The CLARION Cognitive Architecture: Extending Cognitive Modeling to Social Simulation

Ron Sun

October 11, 2004

1 Introduction

This chapter presents an overview of a relatively recent cognitive architecture for modeling cognitive processes of individual cognitive agents (in a psychological sense) (Sun et al 1998, 2001, Sun 2002). We will start with a look at some general ideas underlying this cognitive architecture and the relevance of these ideas to social simulation.

In the attempt to tackle a host of issues arising from computational cognitive modeling that are not adequately addressed by many other existent cognitive architectures, such as the implicit-explicit interaction, the cognitive-metacognitive interaction, and the cognitive-motivational interaction, CLARION, a modularly structured cognitive architecture, has been developed (Sun 2002, Sun et al 1998, 2001). Overall, CLARION is an integrative model. It consists of a number of functional subsystems (for example, the action-centered subsystem, the metacognitive subsystem, and the motivational subsystem). It also has a dual representational structure — implicit and explicit representations being in two separate components in each subsystem. Thus far, CLARION has been successful in capturing a variety of cognitive processes in a variety of task domains based on this division of modules (Sun et al 2002). See Figure 1 for a sketch of the architecture.

A key assumption of CLARION, which has been argued for amply before (see Sun et al 1998, 2001, Sun 2002), is the dichotomy of implicit and explicit cognition. Generally speaking, implicit processes are less accessible and more "holistic", while explicit processes are more accessible and more crisp

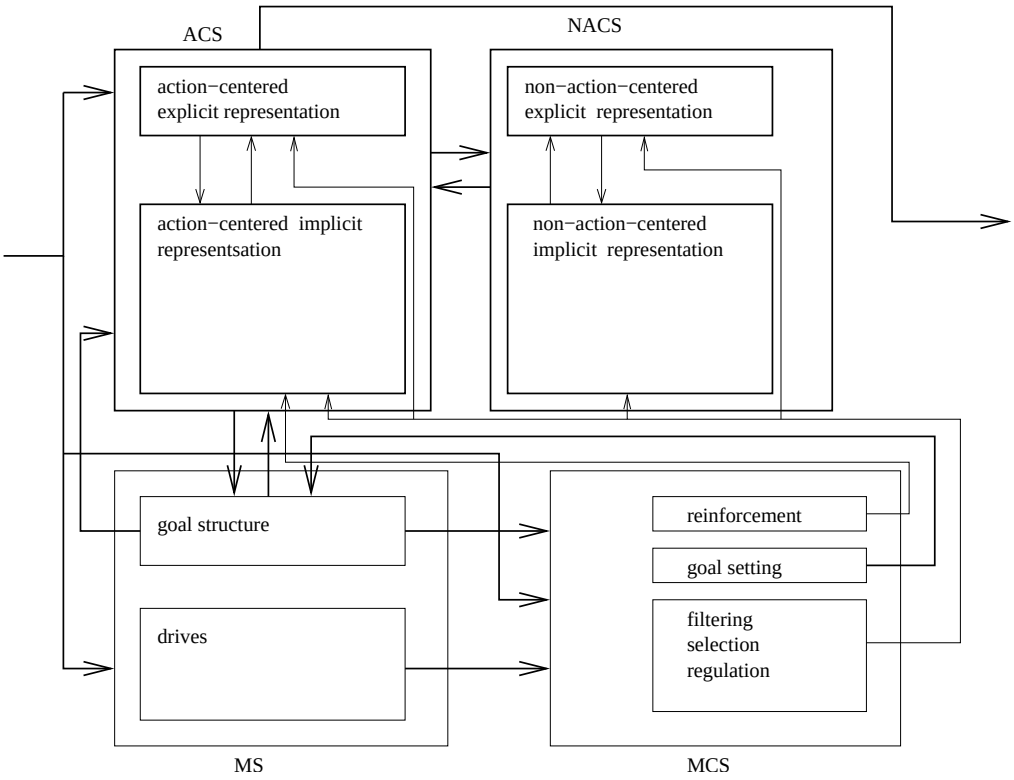

Figure 1: The CLARION architecture. ACS denotes the action-centered subsystem, NACS the non-action-centered subsystem, MS the motivational subsystem, and MCS the meta-cognitive subsystem.

(Reber 1989, Sun 2002). This dichotomy is closely related to some other well-known dichotomies in cognitive science: the dichotomy of symbolic versus subsymbolic processing, the dichotomy of conceptual versus subconceptual processing, and so on (Smolensky 1988, Sun 1994). The dichotomy can be justified psychologically, by the voluminous empirical studies of implicit and explicit learning, implicit and explicit memory, implicit and explicit perception, and so on (Reber 1989, Seger 1994, Cleeremans et al 1998, Sun 2002). In social psychology, there are similar dual-process models, for describing socially relevant cognitive processes (Chaiken and Trope 1999). Denoting more or less the same distinction, these dichotomies serve as justifications for the more general notions of implicit versus explicit cognition, which is the focus of CLARION. See Sun (2002) for an extensive treatment of this distinction.

Beside the above oft-reiterated point about CLARION, there are also a number of other characteristics that are especially pertinent to its application to social simulation, such as its focus on (1) the cognition-motivation-environment interaction, (2) the bottom-up and top-down learning, and (3) the cognitive-metacognitive interaction.

For instance, one particularly pertinent characteristic of this cognitive architecture is its focus on the cognition-motivation-environment interaction. Essential motivations of an agent, its biological needs in particular, arise naturally, prior to cognition (but interact with cognition of course). Such motivations are the foundation of action and cognition. In a way, cognition is evolved to serve the essential needs and motivations of an agent. Cognition, in the process of helping to satisfy needs and following motivational forces, has to take into account environments, their regularities and structures. Furthermore, some needs and motivations are inherently social or socially oriented. Thus, cognition bridges the needs and motivations of an agent and its environments (be it physical or social), thereby linking all three in a "triad" (see Chapter 1 of this book for more discussions).

Another important characteristic of this cognitive architecture is that an agent may learn on its own, regardless of whether or not there is a priori or externally provided domain knowledge. Learning may proceed on a trial-and-error basis. Furthermore, through a bootstrapping process, or "bottom-up learning" as has been termed (Sun et al 2001), explicit and abstract domain knowledge may be developed, in a gradual and incremental fashion (Karmiloff-Smith 1986). This is significantly different from other cognitive architectures (e.g., Anderson and Lebiere 1998). Likewise, in CLARION, it is

not necessary to have a priori explicit knowledge of needs, desires, and other motivational structures. Again, explicit knowledge of needs, desires, and motivations may be acquired through a bottom-up process, gradually and incrementally.

It should be noted that, although it addresses trial-and-error and bottom-up learning, the architecture does not exclude innate biases and innate behavioral propensities from being represented within the architecture. Innate biases and propensities may be represented, implicitly or even explicitly, and they interact with trial-and-error and bottom-up learning, in terms of constraining, guiding, and facilitating learning. In addition to bottom-up learning, top-down learning, that is, assimilation of explicit/abstract knowledge from external sources into implicit forms, is also possible in CLARION (Anderson and Lebiere 1998; Sun 2003).

Yet another important characteristic of this architecture is that multiple subsystems interact with each other constantly. In this architecture, these subsystems have to work closely with each other in order to accomplish cognitive processing. The interaction among these subsystems may include some "executive control" of some subsystems. It may also include metacognitive monitoring and control of on-going processing. It is worth noting that such cognitive-metacognitive interaction has not yet been fully addressed by other cognitive architectures such as ACT-R or Soar (but see, e.g., Sloman 2000). Note that social interaction is made possible by the (at least partially) innate ability of cognitive agents to reflect on, and to modify dynamically, their own behaviors (Tomasello 1999). The metacognitive self monitoring and control enables agents to interact with each other and with their environments more effectively, for example, by avoiding social impasse — impasse that are created because of the radically incompatible behaviors of multiple cognitive agents (see, for example, Sun 2001).

As mentioned earlier, the architecture also includes motivational structures, and therefore the interaction between motivational structures and other subsystems within the architecture is also prominent (again significantly different from other cognitive architectures such as ACT-R and Soar). This characteristic is also important for social interaction. Each agent in a social situation carries with it its own needs, desires, and motivations. Social interaction is possible in part because agents can understand and appreciate each other's (innate or acquired) motivational structures (Tomasello 1999, Bates et al 1992). On that basis, agents may find ways to cooperate.

In the remainder of this chapter, first, the overall structure of CLARION is presented in the next section. Then, each subsystem is presented in subsequent sections. Together, these sections substantiate all of the characteristics of CLARION discussed earlier. A discussion section follows, which addresses some general issues in extending cognitive modeling to social simulation with CLARION. It further explicates how these characteristics discussed earlier support cognitive modeling and social simulation in substantial ways. A summary section then completes this chapter.

2 The Overall Architecture

CLARION is intended for capturing all the essential cognitive processes within an individual cognitive agent. As mentioned before, 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 metacognitive subsystem (the MCS). The role of the ACS is to control actions, regardless of whether the actions are for external physical movements or internal mental operations. The role of the NACS is to maintain general knowledge, either implicit or explicit. The role of the MS 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 MCS is to monitor, direct, and modify the operations of the ACS 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; this distinction has been argued for earlier (see also Reber 1989, Seger 1994, and Cleeremans et al 1998). Let us consider the representational forms that need to be present for encoding these two different types of knowledge. Notice the fact that the relatively inaccessible nature of implicit knowledge may be captured by subsymbolic, distributed representation provided, for example, by a backpropagation network (Rumelhart et al 1986). This is because distributed representational units in the hidden layer(s) of a backpropagation network are capable of accomplishing computations but are subsymbolic and generally not individually meaningful

(Rumelhart et al 1986, Sun 1994). This characteristic of distributed representation, which renders the representational form less accessible, accords well with the relative inaccessibility of implicit knowledge (Reber 1989, Seger 1994, Cleeremans et al 1998). In contrast, explicit knowledge may be captured in computational modeling by symbolic or localist representation (Clark and Karmiloff-Smith 1993), in which each unit is more easily interpretable and has a clearer conceptual meaning. This characteristic of symbolic or localist representation captures the characteristic of explicit knowledge being more accessible and more manipulable (Smolensky 1988, Sun 1994).

Accessibility here refers to the direct and immediate availability of mental content for the major operations that are responsible for, or concomitant with, consciousness, such as introspection, forming higher-order thoughts, and verbal reporting. The dichotomous difference in the representations of the two different types of knowledge leads naturally to a two-level architecture, whereby each level uses one kind of representation and captures one corresponding type of process (implicit or explicit).

Let us now turn to learning. First, there is the learning of implicit knowledge at the bottom level. One way of implementing a mapping function to capture implicit knowledge is to use a multi-layer neural network (e.g., a three-layer backpropagation network). Adjusting parameters of this mapping function to change input/output mappings (that is, learning implicit knowledge) may be carried out in ways consistent with the nature of distributed representation (e.g., as in backpropagation networks), through trial-and-error interaction with the world. Often, reinforcement learning can be used (Sun et al 2001), especially Q-learning (Watkins 1989), implemented using backpropagation networks. In this learning setting, there is no need for a priori knowledge or external teachers providing desired input/output mappings. On the other hand, in the learning settings where desired input/output mappings are available, straight backpropagation (a supervised learning algorithm) can be used (Rumelhart et al 1986). Such (implicit) learning may be justified cognitively. For instance, Cleeremans (1997) argued at length that implicit learning could not be captured by symbolic models but neural networks. Sun (1999) made similar arguments.

Explicit knowledge at the top level can also be learned in a variety of ways (in accordance with localist/symbolic representation used there). Because of its representational characteristics, one-shot learning (for example, based on hypothesis testing) is preferred during interaction with the world (Bruner et al 1956, Busemeyer and Myung 1992, Sun et al 2001). With such learning, an agent

explores the world, and dynamically acquires representations and modifies them as needed.

The implicit knowledge already acquired in the bottom level may be utilized in learning explicit knowledge at the top level, through *bottom-up learning* (Sun et al 2001). That is, information accumulated in the bottom level through interacting with the world is used for extracting and then refining explicit knowledge. This is a kind of "rational reconstruction" of implicit knowledge at the explicit level. Conceivably, other types of learning of explicit knowledge are also possible, such as explicit hypothesis testing without the help of the bottom level. Conversely, once explicit knowledge is established at the top level, it may be assimilated into the bottom level. This often occurs during the novice-to-expert transition in instructed learning settings (Anderson and Lebiere 1998). The assimilation process, known as *top-down learning* (as opposed to bottom-up learning), may be carried out in a variety of ways (Anderson and Lebiere 1998, Sun 2003).

Figure 1 presents a sketch of this basic architecture of a cognitive agent, which includes the four major subsystems interacting with each other. The following four sections will describe, one by one and in more detail, these four subsystems of CLARION.

3 The Action-Centered Subsystem

The action-centered subsystem (the ACS) of CLARION is meant to capture the action decision making of an individual cognitive agent in its interaction with the world (see also the chapter by Shell and Mataric in this book). The ACS is the most important part of CLARION. In the ACS, the process for action decision making is essentially the following: Observing the current state of the world, the two levels of processes within the ACS (implicit or explicit) make their separate decisions in accordance with their own knowledge, and their outcomes are somehow "combined". Thus, a final selection of an action is made and the action is then performed. The action changes the world in some way. Comparing the changed state of the world with the previous state, the agent learns (in accordance with Q-learning of Watkins 1989 as mentioned earlier). The cycle then repeats itself.

In this subsystem, the bottom level is termed the IDNs (the Implicit Decision Networks), implemented with neural networks involving distributed representations, and the top level is termed the ARS (the Action Rule Store), implemented using symbolic/localist representations.

The overall algorithm for action decision making by an agent during its interaction with the world is as follows:

1. Observe the current state x.
2. Compute in the bottom level (the IDNs) the "value" of each of the possible actions (a_i's) associated with the state x: $Q(x, a_1)$, $Q(x, a_2)$, ..., $Q(x, a_n)$. Stochastically choose one action according to these values.
3. Find out all the possible actions (b_1, b_2, ..., b_m) at the top level (the ARS), based on the the current state x (which goes up from the bottom level) and the existing rules in place at the top level. Stochastically choose one action.
4. Choose an appropriate action, by stochastically selecting the outcome of either the top level or the bottom level.
5. Perform the action, and observe the next state y and (possibly) the reinforcement r.
6. Update the bottom level in accordance with an appropriate algorithm (to be detailed later), based on the feedback information.
7. Update the top level using an appropriate algorithm (for extracting, refining, and deleting rules, to be detailed later).
8. Go back to Step 1.

The input (x) to the bottom level consists of three sets of information: (1) sensory input, (2) working memory items, (3) the selected item of the goal structure. The sensory input is divided into a number of input dimensions, each of which has a number of possible values. The goal input is also divided into a number of dimensions. The working memory is divided into dimensions as well. Thus, input state x is represented as a set of dimension-value pairs: $(d_1, v_1)(d_2, v_2)....(d_n, v_n)$.

The output of the bottom level is the action choice. It consists of three groups of actions: working memory actions, goal actions, and external actions. [1]

In each network (encoding implicit knowledge), actions are selected based on their values. A Q

[1]Note that afore-mentioned working memory is for storing information temporarily for the purpose of facilitating subsequent decision making (Baddeley 1986). Working memory actions are used either for storing an item in the working memory, or for removing an item from the working memory. Goal structures, a special case of working memory, are for storing goal information specifically.

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. At each step, given state x, the Q values of all the actions (i.e., $Q(x, a)$ for all a's) are computed. Then the Q values are used to decide probabilistically on an action to be performed, through a Boltzmann distribution of Q values:

$$p(a|x) = \frac{e^{Q(x,a)/\alpha}}{\sum_i e^{Q(x,a_i)/\alpha}} \tag{1}$$

where α controls the degree of randomness (temperature) of the decision-making process. (This method is also known as Luce's choice axiom; Watkins 1989.)

The *Q-learning* algorithm (Watkins 1989), a reinforcement learning algorithm, is used for learning implicit knowledge at the bottom level. In the algorithm, $Q(x, a)$ estimates the maximum (discounted) total reinforcement that can be received from the current state x on. Q values are gradually tuned, on-line, through successive updating, which enables reactive sequential behavior to emerge through trial-and-error interaction with the world. Q-learning is implemented in backpropagation networks (see Sun 2003 for details).

Next, explicit knowledge at the top level (the ARS) is captured by *rules* and *chunks*. The condition of a rule, similar to the input to the bottom level, consists of three groups of information: sensory input, working memory items, and the current goal. The output of a rule, similar to the output from the bottom level, is an action choice. It may be one of the three types: working memory actions, goal actions, and external actions. The condition of a rule constitutes a distinct entity known as a chunk; so does the conclusion of a rule.

Specifically, rules are in the following form: *state-specification* $\longrightarrow$ *action*. The left-hand side (the condition) of a rule is a conjunction (i.e., logic AND) of individual elements. Each element refers to a dimension x_i of state x, specifying a value range, for example, in the form of $x_i \in (v_{i1}, v_{i2}, ..., v_{in})$. The right-hand side (the conclusion) of a rule is an action recommendation.

The structure of a set of rules may be translated into that of a network at the top level. Each value of each state dimension (i.e., each feature) is represented by an individual node at the bottom level (all of which together constitute a distributed representation). Those bottom-level feature nodes relevant to the condition of a rule are connected to the single node at the top level representing that condition, known as a chunk node (a localist representation). When given a set of rules, a rule network can be

wired up at the top level, in which conditions and conclusions of rules are represented by respective chunk nodes, and links representing rules are established that connect corresponding pairs of chunk nodes.

To capture the *bottom-up learning* process (Stanley et al 1989, Karmiloff-Smith 1996), the Rule-Extraction-Refinement algorithm (RER) learns rules at the top level using information in the bottom level. The basic idea of bottom-up learning of action-centered knowledge is as follows: If an action chosen (by the bottom level) is successful (i.e., it satisfies a certain criterion), then an explicit rule is extracted at the top level. Then, in subsequent interactions with the world, the rule is refined by considering the outcome of applying the rule: If the outcome is successful, the condition of the rule may be generalized to make it more universal; if the outcome is not successful, then the condition of the rule should be made more specific and exclusive of the current case.

An agent needs a rational basis for making these above decisions. Numerical criteria have been devised for measuring whether a result is successful or not, used in deciding whether or not to apply these operations. The details of the numerical criteria measuring whether a result is successful or not can be found in Sun et al (2001). Essentially, at each step, positive and negative match counts are updated (through measuring whether a rule or a potential rule leads to a positive or negative outcome). Then, on that basis, an information gain measure is computed, which compares different rules and chooses better ones (by essentially comparing their respective positive match ratios). The aforementioned rule learning operations (extraction, generalization, and specialization) are determined and performed based on the information gain measure (see Sun 2003 for details).

On the other hand, in the opposite direction, the dual representation (implicit and explicit) in the ACS also enables *top-down learning*. With explicit knowledge (in the form of rules) in place at the top level, the bottom level learns under the guidance of the rules. That is, initially, the agent relies mostly on the rules at the top level for its action decision making. But gradually, when more and more knowledge is acquired by the bottom level through "observing" actions directed by the rules (based on the same Q-learning mechanism as described before), the agent becomes more and more reliant on the bottom level (given that the inter-level stochastic selection mechanism is adaptable). Hence, top-down learning takes place.

For the stochastic selection of the outcomes of the two levels, at each step, with probability P_{BL}, the outcome of the bottom level is used. Likewise, with probability P_{RER}, if there is at least one RER rule indicating a proper action in the current state, the outcome from that rule set (through competition based on rule utility) is used; otherwise, the outcome of the bottom level is used (which is always available). Other components may be included in a like manner. The selection probabilities may be variable, determined through a process known as "probability matching": that is, the probability of selecting a component is determined based on the relative success ratio of that component. There exists some psychological evidence for such intermittent use of rules; see, for example, Sun et al (2001).

In addition, a set of equations specifies the response times of different components of the ACS and their combination — the overall response time. Those response time equations are based on "base-level activation" — a priming mechanism with gradual fading activation (Anderson and Lebiere 1998; see Sun 2003 for details).

This subsystem has been used for simulating a variety of psychological tasks, including process control tasks in particular (Sun et al 2005). In process control tasks, participants were supposed to control a (simulated) sugar factory. The output of the sugar factory was determined by the current and past inputs from participants into the factory, often through a complex and non-salient relationship. In the ACS of CLARION, the bottom level acquired implicit knowledge (embodied by the neural network) for controlling the sugar factory, through interacting with the (simulated) sugar factory in a trial-and-error fashion. On the other hand, the top level acquired explicit action rules for controlling the sugar factory, mostly through bottom-up learning (as explained before). Different groups of participants were tested, including verbalization groups, explicit instruction groups, and explicit search groups (Sun et al 2005). Our simulation succeeded in capturing the learning results of different groups of participants, mainly through adjusting one parameter that was hypothesized to correspond to the difference among these different groups (that is, the probability of relying on the bottom level; Sun et al 2005).

Besides simulating process control tasks, this subsystem has been employed in simulating a variety of other important psychological tasks, including alphabetic arithmetic tasks, artificial grammar learning tasks, Tower of Hanoi, and so on, as well as social simulation tasks such as organizational decision making (see the chapter by Naveh and Sun in this book).

4 The Non-Action-Centered Subsystem

The non-action-centered subsystem (the NACS) is used for representing general knowledge about the world that is not action-centered, for the purpose of making inferences about the world. It stores such knowledge in a dual representational form (the same as in the ACS): that is, in the form of explicit "associative rules" (at the top level), as well as in the form of implicit "associative memory" (at the bottom level). Its operation is under the control of the ACS.

First, at the bottom level of the NACS, "associative memory" networks (AMNs for short) encode non-action-centered implicit knowledge. Associations are formed by mapping an input to an output. The regular backpropagation learning algorithm, for example, can be used to establish such associations between pairs of input and output (Rumelhart et al 1986).

On the other hand, at the top level of the NACS, a general knowledge store (the GKS) encodes explicit non-action-centered knowledge (cf. Sun 1994). As in the ACS, chunks are specified through dimensional values. The basic form of a chunk consists of a chunk id and a set of dimension-value pairs. A node is set up in the GKS to represent a chunk (which is a localist representation). The chunk node connects to its constituting features (i.e., dimension-value pairs) represented as individual nodes in the bottom level (a distributed representation in the AMNs). Additionally, in the GKS, links between chunks encode explicit associations between pairs of chunk nodes, which are known as associative rules. Such explicit associative rules may be formed (i.e., learned) in a variety of ways in the GKS of CLARION (Sun 2003).

On top of that, similarity-based reasoning may be employed in the NACS. A known (given or inferred) chunk may be compared with another chunk. If the similarity between them is sufficiently high, then the latter chunk is inferred.

Similarity-based and rule-based reasoning can be inter-mixed. As a result of mixing similarity-based and rule-based reasoning, complex patterns of reasoning may emerge. As shown by Sun (1994), different sequences of mixed similarity-based and rule-based reasoning capture essential patterns of human everyday (mundane, commonsense) reasoning.

As in the ACS, top-down or bottom-up learning may take place in the NACS, either to extract

explicit knowledge in the top level from the implicit knowledge in the bottom level, or to assimilate the explicit knowledge of the top level into the implicit knowledge in the bottom level.

As in the ACS, a set of equations determines the response times of different components within the NACS (again based on "base-level activation"; see Sun 2003).

The NACS of CLARION has been used to simulate a variety of psychological tasks. For example, in artificial grammar learning tasks, participants were presented with a set of letter strings. After memorizing these strings, they were asked to judge the grammaticality of new strings. Despite their lack of complete explicit knowledge about the grammar underlying the strings, they nevertheless performed well in judging new strings. Moreover, they were also able to complete partial strings in accordance with their implicit knowledge. The result showed that participants acquired fairly complete implicit knowledge although their explicit knowledge was fragmentary at best (Domangue et al 2004). In simulating this task, while the ACS was responsible for controlling the overall operation, the NACS was used for representing most of the relevant knowledge. The bottom level of the NACS acquired implicit associative knowledge that enabled it to complete partial strings. The top level of the NACS recorded explicit knowledge concerning sequences of letters in strings. When given partial strings, the bottom level or the top level might be used, or the two levels might work together, depending on circumstances. Based the above setup, our simulation succeeded in capturing fairly accurately human data in this task across a set of different circumstances (Domangue et al 2004). In addition, many other tasks have been simulated using the NACS.

Let us also look into social situations in which the representations of self and others are important (e.g., Tomasello 1999, Andersen and Chen 2002). The social-cognitive model of transference claims that in an encounter with a new person, an underlying representation of some significant others is activated in a perceiver, leading the perceiver to interpret the new person in ways derived from the stored representation and to respond accordingly. The information one stores for significant others constitutes a system of knowledge that can be activated and brought to the fore in similar contexts. Within CLARION, such representations may be constructed in simulation using both the NACS and the ACS. In the NACS, information about others is stored at both levels as usual: through implicit associative memory as well as through explicit associative rules. Similarity of a new person to a stored representation of a significant other may be detected within the NACS through the working of the two

levels, in ways as sketched earlier. In turn, the detected similarity may trigger associated inferences — deriving information about the new person from the stored information. Similar detection may occur in the ACS. However, in the ACS, instead of inferential processes, actions may be chosen in accordance with the detected similarity.

5 The Motivational Subsystem

Supervisory processes over the operations of the ACS and the NACS are made up of two subsystems in CLARION: the motivational subsystem and the metacognitive subsystem. The motivational subsystem (the MS) is concerned with drives and their interactions (Toates 1986). That is, it is concerned with why an agent does what it does—why an agent chooses the actions it takes. Simply saying that an agent chooses actions to maximize gains, rewards, or payoffs leaves open the question of what determines gains, rewards, or payoffs. The relevance of the motivational subsystem to the main part of the architecture, the ACS, lies primarily in the fact that it provides the context in which the goal and the reinforcement of the ACS are determined. It thereby influences the working of the ACS, and by extension, the working of the NACS.

As an aside, for several decades by now, criticisms of commonly accepted models of human motivations, for example in economics, have focused on their overly narrow views regarding motivations, for example, solely in terms of simple economic reward and punishment (economic incentives and disincentives). Many critics opposed the application of this overly narrow approach to social, behavioral, cognitive, and political sciences. Complex social motivations, such as desire for reciprocation, seeking of social approval, and interest in exploration, also shape human behavior. By neglecting these motivations, the understanding of some key social and behavioral issues (such as the effect of economic incentives on individual behavior) may be hampered. Similar criticisms may apply to work on reinforcement learning in AI (for example, Sutton and Barto 1998).

A set of major considerations that the motivational subsystem of an agent must take into account may be identified. Here is a set of considerations concerning drives as the main constructs (cf. Simon 1967, Tyrell 1993):

- *Proportional activation.* The activation of a drive should be proportional to corresponding offsets,

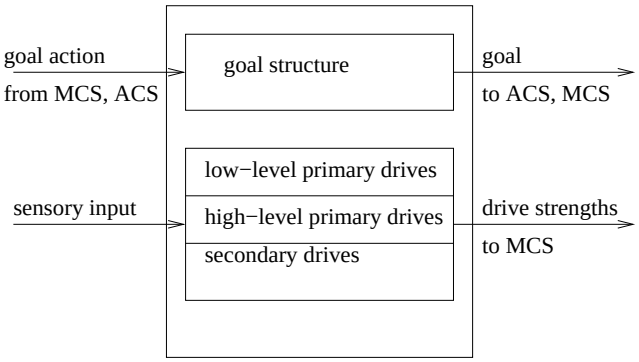

Figure 2: Structure of the motivational subsystem.

or deficits, in related aspects (such as food or water).

- *Opportunism.* An agent needs to incorporate considerations concerning opportunities. For example, the availability of water may lead to prefer drinking water over gathering food (provided that food deficits are not too great).

- *Contiguity of actions.* There should be a tendency to continue the current action sequence, rather than switching to a different sequence, in order to avoid the overhead of switching.

- *Persistence.* Similarly, actions to satisfy a drive should persist beyond minimum satisfaction, that is, beyond a level of satisfaction barely enough to reduce the most urgent drive to be slightly below some other drives. [2]

- *Interruption when necessary.* However, when a more urgent drive arises (such as "avoid-danger"), actions for a lower-priority drive (such as "get-sleep") may be interrupted.

- *Combination of preferences.* The preferences resulting from different drives should be combined to generate a somewhat higher overall preference. Thus, a compromise candidate may be generated that is not the best for any single drive but the best in terms of the combined preference.

A bipartite system of motivational representation is as follows (cf. Simon 1967, Nerb et al 1997). The explicit goals (such as "finding food") of an agent (which is tied to the working of the ACS, as

[2]For example, an agent should not run toward a water source and drink only a minimum amount, and then run toward a food source and eat a minimum amount, then going back to the water source to repeat the cycle.

explained before) may be generated based on internal drive states (for example, "being hungry") of the agent. This explicit representation of goals derives from, and hinges upon, (implicit) drive states. See Figure 2. [3]

Specifically, we refer to as *primary drives* those drives that are essential to an agent and are most likely built-in (hard-wired) to begin with. Some sample low-level primary drives include (cf. Tyrell 1993):

> **Get-food**. The strength of this drive is proportional to *0.95 ** max *(food-deficit, food-deficit * food-stimulus)*. The maximum strength of this drive is 0.95. The actual strength is determined by two factors: *food-deficit* felt by the agent, and the *food-stimulus* perceived by it.
>
> **Get-water**. The strength of this drive is proportional to *0.95 ** max *(water-deficit, water-deficit * water-stimulus)*. This situation is similar to *get-food*.
>
> **Avoid-danger**. The strength of this drive is proportional to *0.98 * danger-stimulus * danger-certainty*. The maximum strength of this drive is 0.98. It is proportional to the danger signal: its distance, severity (disincentive value), and certainty. The first two factors are captured by *danger-stimulus* (which is determined by distance and severity), and the third factor by *danger-certainty*. [4]

These drives may be implemented in a (pre-trained) backpropagation neural network, representing evolutionarily pre-wired instincts.

Beyond such low-level drives (concerning physiological needs), there are also higher-level drives. Some of them are primary, in the sense of being "hard-wired". The "need hierarchy" of Maslow (1987) identifies some of these drives. A few particularly relevant high-level drives include: **belongingness**, **esteem**, **self-actualization**, and so on (Sun 2003).

[3]Note that it is not necessarily the case that the two types of representations directly correspond to each other (e.g., one being extracted from the other), as in the case of the ACS or the NACS.

[4]Other drives include **get-sleep**, **reproduce**, and a set of "avoid saturation" drives, for example, **avoid-water-saturation** or **avoid-food-saturation**. There are also drives for **curiosity** and **avoid-boredom**. See Sun (2003) for further details.

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. Derived drives may include: (1) gradually acquired drives, through "conditioning" (Hull 1951); (2) externally set drives, through externally given instructions. For example, due to the transfer of the desire to please superiors into a specific desire to conform to his/her instructions, following the instructions becomes a (derived) drive.

Explicit goals may be set based on these (primary or derived) drives, as will be explored in the next section (Simon 1967, Nerb et al 1997).

6 The Meta-Cognitive Subsystem

Meta-cognition refers to one's knowledge concerning one's own cognitive processes and their outcomes. Meta-cognition also includes the active monitoring and consequent regulation and orchestration of these processes, usually in the service of some concrete goal (Flavell 1976, Mazzoni and Nelson 1998). This notion of metacognition is operationalized within CLARION.

In CLARION, the metacognitive subsystem (the MCS) is closely tied to the motivational subsystem. The MCS monitors, controls, and regulates cognitive processes for the sake of improving cognitive performance (Simon 1967, Sloman 2000). Control and regulation may be in the forms of setting goals for the ACS, interrupting and changing on-going processes in the ACS and the NACS, setting essential parameters of the ACS and the NACS, and so on. Control and regulation are also carried out through setting reinforcement functions for the ACS on the basis of drive states.

In this subsystem, many types of metacognitive processes are available, for different metacognitive control purposes. Among them, there are the following types (Sun 2003, Mazzoni and Nelson 1998):

(1) behavioral aiming:
setting of reinforcement functions
setting of goals

(2) information filtering:
focusing of input dimensions in the ACS

focusing of input dimensions in the NACS

(3) information acquisition:
selection of learning methods in the ACS
selection of learning methods in the NACS

(4) information utilization:
selection of reasoning methods in the ACS
selection of reasoning methods in the NACS

(5) outcome selection:
selection of output dimensions in the ACS
selection of output dimensions in the NACS

(6) cognitive mode selection:
selection of explicit processing, implicit processing, or a combination thereof (with proper integration parameters), in the ACS

(7) setting parameters of the ACS and the NACS:
setting of parameters for the IDNs
setting of parameters for the ARS
setting of parameters for the AMNs
setting of parameters for the GKS

Structurally, the MCS may be subdivided into a number of modules. The bottom level consists of the following (separate) networks: the goal setting network, the reinforcement function network, the input selection network, the output selection network, the parameter setting network (for setting learning rates, temperatures, etc.), and so on. In a similar fashion, the rules at the top level (if they exist) can be correspondingly subdivided. See Figure 3 for a diagram of the MCS. Further details, such as monitoring buffer, reinforcement functions (from drives), goal setting (from drives), information selection, and so on, can be found in Sun (2003).

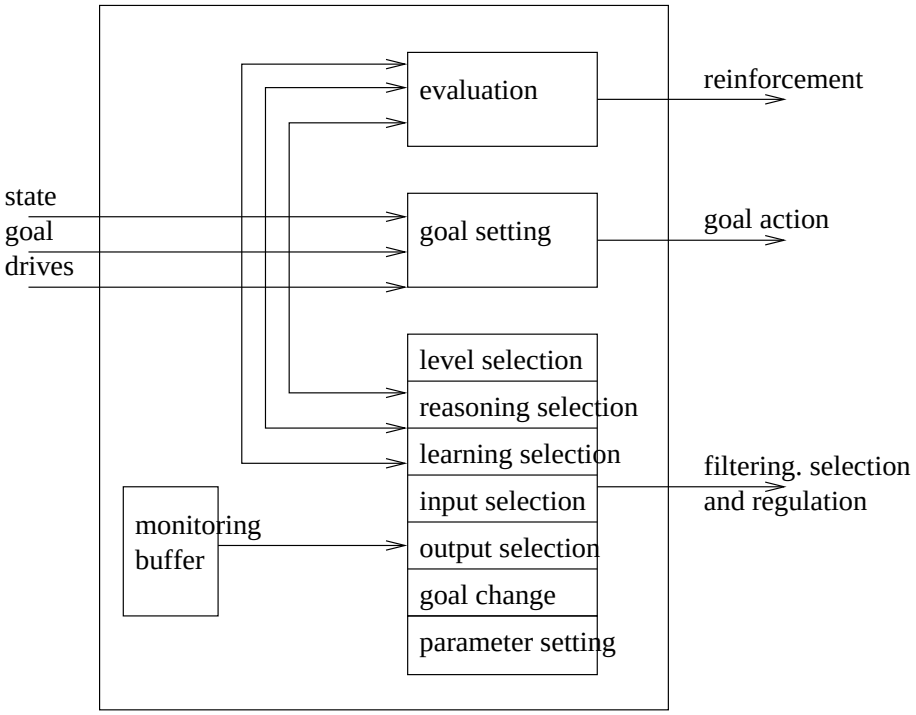

Figure 3: The structure of the metacognitive subsystem.

This subsystem may be pre-trained before the simulation of any particular task (to capture evolutionary pre-wired instincts, or knowledge/skills acquired from prior experience).

7 Discussions

Let us turn to the question of potential contributions of this cognitive architecture to cognitive modeling and social simulation. First of all, let us examine its contributions to computational cognitive modeling. Compared with other existent cognitive architectures, it is unique in that it contains (1) built-in motivational constructs, (2) built-in metacognitive constructs, (3) the separation of the two dichotomies: the dichotomy of implicit versus explicit representation and dichotomy of action-centered versus non-action-centered representation, and (4) both top-down and bottom-up learning. These features are not commonly found in other existing cognitive architectures. Nevertheless, we believe that these features are crucial to the enterprise of cognitive architectures, as they capture important elements in the interaction between an agent and its social and physical world.

For instance, without motivational constructs, a model agent would be literally aimless. It would wonder around the world aimlessly accomplishing hardly anything. Or it would have to rely on knowledge hand coded into it (for example, regarding goals and procedures) in order to accomplish some relatively minor things, usually only in a controlled environment. Or it would have to rely on external "feedback" (reinforcement, reward, punishment, etc.) in order to learn. But the requirement of external feedback begs the question of how such a signal is obtained in the natural world. In contrast, with the motivational subsystem as an integral part of CLARION, it is able to generate such feedback internally and learn on that basis, without requiring a "special" external feedback signal or externally provided and hand coded a priori knowledge (Edelman 1992).

Furthermore, with the two separate, built-in dichotomies, a variety of different types of knowledge may be represented. They include implicit and explicit action-centered knowledge, and implicit and explicit non-action-centered knowledge. These types of knowledge are not only important for modeling individual agents, but also important for modeling social interactions among agents. They capture habitual everyday routines for coping with the everyday world involving other agents, deliberate plans for specific tasks while taking into account other agents, general, explicit, conceptual knowledge about the world and about other agents, implicit associations (formed from prior experience) for priming other knowledge that may involve other agents, and so on. Cognitive models of agents would be much less capable without some of these knowledge types. Social simulation would, likewise, be much less realistic without some of these knowledge types.

On top of that, with the ability to learn in both top-down and bottom-up directions, CLARION captures more realistic learning capabilities of more cognitively realistic agents. The combination of these learning directions, especially bottom-up learning, enables the modeling of the complex interaction of an agent and its environment in learning a variety of different types of knowledge, in a variety of different ways (Sun et al 2001). In particular, they enable the capturing of complex sociocultural learning, among other situations.

Compared with existing social simulations, there are reasons to believe that CLARION has a lot to contribute towards more cognitively realistic social simulation. In existing social simulations, only very rudimentary models of agents have been used for the most part, without detailed, cognitively realistic processes and mechanisms (see, for example, Axelrod 1984, Gilbert and Doran 1994, Prietula

et al 1998, and so on), which may or may not serve well the intended purposes of these social simulations. Compared with such models, cognitive architectures provide a cognitively grounded way of understanding multi-agent interaction, by embodying realistic cognitive constraints and cognitive capabilities of individual agents in their interaction with their environments and with other agents, which may be highly relevant in many circumstances (see, for example, the chapters in Part 3 of this book). This is because cognitive architectures embody detailed (but generic) mechanisms and processes of individual cognition. This point about the importance of cognitive realism has been made by others too, for example, in the context of cognitive realism of game theory (Kahan and Rapaport 1984, Camerer 1997) and cognitive realism of social simulation (Edmonds and Moss 2001). We may even attempt to develop cognitive principles of sociocultural processes (e.g., Boyer and Ramble 2001, Atran and Norenzayan 2003).

CLARION has been successful in simulating a variety of psychological tasks. These 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 (see Sun 2002). Some of these tasks have been explained earlier. In addition, extensive work has been done on a complex minefield navigation task (Sun et al 2001). We have also tackled human reasoning processes through simulating reasoning data. Simulations involving motivational structures and metacognitive processes are also under way. Therefore, we are now in a good position to extend the effort on CLARION to the capturing of a wide range of social phenomena through integrating cognitive modeling and social simulation.

Let us take a brief look at some rather preliminary applications of CLARION to social simulation. In one instance, CLARION was substituted for simpler models previously used in organizational decision making modeling. An exploration was made of the interaction between cognitive parameters that govern individual agents, placement of agents in different organizational structures, and performance of the organization. By varying some factors and measuring the effect on collective performance, a better picture of the interaction between individual cognition and organizational decision making was arrived at (see the chapter by Naveh and Sun in Part 3 of this book). In another instance, CLARION was used to simulate the collective process of academic publication. CLARION reproduced the empirically observed power curves concerning number of publications, based on rather detailed modeling of individual cognitive processes involved. Various cognitive parameters were also tested

and various effects observed. In yet another instance, tribal societies were simulated, on the basis of CLARION modeling individual cognitive processes. In the simulation, different forms of social institutions (such as food distribution, law, political system, and enforcement of law) were investigated and related back to factors of individual cognition. Social institutions affect agents' actions and behaviors, which in turn affect social institutions. In this interaction, individual motivational factors are being taken into consideration, which include social norms, ethical values, social acceptance, empathy, imitation, and so on. The role of metacognitive control is also being investigated in this process. It has been suggested that such simulations are the best way to understand or to validate the significance of contributing cognitive, motivational, and metacognitive factors (see, e.g., chapter 1 in this book). The reader is referred to the chapters in Part 3 of this book for more examples of such social simulations.

8 Summary

In summary, this chapter covers the essentials of the CLARION cognitive architecture, and focuses in particular on the distinguishing features of the architecture. CLARION is distinguished by its inclusion of multiple, interacting subsystems: the action-centered subsystem, the non-action-centered subsystem, the motivational subsystem, and the metacognitive subsystem. It is also distinguished by its focus on the separation and the interaction of implicit and explicit knowledge (in these different subsystems, respectively). Different representational forms have been used for encoding these different types of knowledge, and different learning algorithms have been developed. Both top-down and bottom-up learning have been incorporated into CLARION. With these mechanisms, especially the motivational and metacognitive mechanisms, CLARION has something unique to contribute to cognitive modeling and social simulation.

For the full technical details of CLARION, see Sun (2003), which is available at http://www.cogsci.rpi.edu/~rsun/clarion-pub.html.

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

Acknowledgments

The work on CLARION has been supported in part by Army Research Institute contract DASW01-00-K-0012. Thanks are due to Xi Zhang, Isaac Naveh, Paul Slusarz, Robert Mathews, and many other collaborators, current or past. Thanks are also due to Jonathan Gratch, Frank Ritter, and Bill Clancey for their comments.

References

S. Andersen and S. Chen, (2002). The relational self: An interpersonal social-cognitive theory. *Psychological Review*, 109 (4), 619-645.

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

S. Atran and A. Norenzayan, (2003). Religion's evolutionary landscape: Counterintuition, commitment, compassion, and communion. *Behavioral and Brain Sciences*, in press.

R. Axelrod, (1984). *The Evolution of Cooperation.* Basic Books, New York.

A. Baddeley, (1986). *Working Memory.* Oxford University Press, New York.

J. Bates, A. Loyall, and W. Reilly, (1992). Integrating reactivity, goals, and emotion in a broad agent. *Proceedings of the 14th Meeting of the Cognitive Science Society.*

P. Boyer and C. Ramble, (2001). Cognitive templates for religious concepts: Cross-cultural evidence for recall of counter-intuitive representations. *Cognitive Science*, 25, 535-564.

J. Bruner, J. Goodnow, and J. Austin, (1956). *A Study of Thinking.* Wiley, New York.

J. Busemeyer and I. Myung, (1992). An adaptive approach to human decision making: Learning theory, decision theory, and human performance. *Journal of Experimental Psychology: General*, 121 (2), 177-194.

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, (1997). Principles for implicit learning. In D. Berry (Ed.), *How Implicit is Implicit Learning?* pp. 195-234. Oxford University Press, Oxford, UK.

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

T. Domangue, R. Mathews, R. Sun, L. Roussel, and C. Guidry, (2004). The effects of model-based and memory-based processing on speed and accuracy of grammar string generation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, in press.

G. Edelman, (1992). *Bright Air, Brilliant Fire.* Basic Books, New York.

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.* Lecture Notes in Computer Science, Vol.2130. pp.759-766. Springer-Verlag.

J. Flavell, (1976). Metacognitive aspects of problem solving. In: B. Resnick (ed.), *The Nature of Intelligence.* Erlbaum, Hillsdale, NJ.

N. Gilbert and J. Doran, (1994). *Simulating Societies: The Computer Simulation of Social Phenomena.* UCL Press, London, UK.

C. Hull, (1951). *Essentials of Behavior.* Yale University Press, New Haven, CT.

J. Kahan and A. Rapoport, (1984). *Theories of Coalition Formation.* Erlbaum, Mahwah, NJ.

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

B. Latane, (1981). The psychology of social impact. *American Psychologist*, 36, 343-356.

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

G. Mazzoni and T. Nelson, (eds.) (1998). *Metacognition and Cognitive Neuropsychology.* Erlbaum, Mahwah, NJ.

J. Nerb, H. Spada and A. Ernst, (1997). A cognitive model of agents in a common dilemma. *Proceedings of the 19th Cognitive Science Conference*, 560-565. Erlbaum, Mahwah, NJ.

M. Prietula, K. Carley, and L. Gasser (eds.), (1998). *Simulating Organizations: Computational Models*

of Institutions and Groups. MIT Press, Cambridge, MA.

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

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

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

A. Sloman, (2000). Architectural requirements for human-like agents both natural and artificial. In: *Human Cognition and Social Agent Technology*, K. Dautenhahn (ed.). John Benjamins, Amsterdam.

P. Smolensky, (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences*, 11 (1), 1-74.

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

R. Sun, (2001). Meta-learning in multi-agent systems. *Intelligent Agent Technology: Systems, Methodologies, and Tools*, N. Zhong, J. Liu, S. Ohsuga, and J. Bradshaw (eds.). World Scientific, Singapore.

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

R. Sun, (2003). *A Tutorial on CLARION 5.0*.
http://www.cogsci.rpi.edu/~rsun/sun.tutorial.pdf

R. Sun, (2004). Desiderata for cognitive architectures. *Philosophical Psychology*, in press.

R. Sun, E. Merrill, and T. Peterson, (1998). A bottom-up model of skill learning. *Proceedings of 20th Cognitive Science Society Conference*, 1037-1042, Lawrence Erlbaum Associates, Mahwah, NJ.

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, P. Slusarz, and C. Terry, (2004). The interaction of the explicit and the implicit in skill learning: A dual-process approach. *Psychological Review*, in press.

R. Sun, X. Zhang, P. Slusarz, and R. Mathews, (2005). The interaction of implicit learning, explicit hypothesis testing, and implicit-to-explicit extraction. Submitted.

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

M. Tomasello, (1999). *The Cultural Origins of Human Cognition.* Harvard University Press.

T. Tyrell, (1993). *Computational Mechanisms for Action Selection.* Ph.D Thesis, Oxford University, Oxford, UK.

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