3
Cognitive Architectures

3.1 What Is a Cognitive Architecture?

When we think of architecture, typically what comes to mind is the design of buildings that satisfy some functional need but do so in a way that appeals to the people that use them. Often, architecture inspires some sense of the extraordinary and gives an impression of cohesion that makes the building whole and self-contained. Since the architectural process involves not just imagining bold new concepts but also the creation of detailed designs and technical specifications, architecture has been borrowed by many other desciplines to serve as a catch-all term for the technical specification and design of any complex artifact. Just as with architecture in the built environment, system architecture addresses both the conceptual form and the utilitarian functional aspects of the system, focussing on inner cohesion and self-contained completeness.

We use the term cognitive architecture in exactly this way to reflect the specification of a cognitive system, its components, and the way these components are dynamically related as a whole.

One of the most famous maxims in architecture, and in design generally, is that "form follows function,"[1] the principle that the shape of a building, or any object, should be mainly based on its intended purpose or function. However, in contemporary architecture this is interpreted very broadly to include both utility and aesthetic value: the degree to which it engenders a posi-

[1] The idea that form follows function is due to the nineteenth century architect Louis Sullivan.

tive interaction between people and the building and the degree to which the building is perceived as a complete entity. As we noted in the previous chapter, interaction plays a key role in cognition (and *vice versa*) so this broad interpretation of architecture is very apt when it comes to cognitive architecture: the system the architecture describes must work both at a global system level, enabling the effective interaction of a cognitive agent with the world around it, and at a component level, showing how all the parts should fit together to create the global system: the cohesive whole.

Just as there are different styles and traditions in traditional architecture, each emphasizing different facets of form and function, so too there are many different styles of cognitive architecture, each derived, more or less directly, from the three paradigms of cognitive science we discussed in the previous chapter: the cognitivist, the emergent, and the hybrid. However, the term cognitive architecture can in fact be traced to pioneering work in cognitivist cognitive science.[2] Consequently, it means something very specific in cognitivism. In particular, a cognitive architecture represents any attempt to create what is referred to as a unified theory of cognition.[3] This is a theory that covers a broad range of cognitive issues, such as attention, memory, problem solving, decision making, and learning. Furthermore, a unified theory of cognition should cover these issues from several aspects including psychology, neuroscience, and computer science. Allen Newell's and John Laird's Soar[4] architecture, John Anderson's ACT-R[5] architecture, and Ron Sun's CLARION architecture are typical candidate unified theories of cognition.[6]

Since unified theories of cognition are concerned with the computational understanding of *human* cognition, cognitivist cognitive architectures are concerned with human cognitive science as well as artificial cognitive systems. There is an argument that the term cognitive architecture should be reserved for systems that model human cognition and that the term "agent architecture" would be a better term to refer to general intelligent behaviour, including both human and computer-based artificial cognition. However, it has become common to use the term cognitive architecture in this more general sense so we will

[2] The term cognitive architecture is due to Allen Newell and his colleagues in their work on unified theories of cognition [41, 43].

[3] Unified theories of cognition are discussed in depth in Allen Newell's book of the same name [43] and John Anderson's paper "An integrated theory of the mind" [109].

[4] For more details on the Soar cognitive architecture, please refer to the papers by Allen Newell, John Laird, and colleagues [42, 110, 111, 112, 113], John Laird's book [114], and read Section 3.4.1 in this chapter.

[5] For more details on the ACT-R cognitive architecture, please refer to [109, 115].

[6] The CLARION cognitive architecture is described in depth in, e.g., [116, 117].

use it throughout the book to refer to both human and artificial cognitive systems.

Although the term cognitive architecture originated in cognitivist cognitive science, it has also been adopted in the emergent paradigm where it has a sightly different meaning. Consequently, we will begin by considering exactly what a cognitive architecture does involve in the two different approaches: cognitivist and emergent. Following that, we will discuss the features of a cognitive architecture that are considered to be necessary and desirable. Finally, we will look at three specific cognitive architectures — one from the cognitivist paradigm of cognitive science, one from the emergent, and one from the hybrid paradigm — in different levels of detail to get some understanding of what they involve and the role they play in the design of a working cognitive system.

3.1.1 The Cognitivist Perspective

In the cognitivist paradigm, the focus in a cognitive architecture is on the aspects of cognition that are *constant over time* and that are *independent of the task*.[7] In the words of Ron Sun, a leading exponent of cognitive architectures, [17]:

> "a cognitive architecture is a broadly-scoped domain-generic computational cognitive model, capturing the essential structure and process of the mind, to be used for broad, multiple-level, multiple-domain analysis of behaviour."

[7] The idea that a cognitive architecture focusses on those aspects of cognition that are constant over time and independent of the task, i.e. unchanging from situation to situation, is widely supported in the literature; for example, see [118, 119, 120, 121].

Since a cognitive architecture represents the fixed part of cognition, it cannot accomplish anything in its own right. A cognitivist cognitive architecture is a generic computational model that is neither domain-specific nor task-specific. To do something, i.e. to perform a given task, it needs to be provided with the knowledge to perform any given task. It is the knowledge which populates the cognitive architecture that provides the means to perform a task or to behave in some particular way. This combination of a given cognitive architecture and a particular knowledge set is generally referred to as a *cognitive model*.

So, where does this knowledge come from? In most cognitivist systems the knowledge incorporated into the model is normally determined by the person who designed the architecture, and often this knowledge is highly crafted, possibly drawing on years of experience working in the problem domain. Machine learning is increasingly used to augment and adapt this knowledge but typically you need to provide a critical minimum amount of knowledge in order to get the learning started.

The cognitive architecture itself determines the overall structure and organization of a cognitive system, including the component parts or modules, the relations between these modules, and the essential algorithmic and representational details within them. The architecture specifies the formalisms for knowledge representations and the types of memories used to store them, the processes that act upon that knowledge, and the learning mechanisms that acquire it. Usually, it also provides a way of programming the system so that a cognitive system can be customized for some application domain.

A cognitive architecture plays an important role in computational modelling of cognition in that it makes explicit the set of assumptions upon which that cognitive model is founded. These assumptions are typically derived from several sources: biological or psychological data, philosophical arguments, or *ad hoc* working hypotheses inspired by work in different disciplines such as neurophysiology, psychology, or artificial intelligence. Once it has been created, a cognitive architecture also provides framework for developing the ideas and assumptions encapsulated in the architecture.

3.1.2 The Emergent Perspective

Emergent approaches to cognition focus on the development of the agent from a primitive state to a fully cognitive state over its life-time. Although the concept of a cognitive architecture has its origins in cognitivism as the timeless fixed part of a cognitive system that provides the framework for processing knowledge, the term cognitive architecture is also used with emergent approaches. In this case, it isn't so much the framework that com-

plements the knowledge as it is the framework that facilitates development. In this sense, an emergent cognitive architecture is essentially equivalent to the phylogenetic configuration of a newborn cognitive agent: the initial state from which it subsequently develops. In other words, an emergent cognitive architecture is everything a cognitive system needs to get started. This doesn't guarantee successful development, though, because development also requires exposure to an environment that is conducive to development, one in which there is sufficient regularity to allow the system to build a sense of understanding of the world around it, but not excessive variety that would overwhelm an agent which has inherent limitations on the speed with which it can develop. Thus, in a way that parallels the two-sided coin of cognitivist cognition — architecture and knowledge — emergent cognition also has two sides: architecture and gradually-acquired experience. These two sides of the emergent coin are referred to as phyogeny and ontogeny (or ontogenesis), the latter being the interactions and experiences that a developing cognitive system is exposed to as it acquires an increasing degree of cognitive capability.

With emergent approaches, the cognitive architecture provides a way of dealing with the intrinsic complexity of a cognitive system, by providing some form of structure within which to embed the mechanisms for perception, action, adaptation, anticipation, and motivation that enable the ontogenetic development over the system's life-time. It is this complexity that distinguishes an emergent developmental cognitive system from, for example, a connectionist system such as an artificial neural network that performs just one or two functions such as recognition or control. Of course, an emergent cognitive architecture might comprise many individual neural networks and, as we will see later, some do.

So, the cognitive architecture of an emergent system thus provides the basis for its subsequent development. It's worth remarking that, as a consequence of this development, the architecture itself might change. Thus, an emergent cognitive architecture isn't necessarily fixed and timeless: it is a point of departure.

The presence of innate capabilities in an emergent system does not imply that the architecture is necessarily functionally modular, *i.e.* that the cognitive system is comprised of distinct modules each one carrying out a specialized cognitive task.[8] If modularity is present, it may be because it develops this modularity through experience as part of its ontogenesis rather than being prefigured by the phylogeny of the system. The cognitivist and emergent perspectives differ somewhat on the issue of innate structure. While in an emergent system the cognitive architecture *is* the innate structure, this is not necessarily so with a cognitivist system.[9]

Sometimes, especially in developmental robotics, the term epigenesis is used instead of ontogensis, and developmental robotics is sometimes referred to as epigenetic robotics. Epigenesis has its roots in biology where it refers to the way an organism develops through cell-division into a a viable complex entity. This happens through gene expression so that the epigenesis refers to the changes that result from factors other than those determined by the organism's DNA. Ontogenesis also refers to early cellular development but more generally it refers to the development of the organism *over its full lifetime*. Thus, it includes the development of the agent after birth, including its cognitive development, and so embraces, for example, developmental psychology. Since the epigenetic process focusses exclusively on the very early growth of the agent and the way its final structure is determined, in artificial cognitive systems, epigenesis would probably be better reserved to reflect the autonomous formation and construction of cognitive architecture prior to development as a consequence of experience. To avoid confusion, we will avoid using the term epigenesis and epigenetic robotics, and refer to ontogeneis and developmental robotics on the understanding that we are discussing the development of an entity after it has been born (in the case of natural cognitive systems) or realized as a physical system (in the case of artifical cognitive systems). For the most part, we won't discuss the issue of how a cognitive architecture might emerge or develop prior to this point, although, as we will see, the configuration of a complete emergent cognitive architecture isn't a straightforward task and

[8] Heinz von Foerster argues that the constituents of a cognitive architecture cannot be separated into distinct functional components: "In the stream of cognitive processes one can conceptually isolate certain components, for instance (i) the faculty to perceive, (ii) the faculty to remember, and (iii) the faculty to infer. But if one wishes to isolate these faculties functionally or locally, one is doomed to fail. Consequently, if the mechanisms that are responsible for any of these faculties are to be discovered, then the totality of cognitive processes must be considered." [122], p. 105.

[9] Ron Sun contends that "an innate structure can, but need not, be specified in an initial architecture" [117]. He argues that an innate structure does not have to be specified or involved in the computational modelling of cognition and that architectural detail may indeed result from ontogenetic development. However, he suggests that non-innate structures should be avoided as much as possible and that we should adopt a minimalist approach: an architecture should include only minimal structures and minimal learning mechanisms which should be capable of "bootstrapping all the way to a full-fledged cognitive model."

it is conceivable that epigenetic considerations might be able to shed some light on the matter.

Finally, we remind ourselves that the emergent paradigm rejects the position that cognitivism takes on two key issues: the dualism that separates the mind and body and treats them as distinct entities and the functionalism that treats cognitive mechanisms independently of the physical platform. The logical separation of mind and body, and of mechanism and physical realization, means that cognition can, in principle, be studied in isolation from the physical system in which it occurs. The emergent paradigm takes the opposite view, holding that the physical system — the body — is just as much a part of the cognitive process as are the cognitive mechanisms in the brain. Consequently, an emergent cognitive architecture will ideally reflect in some way the structure and capabilites — the morphology — of the physical body in which it is embedded and of which it is an intrinsic part. We consider these aspects in detail in Chapter 5 on embodiment.

3.2 *Desirable Characteristics*

When we say that an emergent cognitive architecture *ideally* reflects the form and capabilities of its associated physical body, we recognize that very few, if any, current cognitive architectures have managed to do this. There is a gap at present between what we know a cognitive architecture should be and what in fact existing architectures have managed to achieve. In this section, we focus on the ideal features of a cognitive architecture.

3.2.1 *Realism*

We begin with some features related to the realism of the architecture. Since a cognitivist cognitive architecture represents a unified theory of cognition, and hence a theory of human cognition, it should strive to exhibit several types of realism.[10]

First, it should enable the cognitive agent to operate in its natural environment, engaging in "everyday activities." This is referred to as *ecological realism*. It means that the architecture

[10] These different types of realism — ecological, bio-evolutionary, cognitive — as well as several other desirable characteristics of a cognitive architecture are described by Ron Sun in his paper "Desiderata for Cognitive Architectures" [117].

has to deal with many concurrent and often conflicting goals in an environment about which the agent probably doesn't know everything. In fact, that's exactly the point of cognition: being able to deal with these uncertainties and conflicts in a way that still gets the job done, whatever it is. So, ecological realism goes to the very heart of cognition.

Second, since human intelligence evolved from the capabilities of earlier primates, ideally a cognitive model of human intelligence should be reducible to a model of animal intelligence. This is *bio-evolutionary realism*. Sometimes, this is taken the other way around by focussing on simpler models of cognition as exhibited by other species — birds and rats, for example — and then attempting to scale them up to human-level cognition.

Third, a cognitive architecture should capture the essential characteristics of human cognition from several perspectives: psychology, neuroscience, and philosophy, for example. This is referred to as *cognitive realism*. To an extent, this means that the cognitive architecture, and the overall cognitive model of which it is an essential part, should be complete.

Finally, as with all good science, new models should draw on, subsume, or supercede older models (this means that a cognitive architecture should strive for *inclusivity of prior perspectives*[11]).

[11] Ron Sun refers to this inclusivity as "eclecticism of methodologies and techniques" [117].

3.2.2 *Behavioural Characteristics*

Several behavioural and cognitive characteristics should ideally be captured by a cognitive architecture and exhibited by a cognitive system.[12] From a behavioural perspective, a cognitive architecture should not have to employ excessively complicated conceptual representations and extensive computations devoted to working through alternative strategies. The cognitive system should behave in a direct and immediate manner, making decisions and acting in an effective and timely manner. Furthermore, a cognitive system should operate one step at a time, in a sequence of actions extended over time. This gives rise to the desirable characteristic of being able to learn routine behaviours gradually, either by trial-and-error or by copying other cognitive agents.

[12] Again, these ideal behavioural and cognitive characteristics are described by Ron Sun in his paper "Desiderata for Cognitive Architectures" [117].

3.2.3 *Cognitive Characteristics*

As far as cognitive characteristics are concerned, a cognitive architecture should comprise two distinct types of process: one explicit, the other implicit. The explicit processes are accessible and precise whereas the implicit ones are inaccessible and imprecise. Furthermore, there should be a synergy borne of interaction between these two types of process. There are, for example, explicit and implicit learning processes and these interact.[13] To a significant extent, these cognitive characteristics reflect a hybrid approach to cognition: strict emergent approaches would not be able to deliver on the requirement for accessibility, which cognitivist approaches most certainly would. At the same time, not all cognitive architectures make use of implicit processes, such as connectionist learning, although there is an increasing trend to do so, as we will see below when we survey three current cognitive architectures.

[13] These cognitive characteristics are reflected in Sun's own cognitive architecture CLARION [116, 17], in which implicit processes operate on connectionist representations and explicit processes on symbolic representations (thus, CLARION is a hybrid cognitive architectures).

3.2.4 *Functional Capabilities*

In fulfilling these roles, an ideal cognitive architecture should ideally exhibit several functional capabilities.[14]

A cognitive architecture should be able to recognize objects, situations, and events as instances of known patterns and it must be able to assign them to broader concepts or categories. It should also be able to learn new patterns and categories, modify existing ones, either by direct instruction or by experience.

Since a cognitive architecture exists to support the actions of a cognitive agent, it should provide a way to identify and represent alternative choices and then decide which are the most appropriate and select an action for execution. Ideally, it should be able to improve its decisions through learning.

It should have some perceptual capacity — vision, hearing, touch, for example[15] — and, since a cognitive agent typically has limited resources for processing information, it should have an attentive capacity to decide how to allocate these resources and to detect what is immediately relevant.

A cognitive architecture should also have some mechanism to predict situations and events, i.e. to anticipate the future. Often,

[14] Pat Langley, John Laird, and Seth Rogers [120] catalogue nine functional capabilities that should be exhibited by an ideal cognitive architecture. Although they focus mainly on cognitivist cognitive architectures in their examples, the capabilities they discuss also apply for the most part to emergent systems. Ron Sun lists a similar list of twelve functional capabilities [17].

[15] There are two categories of perception: exteroception and proprioception. Exteroception includes all those modalities which sense the external world, such as vision, hearing, touch, and smell. Proprioception is concerned with sensing the status or configuration of the agent's body; whether an arm is extended and by how much, for example.

this ability will be based on an internal model of the cognitive agent's environment. Ideally, a cognitive architecture should have a mechanism to learn these models from experience and improve them over time.

To achieve goals, it must have some capability to plan actions and solve problems. A plan requires some representation of a partially-ordered sequence of actions and their effects. Incidentally, problem solving differs from planning in that it may also involve physical change in the agent's world.

The knowledge that complements a cognitive architecture constitutes the agent's beliefs about itself and its world, and planning is focussed on using this knowledge to effect some action and achieve a desired goal. The cognitive architecture should also have a reasoning mechanism which allows the cognitive system to draw inferences from these beliefs, either to maintain the beliefs or to modify them.

A cognitive architecture should have some mechanism to represent and store motor skills that can be used in the execution of an agent's actions. As always, an ideal cognitive architecture will have some way of learning these motor skills from instruction or experience.

It should be able to communicate with other agents so that they can obtain and share knowledge. This may also require a mechanism for transforming the knowledge from internal representations to a form suitable for communication.

It may also be useful for a cognitive archtitecture to have additional capabilities which are not strictly necessary but which may improve the operation of the cognitive agent. These are referred to as meta-cognition (sometimes called meta-management) functions and they are concerned with remembering (storing and recalling) the agent's cognitive experiences and reflecting on them, for example, to explain decisions, plans, or actions in terms of the cognitive steps that led to them.[16]

Finally, an ideal cognitive architecture should have some way of learning to improve the performance of all the foregoing functions and to generalize from specific experiences of the cognitive system.[17]

In summary, an ideal cognitive architecture supports at least

[16] For more details on the importance of meta-management in cognitive architectures, see Aaron Sloman's paper "Varieties of affect and the CogAff architecture schema" [16].

[17] By generalizing from specific experiences, the cognitive agent is engaging in inductive inference.

the following nine functional capabilities:

1. Recognition and categorization;
2. Decision making and choice;
3. Perception and situation assessment;
4. Prediction and monitoring;
5. Problem solving and planning;
6. Reasoning and belief maintenance;
7. Execution and action;
8. Interaction and communication;
9. Remembering, reflection, and learning.

This list is not exhaustive and one could add other functionalities: the need for multiple representations, the need for several types of memory, and the need to have different types of learning, for example. We discuss these issues in Chapters 6 and 7.

3.2.5 Development

One thing should strike you about the list above: it doesn't explictly address development. That's because, for the most part, it results from research in cognitivist cognitive architectures. For emergent cognitive architecture that focus on development, Jeffrey Krichmar has identified several desirable characteristics.[18] First, he suggests that the architecture should address the dynamics of the neural element in different regions of the brain, the structure of these regions, and especially the connectivity and interaction between these regions. Second, he notes that the system should be able to effect perceptual categorization: i.e. to organize unlabelled sensory signals of all modalities into categories without prior knowledge or external instruction. In effect, this means that the system should be autonomous and, as a developmental system, it should be a model generator, rather than a model fitter.[19] Third, a developmental system should have a physical instantiation, i.e. it should be embodied, so that it is tightly coupled with its own morphology and so that it can explore its environment. Fourth, the system should engage in some behavioural task and, consequently, it should have some minimal

[18] While not specifically targetting cognitive architectures, Jeffrey Krichmar's design principles for developmental artificial brain-based devices [123, 124, 125] are directly applicable to emergent systems in general.

[19] The distinction between model generation and model fitting in cognitive systems is also emphasized by John Weng in his paper "Developmental Robotics: Theory and Experiments" [126].

set of innate behaviours or reflexes in order to explore and survive in its initial environmental niche. From this minimum set, the system can learn and adapt so that it improves its behaviour over time. Fifth, developmental systems should have a means to adapt. This implies the presence of a value system, i.e. a set of motivations that guide or govern its development.[20] These should be non-specific[21] modulatory signals that bias the dynamics of the system so that the global needs of the system are satisfied: in effect, so that the system's autonomy is preserved or enhanced.

[20] For an overview of the role of value systems in cognitive systems, see the paper by Kathryn Merrick "A Comparative Study of Value Systems for Self-motivated Exploration and Learning by Robots" [127] and the paper by Pierre-Yves Oudeyer, Frédéric Kaplan, and Verena Hafner "Intrinsic motivation systems for autonomous mental development" [128].

[21] Non-specific in the sense that they don't specify what actions to take.

3.2.6 Dynamics

It is clear that a cognitive system is going to be a very complex arrangement of components parts. After all, that's why an architecture is necessary in the first place. However, there is more to an architecture than just its components: there is also the manner in which they are connected with one another and the dynamic behaviour of the various components as they interact with one another and as the agent interacts with its environment. A cognitive architecture needs to be complex enough to capture these dynamics without being excessively complicated. It should incorporate only what is necessary without compromising its eco-realism. Clearly, this is a difficult balance to get right and, as we mentioned above, very few cognitive architectures fully support all of the desired characteristics at present.[22] Many challenges remain and there is a long list of issues where our understanding is inadquate. Example include[23] understanding the mechanisms for selective attention, the processes for categorization, developing support for episodic memory and processes to reflect on it, developing support for multiple knowledge representation formalisms, the inclusion of emotion in cognitive architectures to modulate cognitive behaviour, and the impact of physical embodiment on the overall cognitive process, including the agent's internal drives and goals.

[22] See Ron Sun's paper "The importance of cognitive architectures: an analysis based on CLARION" [17] for a more extended discussion of the degree to which contemporary cognitive architectures exhibit the desirable characteristics of an ideal architecture.

[23] This list of research challenges is taken from the paper by Pat Langley, John Laird, and Seth Rogers "Cognitive architectures: Research issues and challenges" [120].

3.3 *Designing a Cognitive Architecture*

Before we move on to look at some cognitive architectures that have been developed in recent years, we will first say a few words about how one might go about designing one. Given the apparent complexity of a cognitive architecture, the long list of desirable characteristics set out above, and the many research challenges we still face, it should be evident that this is not a simple matter. However, a relatively straight-forward three-step process has been proposed by Aaron Sloman and his co-workers.[24] First, the requirements of the architecture needs to be identified, partly through an analysis of several typical scenarios in which the eventual agent would demonstrate its competence. These requirements are then used to create an *architecture schema*: a task- and implementation-independent set of rules for structuring processing components and information, and controlling information flow. This schema leaves out much of the detail of the final design choices, detail which is finally filled in at the third step by an instatiation of the architecture schema in a cognitive architecture proper on the basis of a specific scenario and its attendant requirements. This process is particularly suited to cognitivist cognitive architectures because it emphasizes the logical division of task-independent processing mechanisms and structure from task-dependent knowledge.

3.4 *Example Cognitive Architectures*

For the remainder of the chapter, the term cognitive architecture will be used in a general sense without specific reference to the underlying paradigm, cognitivist or emergent. By this we interpret it to mean the minimal configuration of a system that is necessary for the system to exhibit cognitive capabilities and behaviours, i.e. the specification of the components in a cognitive system, their function, and their organization as a whole.

In the following, we will provide a brief overview of a sample of three cognitive architectures, one from the cognitivist paradigm of cognitive science (Soar), one from the emergent (Darwin), and one from the hybrid paradigm (ISAC).[25]

[24] The three-step process for designing a cognitive architecture is discussed in a technical report by Nick Hawes, Jeremy Wyatt, and Aaron Sloman "An architecture schema for embodied cognitive systems" [129].

[25] There are many other cognitive architectures in all three paradigms: cognitivist, emergent, and hybrid. These include for example ACT-R [109, 115], CoSy Architecture Schema [129, 130], GLAIR [131], ICARUS [132, 133] (cognitivist); Cognitive-Affective Architecture Schematic [134, 135], Global Workspace [101], iCub [136, 137], SASE [126, 138] (emergent); and CLARION [116, 17], HUMANOID [139], LIDA [140, 141], PACO-PLUS [142] (hybrid). On-line surveys of cognitive architectures can be found on the website of the Biologically Inspired Cognitive Architectures Society [143] and on the website of the University of Michigan [45]. Surveys published in the literature include an overview published by the author, Claes von Hofsten, and Luciano Fadiga in "A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents" [103] and updated in *A Roadmap for Cognitive Development in Humanoid Robots* [12], and a survey by Włodzisław Duch and colleagues "Cognitive Architectures: Where do we go from here?" [144].

3.4.1 Soar

Soar[26] is a candidate Unified Theory of Cognition and, as such, it is a quintessential cognitivist cognitive architecture. It is also an iconic one, being one of the very first cognitive architectures to be developed. Futhermore, it was created by Allen Newell (the person who introduced the idea of a unified theory of cognition) and his colleagues, and has been continually enhanced over the past 25 years or so. Hence, Soar occupies a special place in the history of cognitive architectures and their continuing evolution. As we will see, the themes raised by Soar are reflected in several other cognitive architectures.

[26] For more details on the Soar cognitive architecture, please refer to the papers by Allen Newell, John Laird, and colleagues [42, 110, 111, 112, 113] and the book by John Laird [114].

We will begin by reminding ourselves of the key ideas underpinning cognitivism. It is important to do this because Soar was built on these and the way it operates refects the fundamental assumptions of cognitivism. We will then give a very brief sketch of the way Soar operates, just to get a feeling for the way it works.

As we have already said, in cognitivism a cognitive architecture represents the aspects of cognition that are constant over time and independent of the task. To do something, i.e. to perform a given task, a cognitivist cognitive architecture needs to be provided with the knowledge to perform the task (or it needs to acquire this knowledge for itself). This combination of a given cognitive architecture and a particular knowledge set is referred to as a *cognitive model* and it is this knowledge which populates the cognitive architecture that provides the means to perform a task or to behave in some particular way. To put it another way, cognitive behaviour equals architecture combined with content.

An architecture is a theory about what is common to the content it processes and Soar is a particular theory of what cognitive behaviours have in common. In particular, the Soar theory holds that cognitive behaviour has at least the following characteristics: it is goal-oriented, it reflects a complex environment, it requires a large amount of knowledge, and it requires the use of symbols and abstraction. The idea of abstraction is very important in cognition. It comes down to the difference between the concept of something *vs.* something in particular. For example, a shirt as

a garment to provide warmth and protection *vs.* this particular blue shirt with a button-down collar and an embroidered logo on the pocket. The knowledge you have of a shirt as an abstract concept can be elicited — recalled and used — by something other than your particular perceptions in all their detail. This is referred to as a symbol (or set of symbols) and the knowledge is referred to as symbolic knowledge. The Soar cognitive architecture focusses on processing symbolic knowledge and matching it with knowledge that relate to current perceptions and actions. Let's now sketch out how it does this.

First, Soar is a production system (sometimes called a rule-based system). A production is effectively a condition-action pair and a production system is a set of production rules and a computational engine for interpreting or executing productions. Rules in Soar are called associations. Thus, the core of Soar comprises two memories, one called the long-term memory (sometimes referred to as recognition memory) which holds the productions rules, and one called working memory (also called declarative memory), which holds the attribute values that reflect Soar's perceptions and actions). In addition, there are several processes: one called *elaboration* which matches the productions and the attribute values (i.e it decides which productions can fire), one for determining the preferences to use in the decision process, and one called *chunking* which effectively learns new production rules (called *chunks*).

Soar operates in a cyclic manner with two distinct phases: a production cycle and a decision cycle. First, all productions that match the contents of declarative (working) memory fire. A production that fires may alter the state of declarative memory and cause other productions to fire. This continues until no more productions fire. At this point, the decision cycle begins and a single action is selected from several possible actions. The selection is based on stored action preferences.

Since there is no guarantee that the action preferences will be unambiguous or that they will lead to a unique action or indeed any action, the decision cycle may lead to what is known as an *impasse*. If this happens, Soar sets up a new state in a new problem space — a sub-goal — with the goal of resolving the im-

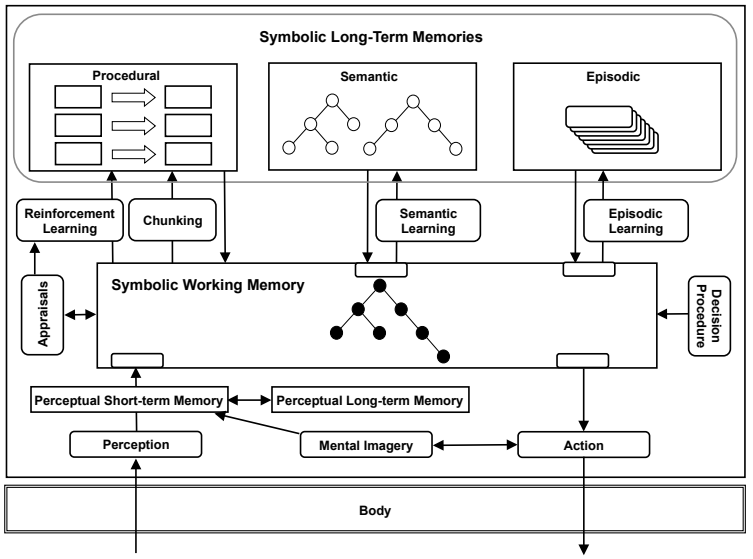

Figure 3.1: The Soar cognitive architecture, v. 9. From [114], © 2012, with permission from MIT Press.

passe. This process is known as *universal sub-goaling*. Resolving one impasse may cause others and the sub-goaling process continues. Eventually, all impasses should be resolved. In the case where the situation degenerates with Soar having insufficient knowledge to resolve the impasse, it chooses randomly between possible actions.

Whenever an impasse is resolved, Soar creates a new production rule, i.e. a new association, which summarizes the processing that occurred in the sub-state in solving the sub-goal. As we noted above, this new learned association is called a *chunk* and the Soar learning process is referred to as *chunking*.

As we said at the outset, the Soar cognitive architecture continues to evolve. While the foregoing description of Soar focussed on the production system that is so characteristic of cognitivist cognitive architectures, Soar also has the potential to be used for cognitive robotics. To facilitate this, the Soar architecture has been extended (see Figure 3.1) to embrace many of the components of emergent and hybrid cognitive architectures such as episodic memory, procedural memory, semantic memory, and associated learning techniques, e.g. reinforcement learning, as

well as the crucial capability for internal simulation of perception and action using mental imagery. We discuss these topics in more detail in Chapter 7.

3.4.2 *Darwin: Neuromimetic Robotic Brain-Based Devices*

Darwin[27] is a series of robot platforms designed to experiment with developmental agents. These agents are *brain-based devices* (BBDs) which exploit a simulated nervous system that can develop spatial and episodic memory as well as recognition capabilities through autonomous experiential learning, i.e. by exploring and interacting with the world around them. BBDs are neuromimetic — they mimic the neural structure of the brain — and are closely aligned with enactive and connectionist models. However, they differ from many connectionist approaches in that they focus on the nervous system as a whole, its constituent parts, and their interaction, rather than on a neural implementation of some individual memory, control, or recognition function.

[27] For more details on the Darwin cognitive architecture, please refer to [123, 124, 125, 145, 146, 147].

The principal neural mechanisms of a BBD are synaptic plasticity, a reward (or value) system, reentrant connectivity, dynamic synchronization of neuronal activity, and neuronal units with spatiotemporal response properties. Adaptive behaviour is achieved by the interaction of these neural mechanisms with sensorimotor correlations[28] which have been learned autonomously through active sensing and self-motion.

[28] Sensorimotor correlations are sometimes referred to as *contingencies*.

Different versions of Darwin exhibit different cognitive capabilities. For example, Darwin VIII is capable of discriminating reasonably simple visual targets (coloured geometric shapes) by associating them with an innately preferred auditory cue. Its simulated nervous system contains 28 neural areas, approximately 54,000 neuronal units, and approximately 1.7 million synaptic connections. The architecture comprises regions for vision (V1, V2, V4, IT), tracking (C), value or saliency (S), and audition (A). Gabor filtered images, with vertical, horizontal, and diagonal selectivity, and red-green colour filters with on-centre off-surround and off-centre on-surround receptive fields, are fed to V1. Sub-regions of V1 project topographically to V2 which in turn projects to V4. Both V2 and V4 have excitatory and in-

hibitory reentrant connections. V4 also has a non-topographical projection back to V2 as well as a non-topographical projection to IT, which itself has reentrant adaptive connections. IT also projects non-topographically back to V4. The tracking area (C) determines the gaze direction of Darwin VIII's camera based on excitatory projections from the auditory region A. This causes Darwin to orient toward a sound source. V4 also projects topographically to C causing Darwin VIII to centre its gaze on a visual object. Both IT and the value system S have adaptive connections to C which facilitates the learned target selection. Adaptation is effected using the Hebbian-like learning.[29] From a behavioural perspective, Darwin VIII is conditioned to prefer one target over others by associating it with the innately preferred auditory cue and to demonstrate this preference by orienting towards the target.

[29] Specifically, the Hebbian-like learning uses the Bienenstock-Cooper-Munroe (BCM) rule [148]; also see Chapter 2, Section 2.2.1.

Darwin IX can navigate and categorize textures using artificial whiskers based on a simulated neuroanatomy of the rat somatosensory system, comprising 17 areas, 1101 neuronal units, and approximately 8400 synaptic connections.

Darwin X is capable of developing spatial and episodic memory based on a model of the hippocampus and surrounding regions. Its simulated nervous system contains 50 neural areas, 90,000 neural units, and 1.4 million synaptic connections. It includes a visual system, head direction system, hippocampal formation, basal forebrain, a value/reward system based on dopaminegic function, and an action selection system. Vision is used to recognize objects and then compute their position, while odometry is used to develop head direction sensitivity.

3.4.3 ISAC

ISAC[30] — Intelligent Soft Arm Control — is a hybrid cognitive architecture for an upper torso humanoid robot (also called ISAC). From a software engineering perspective, ISAC is constructed from an integrated collection of software agents and associated memories. Agents encapsulate all aspects of a component of the architecture, operate asynchonously (i.e. without a shared clock to keep the processing of all agents locked in step

[30] For a more detailed description of the ISAC cognitive architecture, please refer to "Implementation of Cognitive Control for a Humanoid Robot" by Kazuhiko Kawamura and colleagues [149].

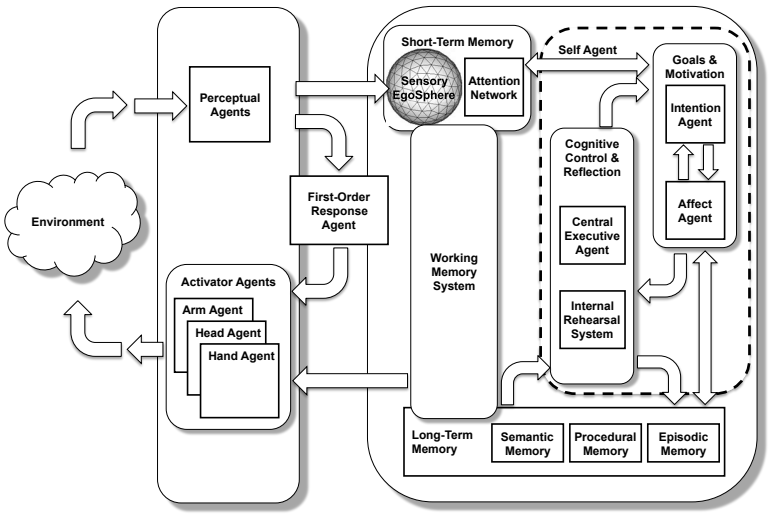

Figure 3.2: The ISAC cognitive architecture. From [149], © 2008, with permission from World Scientific Publishing Company.

with each another), and communicate with each other by passing messages.

As shown in Figure 3.2, the multi-agent ISAC cognitive architecture comprises activator agents for motion control, perceptual agents, and a First-order Response Agent (FRA) to effect reactive perception-action control. It has three memory systems: Short-term memory (STM), Long-term memory (LTM), and a working memory system (WMS).

STM has a robot-centred spatio-temporal memory of the perceptual events currently being experienced. This is called a Sensory EgoSphere (SES) and it is a discrete representation of what is happening around the robot, represented by a geodesic sphere indexed by two angles: horizontal (azimuth) and vertical (elevation). STM also has an attentional network that determines the perceptual events that are most relevant and then directs the robot's attention to them.

LTM stores information about the robot's learned skills and past experiences. LTM is made up of semantic, episodic, and procedural memory. Together, the semantic memory and episodic memory make up the robot's declarative memory of

the facts it knows. On the other hand, procedural memory stores representations of the motions the robot can perform.

ISAC's episodic memory abstracts past experiences and creates links or associations between them. It has multiple layers. At the bottom, an episodic experience contains information about the external situation (i.e. task-relevant percepts from the SES), goals, emotions (in this case, internal evaluation of the perceived situation), actions, and outcomes that arise from actions, and valuations of these outcomes (e.g. how close they are to the desired goal state and any reward received at a result). Episodes are connected by links that encapsulate behaviours: transitions from one episode to another. Higher layers abstract away specific details and create links based on the transitions at lower levels. This multi-layered approach allows for efficient matching and retrieval of memories.

WMS, inspired by neuroscience models of brain function, temporarily stores information that is related to the task currently being executed. It forms a type of cache memory for STM and the information it stores, called chunks, encapsulates expectations of future reward that are learned using a neural network.

Cognitive behaviour is the responsibility of a Central Executive Agent (CEA) and an Internal Rehearsal System, a system that simulates the effects of possible actions. Together with a Goals & Motivation sub-system comprising an Intention Agent and an Affect Agent, the CEA and Internal Rehearsal System form a compound agent called the Self Agent that, along with the FRA, makes decisions and acts according to the current situation and ISAC's internal states. The CEA is responsible for cognitive control, invoking the skills required to perform some given task on the basis of the current focus of attention and past experiences. The goals are provided by the Intention Agent. Decision-making is modulated by the Affect Agent.

ISAC works the following way. Normally, the First-order Response Agent (FRA) produces reactive responses to sensory triggers. However, it is also responsible for executing tasks. When a task is assigned by a human, the FRA retrieves the skill from procedural memory in LTM that corresponds to the skill described in the task information. It then places it in the

WMS as chunks along with the current percept. The Activator Agent then executes it, suspending execution whenever a reactive response is required. If the FRA finds no matching skill for the task, the Central Executive Agent takes over, recalling from episodic memory past experiences and behaviours that contain information similar to the current task. One behaviour-percept pair is selected, based on the current percept in the SES, its relevance, and the likelihood of successful execution as determined by internal simulation in the IRS. This is then placed in working memory and the Activator Agent executes the action.

As with Soar and Darwin, there are many features in the ISAC architecture that we will discuss in greater depth later in the book, such as attention (Chapter 5, Section 5.6), the role of affect and motivation in cognition (Chapter 6, Section 6.1.1), episodic, semantic, procedural, declarative, long-term, short-term, and working memory (Chapter 7, Section 7.2), and internal simulation (Chapter 7, Section 7.4).

3.5 Cognitive Architectures — What Next?

In this chapter, we began to put some flesh on the bones of the theoretical issues set out in Chapter 2 by addressing the blueprint of every cognitive system: its architecture. This took us on a long journey, from our discussion of what a cognitive architecture means for the cognitivisit, emergent, and hybrid paradigms of cognitive science, through quite a long list of the attributes that an ideal cognitive architecture should exhibit, to short summaries of three typical cognitive architectures, one cognitivist, one emergent, and one hybrid. On the way, we've encountered many new ideas and concepts which we had to gloss over far too quickly. Our goal now is to deepen our understanding of some of these issues: autonomy, embodiment, development, learning, memory, prospection, knowledge, and representation, for example. We turn our attention first to autonomy, a concept that is difficult to model and even harder to synthesize in artificial systems.