The Importance of Cognitive Architectures: An Analysis Based on CLARION

Ron Sun
Cognitive Science Department
Rensselaer Polytechnic Institute
Phone: (518) 276 3409
Fax: (518) 276 8268
Email: rsun@rpi.edu
URL: http://www.cogsci.rpi.edu/∼rsun

September 15, 2006

Abstract

Research in computational cognitive modeling investigates the nature of cognition through developing process-based understanding by specifying computational models of mechanisms (including representations) and processes. In this enterprise, a cognitive architecture is a domain-generic computational cognitive model that may be used for a broad, multiple-level, multiple-domain analysis of behavior. It embodies generic descriptions of cognition in computer algorithms and programs. Developing cognitive architectures is a difficult but important task. In this article, discussions of issues and challenges in developing cognitive architectures will be undertaken, and an example cognitive architecture (CLARION) will be described. [1]

KEYWORDS: cognition, modeling, cognitive architecture

1 Introduction

For a long time, my own research has been focused on so called hybrid systems, in particular, systems that integrate the then newly emerging connectionist models and the more traditional symbolic processing models. Such systems arguably combine the strength of connectionist models and symbolic models, thus possessing a wider range of capabilities (Sun and Bookman 1994, Sun 1994). They have been used both in cognitive modeling—in understanding human cognition

[1]This article is based on a keynote address at NeSy'05, a keynote address at PRIMA'05, and a plenary talk at KES'05, by the author during the summer and the fall of 2005.

through developing computational models of cognitive processes, and in building intelligent systems for practical applications.

The general idea behind this body of work on hybrid systems, developing more comprehensive models through integrating a variety of techniques, can be further extended into so called cognitive architectures, that is, cognitive models that are domain-generic and encompass a wide range of cognitive capabilities.

Another perspective on this is that of a progression from engineering to science. While most of the work on hybrid systems (mostly within the field of artificial intelligence) takes an engineering approach, the research on cognitive architectures (mostly within the field of cognitive science) takes a scientific approach—focusing on gathering empirical data and developing models that serve as scientific theories and scientific explanations of the data through an iterative hypothesis-test cycle. The function of a cognitive architecture is to provide an essential framework to facilitate more detailed modeling and exploration of various components and processes of the mind.

Developing cognitive architectures is a difficult challenge. In this article, the importance of, and issues and challenges in, developing cognitive architectures will be discussed, examples of cognitive architectures will be given, and future directions will be outlined (Sun 2007b). In the next section, the question of what a cognitive architecture is is answered. In section 3, the importance of cognitive architectures is addressed. In section 4, to further clarify the importance of cognitive architectures, a multi-level framework for cognitive modeling is outlined. In section 5, some background regarding the development of cognitive architectures is provided. Then, in section 6, an example cognitive architecture is presented in detail and it applications to cognitive modeling, artificial intelligence, and social simulation described. In section 7, the significant challenges related to developing cognitive architectures are articulated. Finally, section 8 concludes this article.

2 What is a Cognitive Architecture?

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 a broad, multiple-level, multiple-domain analysis of behavior (Newell 1990, Sun 2002).

Let us explore this notion of architecture with an analogy. The architecture for a building consists of its overall framework and its overall design, as well as roofs, foundations, walls, windows, floors, and so on. Furniture and appliances can be easily rearranged and/or replaced and therefore they are not part of the architecture. By the same token, a cognitive architecture includes overall structures, essential divisions of modules, essential relations between modules, basic representations and algorithms within modules, and a variety of other aspects (Sun 2004). In general, an architecture includes those aspects of a system that are relatively invariant across time, domains, and individuals. It deals with componential processes of cognition in a structurally and mechanistically

well defined way.

In relation to understanding the human mind (i.e., cognitive science), a cognitive architecture provides a concrete framework for more detailed modeling of cognitive phenomena, through specifying essential structures, divisions of modules, relations between modules, and so on. Its function is to provide an essential framework to facilitate more detailed modeling and exploration of various components and processes of the mind. Research in computational cognitive modeling explores the essence of cognition and various cognitive functionalities through developing detailed, process-based understanding by specifying computational models of mechanisms and processes. It embodies descriptions of cognition in computer algorithms and programs. That is, it produces runnable computational models. Detailed simulations are then conducted based on the computational models. In this enterprise, a cognitive architecture may be used for a broad, multiple-level, multiple-domain analysis of cognition.

In relation to building intelligent systems, a cognitive architecture specifies the underlying infrastructure for intelligent systems, which includes a variety of capabilities, modules, and subsystems. On that basis, application systems can be more easily developed. A cognitive architecture carries also with it theories of cognition and understanding of intelligence gained from studying the human mind. Therefore, the development of intelligent systems can be more cognitively grounded, which may be advantageous in many circumstances.

3 Why are Cognitive Architectures Important?

While there are all kinds of cognitive architectures in existence, in this article I am concerned specifically with psychologically oriented cognitive architectures (as opposed to software engineering oriented "cognitive" architectures): their importance and their applications. Psychologically oriented cognitive architectures are particularly important (compared with software engineering oriented "cognitive" architectures) because (1) they shed new light on human cognition and therefore they are useful tools for advancing the understanding of cognition, (2) they may (in part) serve as a foundation for understanding collective human behavior and social phenomena (to be detailed later), (3) furthermore, they are "intelligent" systems that are cognitively realistic (relatively speaking) and therefore they are more human-like in many ways, Let us examine the importance of this type of cognitive architecture.

For cognitive science, the importance of such cognitive architectures lie in the fact that they are beneficial to understanding the human mind. In understanding cognitive phenomena, the use of computational simulation on the basis of cognitive architectures forces one to think in terms of process, and in terms of detail. Instead of using vague, purely conceptual theories, cognitive architectures force theoreticians to think clearly. They are therefore critical tools in the study of the mind. Researchers who use cognitive architectures must specify a cognitive mechanism in sufficient detail to allow the resulting models to be implemented on computers and run as simulations. This

approach requires that important elements of the models be spelled out explicitly, thus aiding in developing better, conceptually clearer theories. It is certainly true that more specialized, narrowly scoped models may also serve this purpose, but they are not as generic and as comprehensive and thus they are not as useful (more detailed discussions later).

An architecture serves as an initial set of assumptions to be used for further modeling of cognition. These assumptions, in reality, may be based on either available scientific data (for example, psychological or biological data), philosophical thoughts and arguments, or ad hoc working hypotheses (including computationally inspired such hypotheses). An architecture is useful and important precisely because it provides a comprehensive initial framework for further modeling in a variety of task domains.

Cognitive architectures also provide a deeper level of explanation. Instead of a model specifically designed for a specific task (often in an ad hoc way), using a cognitive architecture forces modelers to think in terms of the mechanisms and processes available within a generic cognitive architecture that are not specifically designed for a particular task, and thereby to generate explanations of the task that is not centered on superficial, high-level features of a task (as often happens with specialized, narrowly scoped models), that is, to generate explanations of a deeper kind. To describe a task in terms of available mechanisms and processes of a cognitive architecture is to generate explanations centered on primitives of cognition as envisioned in the cognitive architecture, and therefore such explanations are deeper explanations. Because of the nature of such deeper explanations, this style of theorizing is also more likely to lead to unified explanations for a large variety of data and/or phenomena, because potentially a large variety of tasks, data, and phenomena can be explained on the basis of the same set of primitives provided by the same cognitive architecture. Therefore, using cognitive architectures leads to comprehensive theories of the mind (Newell 1990, Anderson and Lebiere 1998, Sun 2002), unlike using more specialized, narrowly scoped models.

While the importance of being able to reproduce the nuances of empirical data from specific psychological experiments is evident, broad functionality in cognitive architectures is also important (Newell 1990), as the human mind needs to deal with the full cycle that includes all of the followings: transducing signals, processing them, storing them, representing them, manipulating them, and generating motor actions based on them. There is clearly a need to develop generic models of cognition that are capable of a wide range of functionalities, to avoid the myopia often resulting from narrowly-scoped research (in psychology in particular).

In all, cognitive architectures are believed to be essential in advancing the understanding of the mind (Anderson 1983, Newell 1990, Sun 2002). Therefore, developing cognitive architectures is an important enterprise in cognitive science.

On the other hand, for the fields of artificial intelligence and computational intelligence (AI/CI), the importance of cognitive architectures lies in the fact that they support the central goal of AI/CI—Building artificial systems that are as capable as human beings. Cognitive architectures help us to reverse engineer the only truly intelligent system around—the human mind. They

level	object of analysis	type of analysis	model
1	inter-agent/collective processes	social/cultural	collections of agent models
2	agents	psychological	individual agent models
3	intra-agent processes	componential	modular construction of agent models
4	substrates	physiological	biological realization of modules

Figure 1: A hierarchy of four levels.

constitute a solid basis for building truly intelligent systems, because they are well motivated by, and properly grounded in, existing cognitive research. The use of cognitive architectures in building intelligent systems may also facilitate the interaction between humans and artificially intelligent systems because of the similarity between humans and cognitively based intelligent systems.

It is also worth noting that cognitive architectures are the antithesis of "expert systems": Instead of focusing on capturing performance in narrow domains, they are aimed to provide broad coverage of a wide variety of domains (Langley and Laird 2003). Business and industrial applications of intelligent systems increasingly require broadly scoped systems that are capable of a wide range of intelligent behaviors, not just isolated systems of narrow functionalities. For example, one application may require the inclusion of capabilities for raw image processing, pattern recognition, categorization, reasoning, decision making, and natural language communications. It may even require planning, control of robotic devices, and interactions with other systems and devices. Such requirements accentuate the importance of research on broadly scoped cognitive architectures that perform a wide range of cognitive functionalities across a variety of task domains (as opposed to more specialized systems).

4 Multiple Levels of Explanations

A broad perspective on the social and behavioral sciences can lead to a view of multiple "levels" of analysis encompassing multiple disciplines. That is, a set of related disciplines may be readily cast as a set of different levels of analysis, from the most macroscopic to the most microscopic. These different *levels* include: the sociological level, the psychological level, the componential level, and the physiological level. In other words, as has been argued in Sun et al (2005), one may view different disciplines as different levels of abstraction in the process of exploring essentially the same broad set of questions (cf. Newell 1990). See Figure 1.

First of all, there is the sociological level, which includes collective behaviors of agents (Durkheim 1895), inter-agent processes (Vygotsky 1986), sociocultural processes, social structures and organizations, as well as interactions between agents and their (physical and sociocultural) environments. Although studied extensively by sociology, anthropology, political science, and economics, this level has tended to be downplayed in cognitive science. Relatively recently, cognitive science, as a whole, has come to grip with the fact that cognition is, at least in part, a sociocultural process (Lave 1988,

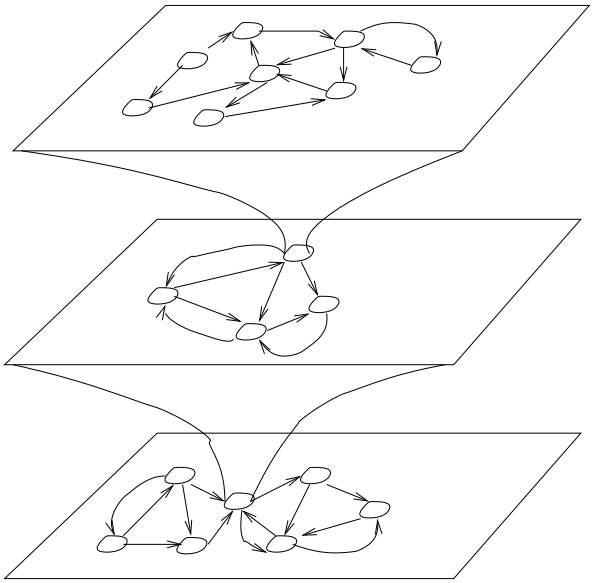


Figure 2: The cascading levels of analysis, with cross-level mappings.

Hutchins 1995, Zerubavel 1997, Sun 2001b), but much more work is needed along that line. [2]

The next level is the psychological level, which covers individual experiences, individual behaviors, individual performance, as well as beliefs, concepts, and skills employed by individual agents. In relation to the sociological level, the relationship of individual beliefs, concepts, and skills with those of the society and the culture, and the processes of change of these beliefs, concepts, and skills, independent of or in relation to those of the society and the culture, may be investigated (in inter-related and mutually influential ways). At this level, one may examine human behavioral data, and compare them with models (which may be based on cognitive architectures) and with insights from the sociological level and details from the lower levels.

The third level is the componential level. At this level, one studies and models cognitive agents in terms of components (e.g., in the form of a cognitive architecture), with the theoretical language of a particular paradigm (for example, symbolic computation or connectionist networks, or their combinations thereof). At this level, one may specify computationally an overall architecture consisting of multiple components. One may also specify the essential computational processes of each component as well as the essential connections among components. That is, one imputes a computational process onto a cognitive function. Ideas and data from the psychological level (that is, the psychological constraints from above), which bear significantly on the division of components and their possible implementations, are among the most important considerations. This level may also incorporate biological/physiological facts regarding plausible divisions and their implementations (that is, it can incorporate ideas from the next level down—the physiological level, which offers the biological constraints). This level results in cognitive *mechanisms* (though they

[2]See Sun (2001b) for a more detailed argument for the relevance of sociocultural processes to cognition and vice versa.

are computational and thus somewhat abstract compared with physiological-level details). [3]

Although this level concerns intra-agent processes, computational cognitive models (cognitive architectures) developed therein may be used to capture processes at higher levels, including interaction at a sociological level whereby multiple individuals are involved. This can be accomplished, for example, by examining interactions of multiple copies of individual agent models (based on the same cognitive architecture) or those of different individual agent models (based on different cognitive architectures). One may use computation as a means for constructing cognitive architectures at a sub-agent level (the componential level), but one may go up from there to the psychological level and to the sociological level (see the discussion regarding mixing levels in Sun et al 2005).

The lowest level of analysis is the physiological level, that is, the biological substrate (or the biological implementation) of computation (Dayan 2003). This level is the focus of a range of disciplines including biology, physiology, cognitive neuroscience, computational neuroscience, and so on. Although biological substrates are not our main concern here, they may nevertheless provide useful input as to what kind of computation is likely employed and what a plausible architecture (at a higher level) should be like (Piaget 1971). The main utility of this level is to facilitate analysis at higher levels, that is, analysis using low-level information to narrow down choices in selecting computational architectures as well as choices in implementing componential computation. [4]

To illustrate this view of cascading levels, Figure 2 shows the correspondences (mappings) among levels, with a cascade of maps of various resolutions. [5] In this enterprise of multiple levels in the social and behavioral sciences, a cognitive architecture may serve as a centerpiece, tying together various strands of research, and thus serving a role that is significantly different from specialized, narrowly scoped cognitive models. It may serve this purpose due to the comprehensiveness of its functionality and the depth with which it has been developed (at least for some psychologically oriented/grounded cognitive architectures). Thus, detailed mechanisms are developed within a cognitive architecture, which may be tied to low-level cognitive processes, while at the same time a cognitive architecture as a whole may function at a very high level of cognitive and social processes.

[3]The importance of this level has been argued for, for example, in Newell (1990), Anderson and Lebiere (1998), and Sun et al (2004).

[4]Work at this level is basically the reverse-engineering of biological systems. In such a case, what needs to be done is to pinpoint the basic primitives that are of relevance to the higher-level functioning that is of interest. (While many low-level details are highly significant, clearly not all low-level details are significant or even relevant.) After identifying proper primitives, one may study processes that involve those primitives, in mechanistic/computational terms.

[5]There can, of course, be more detailed and finer divisions of levels in many different ways (Sun 2007).

5 Some Background

5.1 A Brief History

Newell (1980, 1990) proposed that cognitive theories should be developed that satisfy multiple criteria, in order to avoid theoretical myopia. He provided two overlapping lists of 13 criteria that a cognitive architecture should satisfy. There have been steady developments of cognitive architectures for the past three decades since Newell's proposal.

Early cognitive architectures often took the form of production systems and were (more or less) concerned with psychological phenomena (e.g., Klahr et al 1989). However, other forms of cognitive architectures have also been developed over the years. They may be in the form of a connectionist model, a constraint satisfaction network, a loosely organized library of models, a hybrid system of different models, and so on. Some of them may be more concerned with applications to building artificial systems than explaining psychological phenomena (see, e.g., Langley and Laird 2003).

Soar, a purely symbolic production architecture, has been continuously developed over the past thirty years, mostly for the purpose of building application systems (Rosenbloom et al 1993, Newell 1990). In Soar, based on the framework of a state space and operators for searching the state space, decisions are made by different productions proposing different operators, when there is a goal on a goal stack. When a sequence of productions leads to achieving a goal, chunking occurs, which creates a single production that summarizes the process (using so-called explanation-based learning). A large amount of initial (a priori) knowledge about states and operators is required for Soar to work.

A series of similar architectures were proposed by Anderson (1983) and Anderson and Lebiere (1998): in particular, ACT* and ACT-R. ACT* is made up of a semantic network (for declarative knowledge) and a production system (for procedural knowledge). Productions are formed through "proceduralization" of declarative knowledge, modified through use by generalization and discrimination (i.e., specialization), and have strengths associated with them which are used for firing. ACT-R is a descendant of ACT*, in which procedural learning is limited to production formation through mimicking and production firing is based on log odds of success. There are also other numerical measures in ACT-R. So ACT-R in fact uses a combined form of symbolic and numerical representations.

CLARION is an integrative cognitive architecture (Sun 2002). Essentially, it consists of two levels, whereby the top level is conceptual (with explicit conceptual representations) and the bottom level subconceptual (consisting of implicit reactive routines) (Sun 1995, 2001). Symbolic and distributed representation are used in the two levels respectively. It is therefore a hybrid connectionist-symbolic system. CLARION is also made up of a number of subsystems, incorporating all of the following: action decision making, memory retrieval and inference, motivational processes, and meta-cognition. These subsystems interact with each other on a continuous basis.

CLARION therefore provides a broad framework. The notion of a central (general-purpose) symbolic processor is rejected in this architecture. The traditional reliance on simple rules (which limits the capabilities of traditional cognitive architectures) is also rejected in CLARION.

There have also been a variety of other computational cognitive architectures in the literature. See Sun (2004) for a brief review.

5.2 Capabilities and Constraints

A cognitive architecture is designed to support a range of capabilities. Together, these capabilities constitute the functionality of a cognitive architecture. Some of the needed functionalities include: perception, categorization, representation (in a broad sense), memory of all sorts, decision making, reasoning, planning (in a broad sense), problem solving, meta-cognition, communication, action control and execution, and learning of all sorts (which may be embedded in all of the above categories of functionalities). There may also be a need for motivational processes, goal representation, and meta-cognitive processes. Most cognitive architectures, including those mentioned above, do not yet support all of these functionalities fully.

An important question with regard to any capability in an architecture is whether the architecture includes that capability as an integral part of the architecture or whether the architecture includes sufficient functionalities that allow the capability to be implemented later on (for a particular task domain or more generically). This determines what we view as an integral part of an architecture and what we view as secondary or derived capability. Sun (2004) provides a discussion of the relation between an architecture and the innate structures in the human mind and the notion of minimality in an architecture. These ideas help us to sort out what should/need to be included in an architecture.

Another issue is accounting for dynamic nature of cognitive processes. Perception, categorization, representation, memory, decision making, reasoning, planning, problem solving, meta-cognition, communication, action control and execution, learning, and motivational processes all interact with each other. Their patterns of interaction change with changing task demands, experiences, sociocultural environments, and so on. Some researchers stress the fact that cognition represents a context sensitive, statistical structure that on the surface changes constantly—a statistical structure in perpetual motion. However, even dynamic systems can be attributed to its constituting elements (otherwise the field of dynamic systems does not need to exist). Thus, we need to strive for complex cognitive architectures that capture dynamics of cognition through capturing its constituting elements.

I should note that in relation to the issue of functionality, there is always a danger of overgenerality in computational cognitive modeling and in cognitive architectures. That is, a model may be under-constrained to the extent that it will match any data (Regier 2003). To address this problem, multiple simulations in multiple domains are needed to narrow down choices and to

constrain parameter spaces (more discussions later).

5.3 AI and Cognitive Science

Artificial intelligence (or computational intelligence) and cognitive science have always been overlapping. Early in their history, this overlap was rather significant. Herbert Simon (1957) once declared that "AI can have two purposes. One is to use the power of computers to augment human thinking. The other is to use a computer's artificial intelligence to understand how humans think." Conversely, ever since the time when cognitive science was first created (that is, the time when cognitive science society was formed), artificial intelligence has been identified as one of the constituting disciplines. However, over time, the two disciplines have grown apart. The difficulty of many computational problems tackled by AI has led it to often adopt brute-force, optimality-oriented, or domain-specific solutions that are not cognitively realistic. Conversely, the need for experimental controllability, detail-orientedness, and precision has led cognitive scientists to focus on some problems and experimental paradigms that AI would not consider interesting.

One important question is whether cognitive science is relevant to solving AI/CI problems at all. Some cognitive scientists believe so. The human mind is one of the most flexible, general, and powerful intelligent systems in existence. Therefore, a good way to solving most AI/CI problems seems to be adopting a cognitive approach.

Consequently, another important question is whether sub-optimal, "satisficing" methods or algorithms that are often discovered by cognitive science research are useful to AI/CI. Many cognitive scientists would say yes. It may be argued that in AI/CI, too much focus has been devoted to the search for domain-specific, brute-force, and/or optimal solutions (Sun 2001). Both robustness over a broad range of problems and computational efficiency (and tractability) are essential to long-term success of AI/CI as a field that generates general theories and general scientific frameworks. Real-world problems are complex and they often include many different aspects, which require generality and broad functionalities in order to be solved in a robust manner. All of these requirements above point to cognitively based approaches toward developing computational models, as human cognition is thus far the best example of intelligent systems that are robust and efficient.

In the reverse direction, can AI/CI contribute to our understanding of human cognition? The answer is yes for many researchers. This is because AI/CI addresses many problems that are central to human cognition, usually in a mathematically/logically motivated or optimal way. AI/CI solutions often reflect fundamental mathematical/logical constraints and regularities, which should be relevant and applicable to all approaches to those problems, including those adopted by human cognition. Therefore, in that sense, they may shed some light on possible details of cognitive processes and mechanisms, and may in some cases lead to better understanding of cognition in general.

As touched upon before, we need to make the distinction between cognitive architectures that

are psychologically oriented versus those that are AI oriented. This is an important distinction, because these two types of cognitive architectures have become quite different, in terms of their focuses, basic assumptions, and degrees of scientific rigor. In this article, I am mostly concerned with models that are grounded in both high-level cognitive theories and psychological data (as discussed earlier). Thus, we are not bringing to this discussion what might be referred to as artificial intelligence theories and models, or software engineering models and tools. For example, we are certainly not treating Deep Blue as a theory of human chess. Instead, we are more interested in putting together a general theory of human cognition that is capable of playing chess (as well as many other activities).

6 An Example of a Cognitive Architecture

6.1 An Overview

Below I will describe a cognitive architecture CLARION. It has been described extensively in a series of previous papers, including Sun and Peterson (1998), Sun et al (2001), and Sun (2002, 2003).

CLARION is an integrative architecture, consisting of a number of distinct subsystems, with a dual representational structure in each subsystem (implicit versus explicit representations). Its subsystems include the action-centered subsystem (the ACS), the non-action-centered subsystem (the NACS), the motivational subsystem (the MS), and the meta-cognitive subsystem (the MCS). The role of the action-centered subsystem is to control actions, regardless of whether the actions are for external physical movements or for internal mental operations. The role of the non-action-centered subsystem is to maintain general knowledge, either implicit or explicit. The role of the motivational subsystem is to provide underlying motivations for perception, action, and cognition, in terms of providing impetus and feedback (e.g., indicating whether outcomes are satisfactory or not). The role of the meta-cognitive subsystem is to monitor, direct, and modify the operations of the action-centered subsystem dynamically as well as the operations of all the other subsystems.

Each of these interacting subsystems consists of two levels of representation (i.e., a dual representational structure): Generally, in each subsystem, the top level encodes explicit knowledge and the bottom level encodes implicit knowledge. The distinction of implicit and explicit knowledge has been amply argued for before (see Reber 1989, Stanley et al 1989, Seger 1994, Cleeremans et al 1998, Sun 2002). The two levels interact, for example, by cooperating in actions, through a combination of the action recommendations from the two levels respectively, as well as by cooperating in learning through a bottom-up and a top-down process (to be discussed below). Essentially, it is a dual-process theory of mind (Chaiken and Trope 1999). See Figure 3.

It has been intended that this cognitive architecture satisfy some basic requirements as follows. It should be able to learn with or without a priori domain-specific knowledge to begin with (Reber

1989, and Sun et al 2001). It also has to learn continuously from on-going experience in the world. As indicated by Medin et al. (1987), Nosofsky et al (1994), and others, human learning is often gradual and on-going. As suggested by Reber (1989), Seger (1994), Anderson (1983), and others, there are clearly different types of knowledge involved in human learning (e.g., procedural vs. declarative, implicit vs. explicit, or subconceptual vs. conceptual). Moreover, different types of learning processes are involved in acquiring different types of knowledge, capturing statistically significant features of the environments (Karmiloff-Smith 1986, Sun et al 2001). It should incorporate both the situated (reactive) view and the cognitivist view (Sun 2002). It should be able to handle complex situations that are not amenable to simple rules. Finally, unlike ACT-R or SOAR, it should more fully incorporate motivational processes as well as meta-cognitive processes. Based on the above considerations, CLARION was developed. [6]

6.2 Some Details

6.2.1 The Action-Centered Subsystem

First, let us focus on the action-centered subsystem (the ACS) of CLARION. The overall operation of the action-centered subsystem may be described as follows:

1. Observe the current state x.
2. Compute in the bottom level the Q-values of x associated with each of all the possible actions a_i's: $Q(x, a_1)$, $Q(x, a_2)$, ..., $Q(x, a_n)$.
3. Find out all the possible actions (b_1, b_2, ..., b_m) at the top level, based on the input x (sent up from the bottom level) and the rules in place.
4. Compare or combine the values of the selected a_i's with those of b_j's (sent down from the top level), and choose an appropriate action b.
5. Perform the action b, and observe the next state y and (possibly) the reinforcement r.
6. Update Q-values at the bottom level in accordance with the *Q-Learning-Backpropagation* algorithm
7. Update the rule network at the top level using the *Rule-Extraction-Refinement* algorithm.
8. Go back to Step 1.

In the bottom level of the action-centered subsystem, implicit reactive routines are learned (incorporating the situated cognition view): A Q-value is an evaluation of the "quality" of an action in a given state: $Q(x, a)$ indicates how desirable action a is in state x (which consists of some sensory input). The agent may choose an action in any state based on Q-values. To acquire the Q-values, the *Q-learning* algorithm (Watkins 1989) may be used, which is a reinforcement learning algorithm. It basically compares the values of successive actions and adjusts an evaluation function on that basis. It thereby develops reactive sequential behaviors or reactive routines (such

[6] CLARION has been implemented as a set of Java packages, available at: http://www.cogsci.rpi.edu/∼rsun/clarion.html

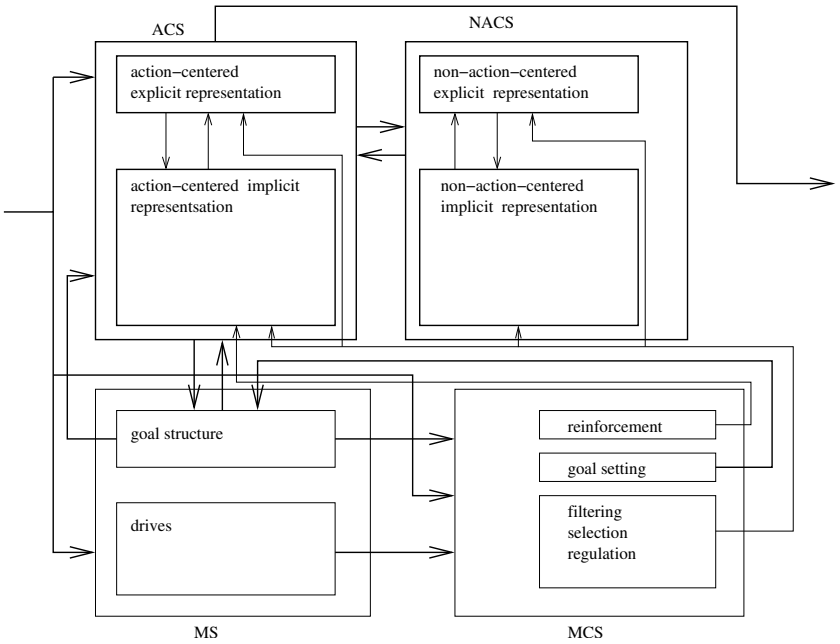


Figure 3: The CLARION Architecture. ACS stands for the action-centered subsystem, NACS the non-action-centered subsystem, MS the motivational subsystem, and MCS the meta-cognitive subsystem.

as navigating through a body of water or handling daily activities, etc., in a reactive way; Sun 2003).

The bottom level of the action-centered subsystem is modular; that is, a number of small neural networks co-exist each of which is adapted to a specific modality, task, or group of input stimuli. This coincides with the modularity claim (Fodor 1983, Karmiloff-Smith 1986, Cosmides and Tooby 1994, Hirschfield and Gelman 1994) that much processing is done by limited, encapsulated (to some extent), specialized processors that are highly efficient. These modules can be developed in interacting with the world (computationally, through various decomposition methods; e.g., Sun and Peterson 1999). Some of them, however, are formed evolutionarily, that is, given a priori to agents, reflecting their hardwired instincts and propensities (Hirschfield and Gelman 1994). Because of these networks, CLARION is able to handle very complex situations that are not amenable to simple rules (see examples later).

In the top level of the action-centered subsystem, explicit symbolic conceptual knowledge is captured in the form of explicit symbolic rules. See Sun (2003) for details. There are many ways in which explicit knowledge may be learned, including independent hypothesis-testing learning and "bottom-up learning" as discussed below.

Autonomous Generation of Explicit Conceptual Structures. Humans are generally able to learn implicit knowledge through trial and error, without necessarily utilizing a priori knowledge. On top of that, explicit knowledge can be acquired also from on-going experience in the world, possibly

through the mediation of implicit knowledge (i.e., bottom-up learning; see Sun 2002, Stanley et al 1989, Karmiloff-Smith 1986). The basic process of bottom-up learning is as follows: if an action implicitly decided by the bottom level is successful, then the agent extracts an explicit rule that corresponds to the action selected by the bottom level and adds the rule to the top level. Then, in subsequent interaction with the world, the agent verifies the extracted rule by considering the outcome of applying the rule: if the outcome is not successful, then the rule should be made more specific and exclusive of the current case; if the outcome is successful, the agent may try to generalize the rule to make it more universal (Michalski 1983). [7] After explicit rules have been learned, a variety of explicit reasoning methods may be used. Learning explicit conceptual representation at the top level can also be useful in enhancing learning of implicit reactive routines at the bottom level (e.g., Sun et al 2001).

Assimilation of Externally Given Conceptual Structures. Although CLARION can learn even when no a priori or externally provided knowledge is available, it can make use of it when such knowledge is available (Schneider and Oliver 1991, Anderson 1983). To deal with instructed learning, externally provided knowledge, in the forms of explicit conceptual structures such as rules, plans, categories, and so on, can (1) be combined with existent conceptual structures at the top level (i.e., internalization), and (2) be assimilated into implicit reactive routines at the bottom level (i.e., assimilation). This process is known as top-down learning. See Sun (2003) for more details.

6.2.2 The Non-Action-Centered Subsystem

The non-action-centered subsystem (NACS) may be used for representing general knowledge about the world (i.e., constituting the "semantic" memory as defined in, e.g., Quillian 1968), for performing various kinds of memory retrievals and inferences. Note that the non-action-centered subsystem is under the control of the action-centered subsystem (through its actions).

At the bottom level, "associative memory" networks encode non-action-centered implicit knowledge. Associations are formed by mapping an input to an output (such as mapping "2+3" to "5"). The regular backpropagation learning algorithm can be used to establish such associations between pairs of inputs and outputs (Rumelhart et al 1986).

On the other hand, at the top level of the non-action-centered subsystem, a general knowledge store encodes explicit non-action-centered knowledge (Sun 1994). In this network, chunks are specified through dimensional values (features). [8] A node is set up in the top level to represent a chunk. The chunk node connects to its corresponding features represented as individual nodes in the bottom level of the non-action-centered subsystem (see Sun 1994). Additionally, links between chunks encode explicit associations between pairs of chunks, known as associative rules. Explicit

[7] The detail of the bottom-up learning algorithm can be found in Sun and Peterson (1998).

[8] The basic form of a chunk is as follows: *chunk-id*$_i$: $(dim_{i_1}, val_{i_1})(dim_{i_2}, val_{i_1})$ ..... (dim_{i_n}, val_{i_n}), where *dim* denotes a particular state/output dimension, and *val* specifies its corresponding value. For example, *table-1: (size, large) (color, white) (number-of-legs, four)* specifies a large, four-legged, white table.

associative rules may be formed (i.e., learned) in a variety of ways (Sun 2003).

During reasoning, in addition to applying associative rules, similarity-based reasoning may be employed in the non-action-centered subsystem. During reasoning, a known (given or inferred) chunk may be automatically compared with another chunk. If the similarity between them is sufficiently high, then the latter chunk is inferred (see Sun 2003 for details; see also Sun 1994, 1995).

As in the action-centered subsystem, top-down or bottom-up learning may take place in the non-action-centered subsystem, either to extract explicit knowledge in the top level from the implicit knowledge in the bottom level or to assimilate explicit knowledge of the top level into implicit knowledge in the bottom level.

6.2.3 The Motivational and the Meta-Cognitive Subsystem

The motivational subsystem (the MS) is concerned with drives and their interactions (Toates 1986), which leads to actions. It is concerned with why an agent does what it does. Simply saying that an agent chooses actions to maximizes gains, rewards, reinforcements, or payoffs leaves open the question of what determines these things. The relevance of the motivational subsystem to the action-centered subsystem lies primarily in the fact that it provides the context in which the goal and the reinforcement of the action-centered subsystem are set. It thereby influences the working of the action-centered subsystem, and by extension, the working of the non-action-centered subsystem.

A bipartite system of motivational representation is in place in CLARION. The explicit goals (such as "finding food") of an agent (which is tied to the working of the action-centered subsystem) may be generated based on internal drive states (for example, "being hungry"). (See Sun 2003 for details.)

Beyond low-level drives (concerning physiological needs), [9] there are also higher-level drives. Some of them are primary, in the sense of being "hard-wired". For example, Maslow (1987) developed a set of these drives in the form of a "need hierarchy". [10] While primary drives are built-in and relatively unalterable, there are also "derived" drives, which are secondary, changeable, and acquired mostly in the process of satisfying primary drives. [11]

The meta-cognitive subsystem (the MCS) is closely tied to the motivational subsystem. The meta-cognitive subsystem monitors, controls, and regulates cognitive processes for the sake of improving cognitive performance (Nelson 1993, Smith et al 2003). Control and regulation may be in the forms of setting goals for the action-centered subsystem, setting essential parameters of

[9] Low-level drives include, for example, *need for food*, *need for water*, *need to avoid danger*, *need to avoid boredom*, and so on (Sun 2003).

[10] A few high-level drives include: *desire for domination*, *desire for social approval*, *desire for following social norms*, *desire for reciprocation*, *desire for imitation* (of certain other people), and so on (Sun 2003).

[11] Admittedly, the coverage of motivations here is rudimentary. However, CLARION illustrates how a few simple motivational constructs can contribute to interesting behavior.

the action-centered subsystem and the non-action-centered subsystem, interrupting and changing on-going processes in the action-centered subsystem and the non-action-centered subsystem, and so on. Control and regulation can also be carried out through setting reinforcement functions for the action-centered subsystem. All of the above can be done on the basis of drive states in the motivational subsystem. The meta-cognitive subsystem is also made up of two levels: th top level (explicit) and the bottom level (implicit).

6.2.4 Too Many Mechanisms?

Therefore, in CLARION, there are a variety of memories: general "semantic" memory in both implicit and explicit forms (in the non-action-centered subsystem, for general knowledge), episodic memory (also in the non-action-centered subsystem), procedural memory in both implicit and explicit forms (in the action-centered subsystem), working memory (in the action-centered subsystem), goal structures (in the action-centered subsystem), and so on. Separately, there are also a variety of subsystems, as discussed above.

Furthermore, CLARION does not represent a static, strictly reductive approach, but accentuates complex dynamic interactions of various processes in a context-sensitive way: perception, categorization, memory, decision making, reasoning, planning, problem solving, meta-cognition, communication, action control and execution, learning, motivational processes, and so on. Various neural and other learning algorithms in CLARION captures complex statistical structures in the world.

Are there too many specialized mechanisms? One argument against CLARION would be that in general, more constrained architectures provide deeper explanations, rather than those with a larger pool of specific mechanisms for specific classes of phenomena. However, this argument ignored the fact that there are severe limitations in other architectures in terms of the range of decision making, learning, reasoning, motivational, and meta-cognitive phenomena those other architectures can capture. If an architecture fails to capture as broad a range of cognitive phenomena, then there is very little to be gained in being "constrained", because no "deep" explanations come out of it when it cannot capture many of the phenomena.

In all, CLARION is grounded in psychological research, is reasonably compact, and matches a range of empirical data (Sun 2002). Let us next look into the variety of data and phenomena CLARION can capture.

6.3 Accounting for Cognitive Data

CLARION has been successful in accounting for and explaining a variety of psychological data. For example, a number of well known skill learning tasks have been simulated using CLARION that span the spectrum ranging from simple reactive skills to complex cognitive skills. The simulated

tasks include serial reaction time tasks, artificial grammar learning tasks, process control tasks, categorical inference tasks, alphabetical arithmetic tasks, and the Tower of Hanoi task (Sun 2002). Among them, serial reaction time and process control tasks are typical implicit learning tasks (mainly involving implicit reactive routines), while Tower of Hanoi and alphabetic arithmetic are high-level cognitive skill acquisition tasks (with a significant presence of explicit processes). In addition, we have done extensive work on a complex minefield navigation task, which involves complex sequential decision making (Sun et al 2001, Sun and Peterson 1998). We have also worked on an organizational decision task (Sun and Naveh 2004), and other social simulation tasks, as well as meta-cognitive tasks. While accounting for various psychological data, CLARION provides explanations that shed new light on cognitive phenomena. This point can be illustrated by the following two examples.

For instance, in Sun and Zhang (2003), we simulated the alphabetic arithmetic task of Rabinowitz and Goldberg (1995). In the task, subjects were asked to solve alphabetic arithmetic problems of the forms: $letter1 + number = letter2$ or $letter1 - number = letter2$, where $letter2$ is $number$ positions up or down from $letter1$ (depending on whether + or − was used; for example, $A + 2 = C$ or $C - 2 = A$). Subjects were given $letter1$ and $number$, and asked to produce $letter2$.

In experiment 1 of Rabinowitz and Goldberg (1995), during the training phase, one group of subjects (the consistent group) received 36 blocks of training, in which each block consisted of the same 12 addition problems. Another group (the varied group) received 6 blocks of training, in which each block consisted of the same 72 addition problems. While both groups received 432 trials, the consistent group practiced on each problem 36 times, but the varied group only 6 times. In the transfer phase, each group received 12 new addition problems (not practiced before), repeated 3 times. The findings were that, at the end of training, the consistent group performed far better than the varied group. However, during transfer, the consistent group performed worse than the varied group. The varied group showed perfect transfer, while the consistent group showed considerable slow-down. See Figures 4.

In experiment 2, the training phase was identical to that of experiment 1. However, during the transfer phase, both groups received 12 subtraction (not addition) problems, which were the reverse of the original addition problems, repeated 3 times. The findings were that, in contrast to experiment 1, during transfer, the consistent group actually performed better than the varied group. Both groups performed worse than their corresponding performance at the end of training, but the varied group showed worse performance than the consistent group. See Figure 5.

How do we make sense of this complex pattern of data? Simulations were conducted based on CLARION. The CLARION model received input and generated output exactly as described earlier (given $letter1$ and $number$ and asked to produce $letter2$). The model generated the output using both explicit knowledge at the top level and implicit knowledge (in the form of neural networks) at the bottom level, both of which were learned through experiences with the task. The simulation captured the data pattern, and provided process-based explanations for it. See the simulation data

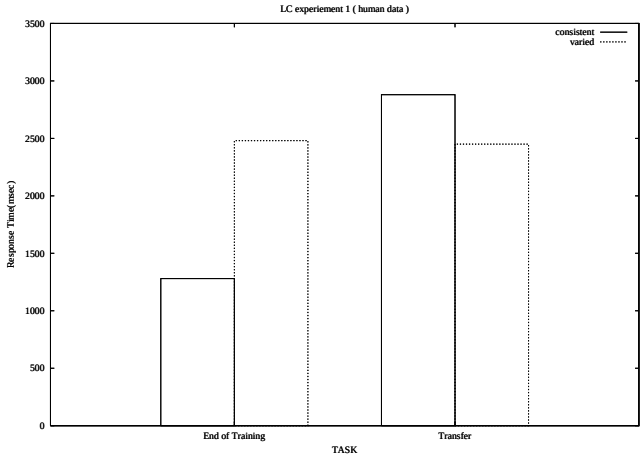


Figure 4: Experiment 1 of the letter counting task (from Rabinowitz and Goldberg 1995).

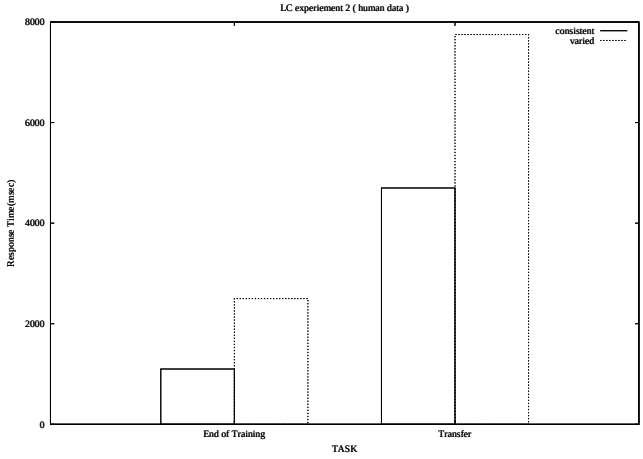


Figure 5: Experiment 2 of the letter counting task (from Rabinowitz and Goldberg 1995).

in Figure 6, which is to be compared with Figure 4. During the training phase of experiment 1, the simulated consistent group had a lower response time, because it had more practice on a smaller number of instances, which led to the better performing bottom level in the action-centered subsystem, as well as better performing instance retrieval from the top level of the non-action-centered subsystem. [12] Because they were better performing, the bottom level of the action-centered subsystem and the top level of the non-action-centered subsystem were more likely to be used in determining the overall outcome of the simulated consistent group, due to the competition among different components. [13] Because these two components had lower response times than other components, [14] a lower overall response time resulted for the simulated consistent group.

CLARION also matched the transfer performance difference between the two groups in experiment 1 (as shown in Figure 6). During the transfer phase of experiment 1, the performance of the simulated consistent group was worsened, compared with its performance at the end of training; the transfer performance of the simulated consistent group was in fact worse than that of the simulated varied group. This is because the simulated consistent group relied more on the bottom level of the action-centered subsystem and the non-action-centered subsystem during training and therefore, the activations of its counting rules (in the top level of the action-centered subsystem, for counting up letter by letter to capture addition), were lower. As a result, it took more time to apply the counting rules during transfer, which it had to apply, due to the fact that it had to deal with a different set of instances during transfer. The performance of the simulated varied group hardly changed, compared with its performance at the end of training, because it relied mostly on the counting rules at the top level of the action-centered subsystem during training (which was equally applicable to both training and transfer). As a result, its counting rules had higher activations, and therefore it performed better than the simulated consistent group during transfer. See Sun and Zhang (2003) for all the related statistical analysis.

As indicated by Figure 7, which is to be compared to Figure 5, the simulation also captured accurately the human data of experiment 2. During the transfer in experiment 2, due to the change in the task setting (counting down as opposed to counting up, due to the use of subtraction as opposed to addition), the practiced rule for counting up was no longer useful. Therefore, both simulated groups had to use a new counting rule (for counting down), which had only the initial low activation for both groups. Similarly, both simulated groups had to use a new instance retrieval rule (for "reverse retrieval" of chunks such as "A + 2 =C"), which also had only the initial low

[12] The bottom level in the action-centered subsystem of the simulated consistent group performed more accurately because of a more focused practice on a few instances by the consistent group (compared with the varied group). The top level of the non-action-centered subsystem of the simulated consistent group was more accurate for the same reason.

[13] We looked at the data, and indeed there were a lot more retrievals from the non-action-centered subsystem by the simulated consistent group than the simulated varied group (Sun and Zhang 2003). The data also showed a higher selection probability of the bottom level of the action-centered subsystem in the simulated consistent group (Sun and Zhang 2003).

[14] It was either inherently so, as in the case of the bottom level of the action-centered subsystem, or due to more frequent use (and consequently higher activations), as in the case of the top level of the non-action-centered subsystem.

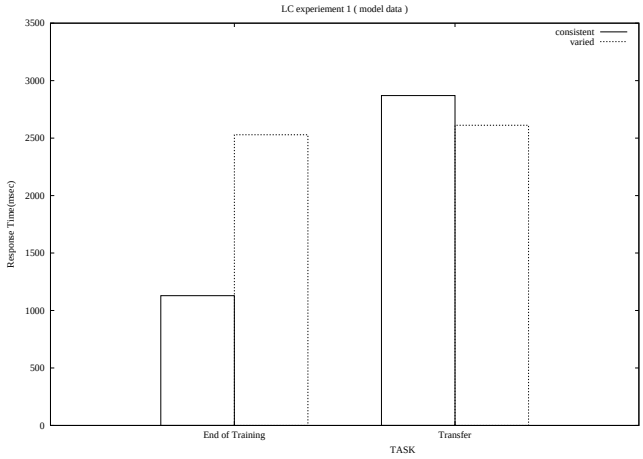


Figure 6: Simulation of experiment 1 of the letter counting task.

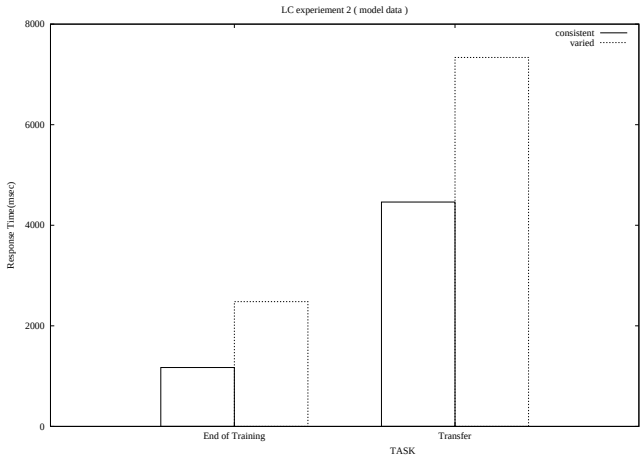


Figure 7: Simulation of experiment 2 of the letter counting task.

activation in both cases. [15] Both simulated groups performed worse than at the end of training for the above reason.

Moreover, this simulation captured the fact that the varied group performed worse than the consistent group during transfer (Figure 7). This difference was explained by the fact that the simulated consistent group had more activations associated with instance chunks (such as "A + 2 =C") than the simulated varied group, because the simulated consistent group had more practice with these chunks. Therefore, the simulated consistent group performed better than the simulated varied group in this phase.

CLARION provides some interesting interpretations of the human data. For example, it attributes in part the performance difference at the end of training between the consistent and the varied group to the difference between relying on implicit knowledge and relying on explicit rules.

[15] Chunks were used in "reverse retrieval" during the transfer phase of experiment 2, because of the reverse relationship between the training and the transfer instances used in this experiment.

Human Data

	Sugar Task	Person Task
control	1.97	2.85
original	2.57	3.75
memory training	4.63	5.33
simple rule	4.00	5.91

Figure 8: The human data from Stanley et al (1989). Each data point indicates the number of on-target responses per trial block.

Moreover, the CLARION simulation was far more accurate than the corresponding ACT-R simulation (Johnson 1998). This fact suggests, to some extent, the advantage of CLARION.

For another instance of simulation, let us look into the simulation of the process control task of Stanley et al (1989). In Stanley et al (1989), two versions of a process control task were used. In the "person" version, subjects were to interact with a computer simulated "person" whose behavior ranged from "very rude" to "loving" (over a total of 12 levels) and the task was to maintain the behavior of the simulated "person" at "very friendly" by controlling his/her own behavior (which could also range over the 12 levels, from "very rude" to "loving"). In the sugar production factory version, subjects were to interact with a simulated factory to maintain a particular production level (out of a total of 12 possible production levels), through adjusting the size of the workforce (which also had 12 levels). In either case, the behavior of the simulated system was determined by $P = 2 * W - P_1 + N$, where P was the current system output, P_1 was the previous system output, W was the subjects' input to the system, and N was noise. [16]

There were four groups of subjects. The control group was not given any explicit how-to instruction and not asked to verbalize. The "original" group was required to verbalize after each block of 10 trials. Other groups of subjects were given explicit instructions in various forms: To the "memory training" group, a series of 12 correct input/output pairs was presented. To the "simple rules" group, a simple heuristic rule ("always select the response level half way between the current production level and the target level") was given. All the subjects were trained for 200 trials (20 blocks of 10 trials). The human performance data is as shown in Figure 8.

A simulation was conducted based on CLARION, in which the model received the current and previous system states as input and generated adjustments to the system as output, using both (learned) explicit knowledge at the top level and (learned) implicit knowledge at the bottom level (Sun et al 2006). The simulation based on CLARION fairly accurately captured the human data, as shown in Figure 9 (see Sun et al 2006 for the statistical analysis). However, to better understand the contributing factors in the model performance, we further performed a componential analysis of the model to discern the contributions of various constituting elements of the model. In particular we compared the contributions of the two different explicit learning methods—independent hypothesis

[16]Noise (N) was added to the output of the system, so that there was a chance of being up or down one level (a 33% chance respectively).

Model Data

	Sugar Task	Person Task
control	1.92	2.62
original	2.77	4.01
memory training	4.45	5.45
simple rule	4.80	5.65

Mean-Squared Errors

	Total	Sugar Task	Person Task
IDN+RER+IRL	0.113	0.178	0.048

Figure 9: The simulation of Stanley et al. (1989). Each data point indicates the number of on-target responses per trial block. IDN refers to the bottom level of the action-centered subsystem. RER refers to bottom-up rule extraction. IRL refers to independent hypothesis testing rule learning.

Mean-Squared Errors

	Total	Sugar Task	Person Task
IDN+RER	1.016	0.407	1.624
IDN+IRL	0.285	0.387	0.184

Figure 10: The simulation of Stanley et al. (1989) with partial models (i.e., with one component removed). IDN refers to the bottom level of the action-centered subsystem. RER refers to bottom-up rule extraction. IRL refers to independent hypothesis testing rule learning.

testing rule learning versus bottom-up rule extraction (that is, IRL versus RER, as discussed in Sun et al 2006). It was discovered that independent hypothesis testing rule learning (IRL) was a lot more important than bottom-up rule extraction (RER) in this case; that is, independent hypothesis testing rule learning was significantly more important in capturing the human data pattern than bottom-up rule extraction. See Figures 10 and 11 for some illustrations (in terms of MSEs).

In all of these cases, the simulation with the CLARION cognitive architecture forced one to think in terms of process, and in terms of details. For instance, in the afore-described simulation of the process control tasks, we investigated detailed computational processes involved in performing this task, in particular the two different explicit learning processes, and generated some conjectures regarding their relative importance.

CLARION also provides a deeper level of explanation. For example, in the case of simulating

Mean-Squared Errors

	Total	Sugar Task	Person Task
IDN+RER $\longrightarrow$ IRL	0.247	0.445	0.048
IDN+IRL $\longrightarrow$ RER	0.237	0.306	0.167

Figure 11: The simulation of Stanley et al. (1989) with partial-full models (i.e., models formed from partial models plus the missing component). IDN refers to the bottom level of the action-centered subsystem. RER refers to bottom-up rule extraction. IRL refers to independent hypothesis testing rule learning.

Agent/Org.	Team(B)	Team(D)	Hierarchy(B)	Hierarchy(D)
Human	50.0	56.7	46.7	55.0
Radar-Soar	73.3	63.3	63.3	53.3
CORP-P-ELM	78.3	71.7	40.0	36.7
CORP-ELM	88.3	85.0	45.0	50.0
CORP-SOP	81.7	85.0	81.7	85.0

Figure 12: Human and simulation data for the organizational decision task. D indicates distributed information access, while B indicates blocked information access. All numbers are percent correct.

the alphabetic arithmetic task, explanations were provided in terms of action-centered knowledge or non-action-centered knowledge, in terms of explicit knowledge or implicit knowledge, or in terms of activations of representational units (in explaining response time differences), and so on. They were deeper because the explanations were centered on lower-level mechanisms and processes.

Due to the nature of such deeper explanations, this approach is also likely to lead to unified explanations, unifying a large variety of data and/or phenomena. For example, all the aforementioned tasks have been explained computationally in a unified way in CLARION.

6.4 Accounting for Social Phenomena

One application of CLARION to social simulation was in understanding organizational decision making and the interaction between organizational structures and cognitive factors in affecting organizational decision making.

In terms of organizational structures, there are two major types: (1) teams, in which individual decisions are treated as votes and the organizational decision is the majority decision; and (2) hierarchies, which are characterized by agents organized in a chain of command, such that information is passed from subordinates to superiors, and the decision of a superior is based solely on the recommendations of his/her subordinates. In addition, organizations are distinguished by the structure of information accessible by each agent. Two varieties of information access are: (1) distributed access, in which each agent sees a different subset of attributes (no two agents see the same subset of attributes), and (2) blocked access, in which several agents may see exactly the same subset of attributes.

The experiments by Carley et al (1998) were done in a 2 x 2 fashion (organization x information access). In addition, human data for the experiment were compared to the results of the four models (Carley et al 1998). [17] See Figure 12.

[17] Among them, CORP-ELM produced the most probable classification based on an agent's own experience, CORP-P-ELM stochastically produced a classification in accordance with the estimate of the probability of each classification based on the agent's own experience, CORP-SOP followed organizationally prescribed standard operating procedure (which involved summing up the values of the attributes available to an agent) and thus was not adaptive, and Radar-Soar was a (somewhat) cognitive model built in Soar, which is based on explicit, elaborate search in problem spaces (Rosenbloom et al 1991).

Agent/Org.	Team(B)	Team(D)	Hierarchy(B)	Hierarchy(D)
Human	50.0	56.7	46.7	55.0
CLARION	53.2	59.3	45.0	49.4

Figure 13: Simulation data for agents running for 3,000 cycles. The human data from Carley et al (1998) are reproduced here. Performance of CLARION is computed as percent correct over the last 1,000 cycles.

In their work, the agent models used were simplistic, and the "intelligence" level was low. Moreover, learning in these simulations was rudimentary: there was no complex learning process as one might observe in humans. With these shortcomings in mind, it is worthwhile to undertake a simulation that involves more complex agent models that more accurately capture human cognition. Moreover, with the use of more cognitively realistic agent models, one may investigate individually the importance of different cognitive capacities and process details in affecting organizational performance (Sun and Naveh 2004).

Hence, a simulation was conducted with CLARION being used for modeling individual agents in an organization. In the simulation, each agent received a subset of information (as specified above) in each instance, and produced a decision using both implicit knowledge at the bottom level and explicit knowledge at the top level (both of which were learned through experiences). The results (see Figure 13) closely accorded with the patterns of the human data, with teams outperforming hierarchal structures, and distributed access proving superior to blocked access. Also, as in humans, performance was not grossly skewed towards one condition or the other, but was roughly comparable across all conditions, unlike some of the simulations in Carley et al (1998). The match with the human data was far better than in the simulations conducted by Carley et al (1998). The better match was due, at least in part, to a higher degree of cognitive realism in our simulation. See Sun and Naveh (2004) for further details, including the interesting effects of varying cognitive parameters.

Another application of CLARION to social simulation was in capturing and explaining the essential process of publication in academic science and its relation to cognitive processes. Science develops in certain ways. In particular, it has been observed that the number of authors contributing a certain number of articles to a scientific journal follows a highly skewed distribution, corresponding to an inverse power law, known as Lotka's law. Simon (1957) developed a simple stochastic process for approximating Lotka's law. One of the assumptions underlying this process was that the probability that a paper would be published by an author who had published i articles was equal to a/i^k, where a was a constant of proportionality. Using Simon's work as a starting point, Gilbert (1997) attempted to model Lotka's law. He obtains his simulation data based on some very simplified assumptions and a set of mathematical equations. To a significant extent, Gilbert's model was not cognitively realistic. The model assumed that authors were non-cognitive and interchangeable; it therefore neglected a host of cognitive phenomena that characterized scientific inquiry (e.g., learning, creativity, evolution of field expertise, etc.).

# of Papers	Actual	Simon's estimate	Gilbert's simulation	CLARION simulation
1	3991	4050	4066	3803
2	1059	1160	1175	1228
3	493	522	526	637
4	287	288	302	436
5	184	179	176	245
6	131	120	122	200
7	113	86	93	154
8	85	64	63	163
9	64	49	50	55
10	65	38	45	18
11 or more	419	335	273	145

Figure 14: Number of authors contributing to *Chemical Abstracts*.

Using a more cognitively realistic model, one could address some of these omissions, as well as exploring other emergent properties of a cognitively based model and their correspondence to real-world phenomena. Using CLARION, individual cognition (including learning) in the academic world was modeled. An individual came up with an idea in the form of a paper, based on the existing, publicly available ideas and his/her own internal cognitive processes (implicit and explicit knowledge learned at the bottom and the top level). That idea was then evaluated based on a number of criteria by the scientific community. It might be rejected, for example, because it was too close to an existing paper. If an individual failed to publish a sufficient number of papers within a certain period of time, the individual would be removed from the system (i.e., "publish or perish"). The results of the simulation based on CLARION are shown in Figures 14 and 15, along with the empirical data (reported by Simon 1957) for *Chemical Abstracts* and *Econometrica*, and the estimates obtained from the previous simulations by Simon (1957) and by Gilbert (1997). The figures in the tables indicate number of authors contributing to each journal, by number of papers each has published.

The CLARION simulation data for the two journals could be fit to the power curve $f(i) = a/i^k$, resulting in an excellent match. The results of the curve fit are shown in Figure 16, along with correlation and error measures.

Note that, in our simulation, the number of papers per author reflected the cognitive ability of an author, as opposed to being based on auxiliary assumptions such as those made by Gilbert (1997). This explains, in part, the greater divergence of our results from the human data: whereas Gilbert's simulation consists of equations selected to match the human data, our approach relies on much more detailed and lower-level mechanisms—namely, a cognitive agent model that is generic rather than task-specific. The result of the CLARION based simulation is therefore emergent, and not a result of specific and direct attempts to match the human data. That is, we put more distance between mechanisms and outcomes, which makes it harder to obtain a match with the human data.

# of Papers	Actual	Simon's estimate	Gilbert's simulation	CLARION simulation
1	436	453	458	418
2	107	119	120	135
3	61	51	51	70
4	40	27	27	48
5	14	16	17	27
6	23	11	9	22
7	6	7	7	17
8	11	5	6	18
9	1	4	4	6
10	0	3	2	2
11 or more	22	25	18	16

Figure 15: Number of authors contributing to *Econometrica*.

Journal	a	k	Pearson R	R-square	RMSE
CA	3806	1.63	0.999	0.998	37.62
E	418	1.64	0.999	0.999	4.15

Figure 16: Results of fitting CLARION data to power curves. CA stands for Chemical Abstracts and E stands for Econometrica.

Thus, the fact that we were able to match the human data reasonably well shows the power of our cognitive architecture based approach.

6.5 A Model for Autonomous Intelligent Systems

CLARION also serves as a model for building autonomous intelligent systems (that is, intelligent agents). We tried to apply CLARION to a few reasonably interesting tasks in this regard, including learning minefield navigation and so on.

As described in Sun and Peterson (1998), CLARION was used to tackle a complex simulated minefield navigation task. The task setting was as shown in Figure 17. The CLARION based agent had to navigate an underwater vessel through a minefield to reach a target location. The agent received information only from a number of instruments, as shown in Figure 18. The sonar gauge showed how close the mines were in 7 equal areas that range from 45 degrees to the left of the agent to 45 degrees to the right. The fuel gauge showed the agent how much time was left before fuel ran out. The bearing gauge showed the direction of the target from the present direction of the agent. The range gauge showed how far the target was from the current location. Using only this sparse information, the agent decided (1) how to turn and (2) how fast to move. The agent, within an allotted time period, could either (a) reach the target (which is a success), (b) hit a mine (a failure), or (c) run out of fuel (a failure). The agent was under severe time pressure, so it had to be reactive in decision making, and it had no time for reasoning, episodic memory retrieval,

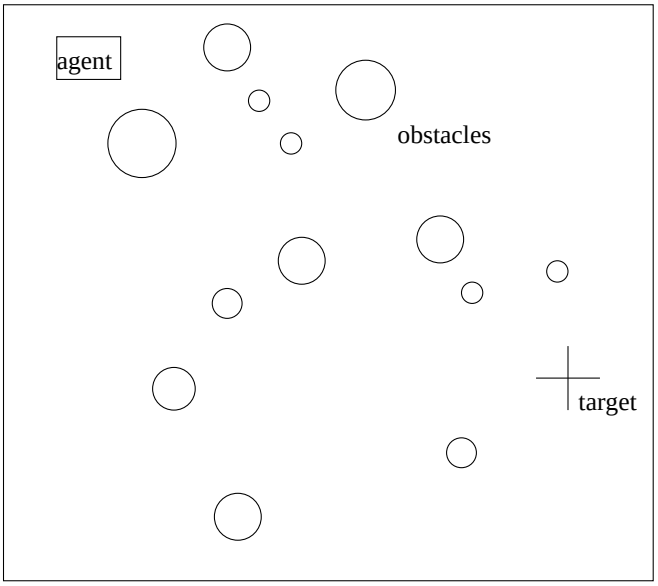


Figure 17: Navigating through mines.

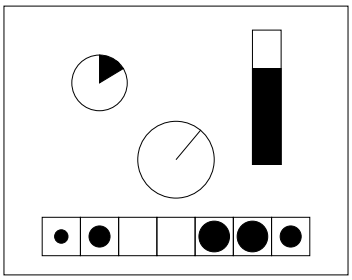
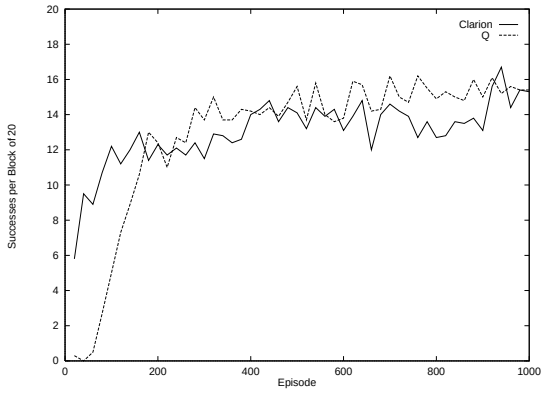


Figure 18: The navigation input. The display at the upper left corner is the fuel gauge; the vertical one at the upper right corner is the range gauge; the round one in the middle is the bearing gauge; the 7 sonar gauges are at the bottom.

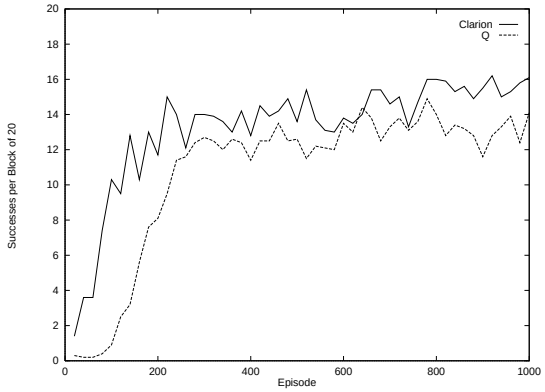
and other slow cognitive processes. This task was thus quite difficult. CLARION based agents were applied to learning this task, starting from scratch, without a priori knowledge. CLARION learned, through trial-and-error, first implicit knowledge at the bottom level and then on that basis explicit knowledge at the top level. CLARION learned to reach a high level of performance through such autonomous learning. See Figure 19 for learning curves. As shown in the figure, CLARION based agents outperformed regular reinforcement learning (i.e., Q-learning) significantly, due to the synergy between the two levels of the ACS.

7 The Challenges Ahead

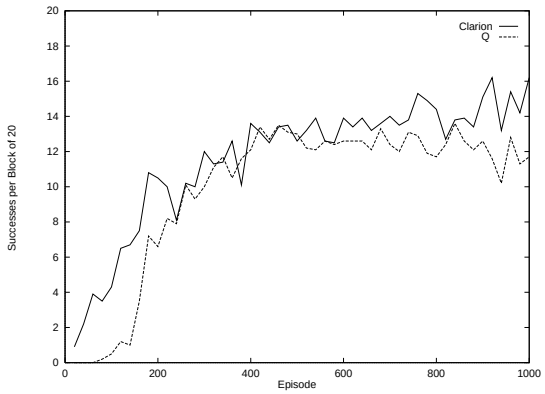
Let us look into some general and specific challenges in developing cognitive architectures in relation to cognitive science, social sciences, and AI/CI.



The 10-mine learning curves



The 30-mine learning curves



The 60-mine learning curves

Figure 19: Learning curves for 10-mine, 30-mine, and 60-mine settings.

In general, developing cognitive architectures is an extremely difficult task, because (1) a cognitive architecture needs to be compact but yet comprehensive in scope, (2) it needs to remain simple yet capture empirical data accurately, (3) it needs to be computationally feasible but also consistent with psychological theories, (4) it needs somehow to sort out and incorporate the myriad of incompatible psychological theories in existence, and so on.

7.1 The Challenges from Cognitive Science

Some have claimed that fundamental scientific discovery and grand scientific theorizing have become a thing of the past. What remains to be done is filling in details and refining some relatively minor points. Fortunately, many cognitive scientists believe otherwise. Researchers in cognitive science are pursuing integrative approaches that attempt to explain data in multiple domains and functionalities (Anderson and Lebiere 1998, Sun 2002). In cognitive science, as in many other scientific fields, significant advances may be made through discovering (hypothesizing and confirming) deep-level principles that unify superficial explanations across multiple domains, in a way somewhat analogous to Einstein's theory that unified electromagnetic and gravitational forces or String Theory that provides even further unifications (Greene 1999). Such theories, based on cognitive architectures, are exactly what cognitive science currently needs.

Cognitive scientists are actively pursuing such theories and, hopefully, will be increasingly doing so. Integrative models may serve as antidotes to the increasing specialization of research. Cognitive architectures that integrate a broad range of cognitive functionalities go against this trend of increasing specialization, and help to fit pieces together again (for example, multiple tasks may be handled in parallel by CLARION). As voiced by many cognitive scientists, the trend of over-specialization is harmful and the reversal of this trend is a necessary step toward further advances of cognitive science (Sun, Honavar, Oden 1999). Developing integrative cognitive architectures is thus a major challenge and a major opportunity in cognitive science.

In developing cognitive architectures, first of all, it is important to keep in mind a broad set of desiderata. For example, in Anderson and Lebiere (2003) a set of desiderata proposed by Newell (1990) was used to evaluate the ACT-R cognitive architecture versus conventional connectionist models. These desiderata include flexible behavior, real-time performance, adaptive behavior, vast knowledge base, dynamic behavior, knowledge integration, natural language, learning, development, evolution, and brain realization. These were considered constraints on cognitive architectures. In Sun (2004), another, broader set of desiderata was proposed and used to evaluate a wider set of cognitive architectures. These desiderata include ecological realism, bio-evolutionary realism, cognitive realism, and many others (see Sun 2004 for details). The advantages of coming up with and applying these sets of desiderata include (1) avoiding overly narrow models, (2) avoiding missing certain crucial functionalities, and (3) avoiding inappropriate approaches or techniques in implementing cognitive architectures. One can reasonably expect that this issue will be driving research in the field of cognitive architectures in the future. The challenge is to come up with

better, more balanced sets of desiderata.

Related to that, some general architectural principles also need to be examined. For example, it was claimed that the strengths of ACT-R derived from its tight integration of the symbolic and the subsymbolic component (Anderson and Lebiere 1998). On the other hand, it has been argued (Sun et al 2005) that the strength of CLARION lies in its partitioning of cognitive processes into two broad categories: implicit and explicit processes (as well as other categories). It is important to explore such broad issues in empirical work, through experimental, simulational, and theoretical means. It is a challenge to methodically explore such issues and reach some theoretically interesting conclusions.

The validation of process details of a cognitive architecture against empirical (e.g., psychological) data has been a difficult, but extremely important, issue. There have been too many instances in the past that research communities rushed into some particular model or some particular approach toward capturing cognition, without knowing exactly how much of the approach or the model was veridical. This often happens at the prompt of overly enthusiastic funding agencies or governmental panels. Often, without much effort at validation, claims are boldly made about the promise of a certain model or a certain approach. Unfortunately, we have seen quite a few setbacks in the history of cognitive science as a result of this cavalier attitude toward science. As in any other scientific fields, painstakingly detailed work needs to be carried out before sweeping claims can be made. This issue of validation of cognitive architectures poses a serious challenge, because of the myriad of mechanisms involved in cognitive architectures, and their variety and complexity. Detailed validation of cognitive architectures has been extremely difficult, unlike the validation of simple, narrowly scoped models. This challenge needs to be met by future research in this field.

Another challenge is how one can come up with a well constrained cognitive architecture with as few parameters as possible while accounting for a large variety of empirical observations and theories (Regier 2003). Complex models have always invoked suspicion in psychology. An exaggerated argument against generic models was examined in Miller et al (1960): "A good scientist can draw an elephant with three parameters, and with four he can tie a knot in its tail. There must be hundred of parameters floating around in this kind of theory and nobody will ever be able to untangle them". Counter-arguments to such objections can be advanced on the basis of the necessity of having complex models in understanding the mind, as argued in Miller et al (1960), Minsky (1985), Newell (1990), Sun (2002), and so on. However, it should be clearly recognized that over-generality, beyond what is minimally necessary, is always a danger in computational cognitive modeling, and in developing cognitive architectures in particular. Models may account well for a large set of data because of their extreme generality, rather than capturing any deep structures and regularities underlying cognitive processes in the human mind. Any cognitive model purporting to capture process details has to deal with this issue, not just cognitive architectures. This problem may be averted, by adopting a broad perspective (philosophical, psychological, biological, as well as computational), and by adopting a multi-level framework (going from sociological, to psychological,

to componential, and to biological levels), as discussed before and as argued in more detail in Sun et al (2005). Techniques have been developed to accomplish this end and more work is needed (see, e.g., Regier 2003).

While emphasizing the importance of being able to capture and explain the nuances of psychological data from experiments, we also need to emphasize the importance of full functionality in cognitive architectures, which often does not get the attention it deserves in cognitive science (and in experimental psychology in particular). As mentioned before, cognitive architectures need to incorporate all of the following functionalities: perception, categorization and concepts, memory, reasoning, decision making, planning, problem solving, motor control, learning, language and communication, meta-cognition, and others. This issue has been raised very early on, but is still a major challenge for cognitive science (Newell 1990, Anderson and Lebiere 2003).

Will this field eventually become a full fledged discipline—computational psychology, just like computational neuroscience or computational physics? This is an interesting but difficult question. There are a number of open issues to be sorted out. For example, how independent can this field be from closely allied fields such as experimental psychology? What will the new relationship be between data generation and modeling? How useful or illuminating can this field be in shedding new light on cognition per se (as opposed to leading up to building intelligent systems)? And so on and so forth. So far, the answers to these questions are by no means clear-cut. However, these are the factors that will determine the future of this field. We may have to wait for things to play out by themselves. The challenge is to better address these questions through research in a gradual and incremental way.

7.2 The Challenges from/to Artificial Intelligence

There have been various criteria proposed for evaluating cognitive architectures in the context of AI/CI. Langley and Laird (2003) proposed a set of useful criteria, including the following: (1) generality, versatility, and taskability, (2) both optimality and scalability (time/space complexity), (3) both reactivity and goal-directed behavior, (4) both autonomy and cooperation, (5) adaptation, learning, and behavioral improvements, and so on. To improve cognitive architectures in each of these aspects is a challenge by itself. These are challenges from the perspective of AI/CI to the field of cognitive architectures.

To develop better cognitive architectures, we need better algorithms, which we may find in various subfields within AI/CI. It is extremely important that AI/CI researchers come up with better and better algorithms, for various functionalities such as information filtering, information retrieval, learning, reasoning, decision making, problem solving, communication, and so on. Only on the basis of such key algorithms that are continuously improving, better and better cognitive architectures may be developed correspondingly.

In particular, we need better natural language processing capabilities, more efficient planning

algorithms, more powerful learning algorithms, and so on. Each of these types of algorithms could potentially significantly improve cognitive architectures, and cognitive architectures cannot be advanced without these algorithms. These are significant challenges from the field of cognitive architectures to AI/CI researchers.

AI/CI researchers also need to develop better computational methods (and algorithms) for putting the pieces together to form better overall architectures. Various pieces have been, or are being, developed by various subfields of AI/CI (including neural networks, reinforcement learning, and so on). Now it is the time to put them together to form a more coherent, better integrated, and better functioning cognitive architecture. Better computational algorithms are needed for this purpose. That is another place where AI/CI researchers can come in. It will be a long and incremental process —the challenge is to continuously improving upon the state of the art and to come up with architectures that better and better mirror the human mind and serve a variety of application domains at the same time.

Cognitive architectures may soon find both finer and broader applications, that is, both at lower levels and at higher levels (see the earlier discussion on levels). For example, some cognitive architectures found applications in large-scale simulation at a social, organizational level. For another example, some other cognitive architectures found applications in interpreting not only psychological data but also neuroimaging data (at a biological level). A review commissioned by the US National Research Council found that computational cognitive modeling had progressed to a degree that had made them useful in a number of application domains (Pew and Mavor 1998). Another review (Ritter, Shadbolt, Elliman, Young, Gobet, and Baxter 2003) reached similar conclusions. Both reviews provided descriptions of some examples of applications of cognitive architectures. Inevitably, this issue will provide challenges for future research (applied, as well as theoretical) in cognitive architectures. In this regard, cognitive architectures appear to be much more promising than simpler, narrowly scoped cognitive models (Pew and Mavor 1998).

7.3 The Challenges from Cognitive Social Simulation

An important development in the social sciences has been that of agent-based social simulation (ABSS). This approach consists of instantiating a population of agents, allowing the agents to run, and observing the interactions between them. [18] The use of agent-based social simulation as a means for computational study of societies mirrors the development of cognitive architectures in cognitive science. Thus, it is time to tackle sociality and social processes through cognitive architectures. So far, however, the two fields of social simulation and cognitive architectures have developed rather separately from each other (with some exceptions; e.g., Carley and Newell 1994,

[18] Agent-based social simulation thus differs markedly from traditional (equation-based) approaches to simulation, where relationships among conceptual entities (e.g., social groups and hierarchies, or markets and taxation systems) are expressed through a set of mathematical equations. Agent-based modeling has a number of advantages over equation-based modeling, including, notably, the ability to represent a heterogeneous population and to realistically model social networks (Axtell 2000).

Sun 2006). That is, most of the work in social simulation assumes very rudimentary cognition on the part of the agents (e.g., Cecconi and Parisi 1998).

The two fields of social simulation and cognitive architectures can be profitably integrated. This is an important challenge. As has been argued before (Sun and Naveh 2004; Moss 1999; Castelfranchi 2006), social processes ultimately rest on the choices and decisions of individuals, and thus understanding the mechanisms of individual cognition can lead to better theories describing the behavior of aggregates of individuals. Although most agent models in social simulation have been extremely simple, a more realistic cognitive agent model, incorporating realistic tendencies, inclinations and capabilities of individual cognitive agents can serve as a more realistic basis for understanding the interaction of individuals (Edmonds and Moss 2001). [19] Compared with more specialized, narrowly scoped models, cognitive architectures certainly have some significant advantages in this regard due to their generality and comprehensiveness (see Sun 2006).

At the same time, by integrating social simulation and cognitive modeling, one can arrive at a better understanding of individual cognition. Traditional approaches to cognitive modeling have largely ignored the potentially decisive effects of socially acquired and disseminated knowledge (including language, norms, beliefs, and so on; Zerubavel 1997). By modeling cognitive agents in a social context, one can learn more about the sociocultural processes that influence individual cognition.

The most fundamental challenge in this regard is to develop better ways of conducting detailed social simulation on the basis of cognitive architectures as building blocks. This is not an easy task. Although some initial work has been done (e.g., Sun and Naveh 2004, Sun 2006), much more work is needed. The challenges from social simulation to researchers of cognitive architectures include the issue of incorporating sociocultural representations, the issue of complexity and scalability, and so on.

One specific challenge is how to enhance cognitive architectures for the purpose of accounting for sociality in individual cognitive agents. There are many questions in this regard. For example, what are the characteristics of a proper cognitive architecture for modeling the interaction of cognitive agents? What additional sociocultural representations (for example, "motive", "obligation", or "norm") are needed in cognitive modeling of multi-agent interaction? See, for example, Sun (2006) and Castelfranchi (2006) for further discussions.

There is also the challenge of computational complexity and thus scalability that need to be addressed. Social simulation could involve a large number of agents, up to thousands. Computational complexity is thus already high, even without involving cognitive architectures as agent models. To incorporate cognitive architectures into social simulation, one has to deal with a great deal of added complexity. Thus, scalability is a big issue. Specialized, narrowly scoped cognitive models may be better at avoiding this problem, but they lack the generality and the comprehensiveness

[19] Although some cognitive details may ultimately prove to be irrelevant, this cannot be determined *a priori*, and thus simulations are useful in determining which aspects of cognition can be safely abstracted away.

that are so attractive for social simulation in general. Much more work is needed in this regard.

8 Concluding Remarks

Significant advances have been achieved in research on cognitive architectures. Nevertheless, there is still a long way to go before we can fully understand the inner working of the human mind and thereby develop computational models, viz. computational cognitive architectures, that faithfully replicate the human mind in all its capacities.

In this article, example cognitive architectures have been presented. However, beyond that, it will be wise to fully explore the space of possible computational cognitive architectures (Sloman and Chrisley 2005, Sun and Ling 1998), in order to further advance the state of the art in cognitive modeling. It will also be necessary to enhance the functionalities of cognitive architectures so that they will be capable of the full range of intelligent behaviors. Tasks of gradually increasing complexity should be tackled. Many challenges and issues should be addressed, including those stemming from cognitive science, from AI/CI, and from social simulation, as discussed earlier.

We have reasons to believe that the field of cognitive architectures will have a significant impact on cognitive science as well as on AI/CI and on social sciences (in terms of understanding cognition, in terms of developing artificially intelligent systems, and so on). It therefore should be considered an interesting intellectual challenge. Correspondingly, a significant amount of careful, methodical work should go into it.

Acknowledgments

This work was carried out while the author was supported in part by ARI grants DASW01-00-K-0012 and W74V8H-04-K-0002 (to Ron Sun and Bob Mathews). Thanks are due to (1) Artur d'Avila Garcez and Pascal Hitzler for the invitation to give a keynote talk at NeSy'05, (2) Rajiv Khosla for the invitation to give a plenary talk at KES'05, and (3) Dickson Lukose and Zhongzhi Shi for the invitation to give a keynote talk at PRIMA'05, all of which together led to the present article. Thanks are also due to Xi Zhang, Isaac (Yizchak) Naveh, Paul Slusarz, Todd Peterson, Bob Mathews, Sean Lane, and others for their collaborations on related research.

References

J. R. Anderson, (1983). *The Architecture of Cognition.* Harvard University Press, Cambridge, MA

J. Anderson and C. Lebiere, (1998). *The Atomic Components of Thought.* Lawrence Erlbaum Associates, Mahwah, NJ.

J. Anderson and C. Lebiere, (2003). The Newell Test for a theory of cognition. *Behavioral and Brain Sciences*, 26, 587-640

K. Carley and A. Newell, (1994). The nature of social agent. *Journal of Mathematical Sociology*, 19 (4), 221-262.

K. Carley, M. Prietula, and Z. Lin, (1998). Design versus cognition: The interaction of agent cognition and organizational design on organizational performance. *Journal of Artificial Societies and Social Simulation*, 1(3).

C. Castelfranchi, (2006). Cognitive architecture and contents for social structures and interactions. In: R. Sun (ed.), *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*. Cambridge University Press, New York.

F. Cecconi and D. Parisi, (1998). Individual versus social survival strategies. *Journal of Artificial Societies and Social Simulation*, 1 (2).
http://www.soc.surrey.ac.uk/JASSS/1/2/1.html

S. Chaiken and Y. Trope (eds.), (1999). *Dual Process Theories in Social Psychology*. Guilford Press, New York.

A. Clark and A. Karmiloff-Smith, (1993). The cognizer's innards: a psychological and philosophical perspective on the development of thought. *Mind and Language*. 8 (4), 487-519.

A. Cleeremans and J. McClelland, (1991). Learning the structure of event sequences. *Journal of Experimental Psychology: General*. 120. 235-253.

A. Cleeremans, A. Destrebecqz and M. Boyer, (1998). Implicit learning: News from the front. *Trends in Cognitive Sciences*, Volume 2, Issue 10, 406-416.

A. Collins and J. Loftus, (1975). Spreading Activation theory of semantic processing, *Psychological Review*, vol.82, pp.407-428.

P. Dayan, (2003). Levels of analysis in neural modeling. In: L. Nadel (ed.), *Encyclopedia of Cognitive Science*. Macmillan, London.

B. Edmonds and S. Moss, (2001). The importance of representing cognitive processes in multi-agent models. In: Dorffner G, Bischof H, and Hornik K (eds.). *Artificial Neural Networks–ICANN'2001*. Springer-Verlag, Berlin. pp. 759-766.

J. Fodor, (1983). *The Modularity of Mind*. MIT Press, Cambridge, MA.

N. Gilbert, (1997), A simulation of the structure of academic science. *Sociological Research Online*, 2(2). Available online at
http://www.socresonline.org.uk/socresonline/2/2/3.html.

G. Greene, (1999). *The Elegant Universe*. Norton, New York.

L. Hirschfield and S. Gelman (eds.), (1994). *Mapping the Mind: Domain Specificity in Cognition and Culture*. Cambridge University Press, Cambridge, UK.

E. Hutchins, (1995). How a cockpit remembers its speeds. *Cognitive Science*, 19, 265-288.

T. Johnson, (1998). Acquisition and transfer of declarative and procedural knowledge. *European Conference on Cognitive Modeling*, pp.15-22. Nottingham University Press, Nottingham, UK.

A. Karmiloff-Smith, (1986). From meta-processes to conscious access: Evidence from children's metalinguistic and repair data. *Cognition*. 23. 95-147.

J. Klahr, P. Langley, and R. Neches (1989). *Production System Models of Learning and Development*. MIT Press, Cambridge, MA.

J. Lave, (1988). *Cognition in Practice*. Cambridge University Press, Cambridge, England.

J. Mandler, (1992). How to build a baby. *Psychological Review*. 99, 4. 587-604.

D. Marr, (1982). *Vision*. W.H. Freeman: New York.

A. Maslow, (1987). *Motivation and Personality*. 3rd Edition. Harper and Row, New York.

J. McClelland, B. McNaughton and R. O'Reilly, (1995). Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102 (3), 419-457.

D. Medin, W. Wattenmaker, and R. Michalski, (1987). Constraints and preferences in inductive learning: An experimental study of human and machine performance. *Cognitive Science*. 11, 299-339.

R. Michalski, (1983). A theory and methodology of inductive learning. *Artificial Intelligence*. Vol.20, pp.111-161.

G. Miller, E. Galanter, and K. Pribram, (1960). *Plans and the Structure of Behavior*. Holt, Rinehart, and Winston, New York.

M. Minsky, (1985). *The Society of Mind*. Simon and Schuster, New York.

T. Nelson, (Ed.) (1993). *Metacognition: Core Readings*. Allyn and Bacon.

A. Newell, (1990). *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA.

A. Newell and H. Simon, (1976). Computer science as empirical inquiry: Symbols and search. *Communication of ACM*. 19. 113-126.

R. Nosofsky, T. Palmeri, and S. McKinley, (1994). Rule-plus-exception model of classification learning. *Psychological Review*. 101 (1), 53-79.

R. W. Pew and A. S. Mavor (eds), (1998). *Modeling Human and Organizational Behavior: Application to Military Simulations*. National Academy Press, Washington, DC.

M. R. Quillian, (1968). Semantic memory. In: M. Minsky (ed.), *Semantic Information Processing*. MIT Press, Cambridge, MA. pp.227-270.

M. Rabinowitz and N. Goldberg, (1995). Evaluating the structure-process hypothesis. In: F. Weinert and W. Schneider, (eds.) *Memory Performance and Competencies*. Lawrence Erlbaum, Hillsdale, NJ.

A. Reber, (1989). Implicit learning and tacit knowledge. *Journal of Experimental Psychology: General*. 118 (3), 219-235.

T. Regier, (2003). Constraining computational models of cognition. In: L. Nadel (ed.), *Encyclopedia of Cognitive Science*, pp.611-615. MacMillan Reference Ltd. London.

F. Ritter, Shadbolt, N., Elliman, D., Young, R., Gobet, F., and Baxter, G., (2003). *Techniques for Modeling Human Performance in Synthetic Environments: A Supplementary Review*. Human Systems Information Analysis Center, Wright-Patterson Air Force Base, Dayton, OH.

P. Rosenbloom, J. Laird, and A. Newell, (1993). *The SOAR Papers: Research on Integrated Intelligence*. MIT Press, Cambridge, MA.

D. Rumelhart, J. McClelland and the PDP Research Group, (1986). *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*. MIT Press, Cambridge, MA.

W. Schneider and W. Oliver (1991), An instructable connectionist/control architecture. In: K. VanLehn (ed.), *Architectures for Intelligence*, Erlbaum, Hillsdale, NJ.

C. Seger, (1994). Implicit learning. *Psychological Bulletin*. 115 (2), 163-196.

E. Servan-Schreiber and J. Anderson, (1987). Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology: Learning, Memory, and Cognition*. 16, 592-608.

H. Simon, (1957), *Models of Man, Social and Rational*. Wiley, NY.

A. Sloman and R. Chrisley, (2005). More things than are dreamt of in your biology: Information processing in biologically-inspired robots. *Cognitive Systems Research*, 6 (2), 145-174.

J. D. Smith, W. E. Shields, and D. A. Washburn, (2003). The comparative psychology of uncertainty monitoring and metacognition. *Behavioral and Brain Sciences*, in press.

W. Stanley, R. Mathews, R. Buss, and S. Kotler-Cope, (1989). Insight without awareness: On the interaction of verbalization, instruction and practice in a simulated process control task. *Quarterly Journal of Experimental Psychology*. 41A (3), 553-577.

R. Sun, (1994). *Integrating Rules and Connectionism for Robust Commonsense Reasoning*. John Wiley and Sons, New York, NY.

R. Sun, (1995). Robust reasoning: Integrating rule-based and similarity-based reasoning. *Artificial Intelligence*. 75, 2. 241-296.

R. Sun, (2001). Does connectionism permit rule learning? *INNS/ENNS/JNNS Newsletter*, No.44, pp.2-3. 2001.

R. Sun, (2001b). Cognitive science meets multi-agent systems: A prolegomenon. *Philosophical Psychology*, Vol.14, No.1, pp.5-28.

R. Sun, (2002). *Duality of the Mind*. Lawrence Erlbaum Associates, Mahwah, NJ.

R. Sun, (2003). *A Tutorial on CLARION*. Technical report, Cognitive Science Department, Rensselaer Polytechnic Institute.
http://www.cogsci.rpi.edu/∼rsun/sun.tutorial.pdf

R. Sun, (2004). Desiderata for cognitive architectures. *Philosophical Psychology*, 17 (3), 341-373.

R. Sun, (2006). Prolegomena to integrating cognitive modeling and social simulation. In: R. Sun (ed.), *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*. Cambridge University Press, New York.

R. Sun, (2007). Introduction to computational cognitive modeling. In: R. Sun (ed.), *The Cambridge Handbook of Computational Cognitive Modeling*. Cambridge University Press, New York.

R. Sun, (2007 b). The challenges of building computational cognitive architectures. In: W. Duch and J. Mandziuk (eds.), *Challenges in Computational Intelligence*. Springer-Verlag, Berlin.

R. Sun and L. Bookman, (eds.) (1994). *Computational Architectures Integrating Neural and Symbolic Processes*. Kluwer Academic Publishers. Norwell, MA.

R. Sun, L. A. Coward, and M. J. Zenzen, (2005). On levels of cognitive modeling. *Philosophical Psychology*, 18 (5), 613-637.

R. Sun, V. Honavar, and G. Oden, (1999). Integration of cognitive systems across disciplinary boundaries. *Cognitive Systems Research*, Vol.1, No.1, pp.1-3.

R. Sun and C. Ling, (1998). Computational cognitive modeling, the source of power and other related issues. *AI Magazine*. Vol.19, No.2, pp.113-120.

R. Sun, E. Merrill, and T. Peterson, (2001). From implicit skills to explicit knowledge: A bottom-up model of skill learning. *Cognitive Science*. Vol.25, No.2, 203-244.

R. Sun and I. Naveh, (2004). Simulating organizational decision making with a cognitive architecture CLARION. *Journal of Artificial Society and Social Simulation*, Vol.7, No.3, June, 2004. http://jasss.soc.surrey.ac.uk/7/3/5.html

R. Sun and T. Peterson, (1998). Autonomous learning of sequential tasks: experiments and analyses. *IEEE Transactions on Neural Networks*, Vol.9, No.6, pp.1217-1234.

R. Sun and T. Peterson, (1999). Multi-agent reinforcement learning: Weighting and partitioning. *Neural Networks*, Vol.12, No.4-5. pp.127-153.

R. Sun, P. Slusarz, and C. Terry, (2005). The interaction of the explicit and the implicit in skill learning: A dual-process approach. *Psychological Review*, 112 (1), 159-192.

R. Sun and X. Zhang, (2003). Accessibility versus action-centeredness in the representation of cognitive skills. *Proceedings of the Fifth International Conference on Cognitive Modeling*, pp.195-200. Universitats-Verlag Bamberg, Bamberg, Germany.

R. Sun, X. Zhang, P. Slusarz, and R. Mathews, (2006). The interaction of implicit learning, explicit hypothesis testing, and implicit-to-explicit extraction. *Neural networks*, in press.

F. Toates, (1986). *Motivational Systems*. Cambridge University Press, Cambridge, UK.

L. Vygotsky, (1986). *Mind in Society*. Lawrence Erlbaum Associates, Hillsdale, NJ.

C. Watkins, (1989). *Learning with Delayed Rewards*. Ph.D Thesis, Cambridge University, Cambridge, UK.

D. Willingham, M. Nissen, and P. Bullemer, (1989). On the development of procedural knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition.* 15, 1047-1060.

E. Zerubavel, (1997). *Social Mindscape: An Invitation to Cognitive Sociology.* Harvard University Press, Cambridge, MA.