulate modality-specific and process-specific information within the components that are directly concerned with it? Cross-modal application of the early processing results can increase accuracy and efficiency in some processes (c.f. (Roy and Mukherjee, 2005)). In other cases information may be irrelevant, and attempts to apply it across modalities may have the opposite effect whilst increasing the computational load on an architecture. We could explore this notion more generally by asking what types of information should, and should not, be made available by architectural components whilst they are processing it, and what use other architectural components could make of such information.

Given the types of information the architecture will be processing, what are the advantages and disadvantages of having a single central representation into which all information is translated? How do these advantages and disadvantages change when additional processes are added into the architecture?

What role does a global alarm mechanism have in PlayMate-like domains, how much information should it have access to, and how much control should it have? For example, an alarm mechanism may have access to all the information in the architecture and risk being swamped by data, or it may have access to limited information streams and risk being irrelevant in many situations.

Does the architecture need some global method for producing serial behaviour from its many concurrently active components, or will such behaviour just emerge from appropriate inter-component interactions? Approaches to component control include a single central component activating other components, a control cycle in which activity is passed between a small number of components, and other variations on this. Are there particular behaviours that are not achievable by an agent with this kind of control, and only achievable by an agent we decentralised control, or vice versa? If such trade-offs exist, how are the relevant to PlayMate-like scenarios?

Given the range of possible goals that will need to be present in the whole system, how should these goals be distributed across its architecture, and how does this distribution affect the range of behaviours that the system can display?

Obviously there are many other questions we could ask about the architecture, such as whether it will facilitate the implementation of mechanisms for acquiring and using orthogonal recombinable competences[4]. The process of designing and implementing architectures to meet a set of requirements involves the regular re-evaluation of the requirements in light of new developments. Inevitably, this means that other questions will be considered, and the above ones reconsidered, as the research progresses.

Acknowledgements

The research reported on in this paper was supported by the EU FP6 IST Cognitive Systems Integrated project Cognitive Systems for Cognitive Assistants "CoSy" FP6-004250-IP.

[4]http://www.cs.bham.ac.uk/research/projects/cosy/papers/#dp0601

References

John R. Anderson and Christian Lebiere. The Newell test for a theory of cognition. *Behavioral & Brain Science*, (26):587–637, 2003.

Joscha Bach. Representations for a complex world: Combining distributed and localist representations for learning and planning. In Stefan Wermter, Günther Palm, and Mark Elshaw, editors, *Biomimetic Neural Learning for Intelligent Robots*, volume 3575 of *Lecture Notes in Computer Science*, pages 265–280. Springer, 2005.

Luc P. Beaudoin. *Goal Processing In Autonomous Agents*. PhD thesis, School of Computer Science, The University of Birmingham, 1994.

Nick Hawes. *Anytime Deliberation for Computer Game Agents*. PhD thesis, School of Computer Science, University of Birmingham, 2004.

Danica Kragic and Henrik I. Christensen. A framework for visual servoing. In Markus Vincze and James L. Crowley, editors, *ICVS-03*, volume 2626 of *LNCS*. Springer Verlag, March 2003.

Pat Langley and John E. Laird. Cognitive architectures: Research issues and challenges. Technical report, Institute for the Study of Learning and Expertise, Palo Alto, CA, 2002.

Deb Roy and Niloy Mukherjee. Towards situated speech understanding: Visual context priming of language models. *Computer Speech and Language*, 19(2):227–248, 2005.

Aaron Sloman. Varieties of affect and the cogaff architecture schema. In *Proceedings of the AISB'01 Symposium on Emotion, Cognition and Affective Computing*, pages 1–10, 2001.

Aaron Sloman. Damasio, descartes, alarms and metamanagement. In *Proceedings of IEEE Conference on Systems, Man, and Cybernetics*, pages 2652–2657, 1998a.

Aaron Sloman. The "semantics" of evolution: Trajectories and trade-offs in design space and niche space. In Helder Coelho, editor, *Progress in Artificial Intelligence, 6th Iberoamerican Conference on AI (IBERAMIA)*, pages 27–38. Springer, Lecture Notes in Artificial Intelligence, Lisbon, October 1998b.

Aaron Sloman and Jackie Chappell. The Altricial-Precocial Spectrum for Robots. In *Proceedings IJCAI'05*, pages 1187–1194, Edinburgh, 2005.