PSYCHOLOGICALLY REALISTIC COGNITIVE AGENTS: TAKING HUMAN COGNITION SERIOUSLY

Ron Sun

Cognitive Science Department

Rensselaer Polytechnic Institute

Troy, NY 12180, USA

phone: 518-276-3409

email: dr.ron.sun [AT] gmail.com

Sebastien Helie

Department of Psychological & Brain Sciences

University of California, Santa Barbara

Santa Barbara, CA 93106, USA

E-mail: sebastien.helie@psych.ucsb.edu

Running head: Psychologically realistic agents

Abstract

Cognitive architectures may serve as a good basis for building mind/brain-inspired, psychologically realistic cognitive agents for various applications that require or prefer human-like behavior and performance. This article explores a well-established cognitive architecture CLARION and shows how its behavior and performance capture human psychology at a detailed level. The model captures many psychological quasi-laws concerning categorization, induction, uncertain reasoning, decision-making, and so on, which indicates human-like characteristics beyond what other models have been shown capable of. Thus, CLARION constitutes an advance in developing more psychologically realistic cognitive agents.

Keywords: psychology, agent, cognitive architecture, CLARION.

Introduction

Cognitive architectures in general may constitute a solid basis for building psychologically realistic (in a sense, mind/brain-inspired) cognitive agents for various applications that require or prefer human-like behavior and performance. In the present article, we explore a well-established cognitive architecture, namely CLARION, to show how its behavior and performance mimics human psychology in a rather detailed way. The model captures, and provides explanations for, psychological 'quasi-laws" (i.e., robust statistical patterns and regularities) of categorization, uncertain reasoning, and decision-making (in addition to other laws in other areas as covered by other publications such as Helie and Sun, 2009), which indicates the human-like characteristics of the model. Thus, we argue that CLARION may serve as a basis for developing psychologically realistic cognitive agents (for various purposes) and constitutes an advance in this area.

"Psychological realism" as mentioned above is obviously important for developing psychologically-oriented cognitive models. Research in cognitive modeling explores the essence of cognition/psychology and various cognitive functionalities through developing detailed, process-based understanding by specifying models of mechanisms and processes. It embodies descriptions of cognition/psychology in computer algorithms and programs. That is, it produces runnable computational models. Detailed simulations are then conducted based on the computational models to validate the models.

Psychological realism, however, is also important for developing artificial cognitive agents for various application purposes. Its importance in this regard lies in the fact that they support the central goal of AI—building artificial systems that are as capable as human beings. Psychologically-oriented cognitive models help us to reverse engineer the only truly intelligent system around—the human mind/brain. They constitute a solid basis for building truly intelligent systems, because they are well motivated by, and properly grounded in, existing cognitive/psychological research. The use of such models in building artificial agents may also facilitate the interaction between humans and artificially intelligent agents because of the similarity of behavior between humans and psychologically realistic agents.

A necessary first step toward developing realistic psychological theories and models that can serve as basis for developing better, next-generation human-like intelligent systems (including afore-mentioned cognitive agents) is unification/integration of cognitive/psychological domains (Newell, 1990). Newell (1990) argued that a lot more data could be used to constraint a theory/model if the theory/model was designed to explain a wider range of phenomena. In particular, unified (integrative) psychological theories/models could be put to test against many well-known and stable regularities of human cognition that have been observed in psychology (i.e., psychological quasi-laws). So far, these integrative theories/models have taken the form of "cognitive architectures", and some of them have been successful in explaining a wide range of data (e.g., Anderson and Lebiere, 1998; Sun, 2002, 2004).

Specifically, a cognitive architecture is a broadly-scoped domain-generic computational cognitive model describing the essential structures and processes of the

mind used for multi-domain analysis of behavior (Sun, 2004). Newell (1990) proposed the Soar (Laird, Newell, and Rosenbloom, 1987) cognitive architecture as an example unified theory. Several other cognitive architectures have been proposed since then (e.g., Anderson and Lebiere, 1998; Meyer and Kieras, 1997; Sun, 2002, 2003). Among them is the CLARION cognitive architecture (Helie and Sun, 2010; Sun, 2002, 2003; Sun, Merrill, and Peterson, 2001; Sun, Slusarz, and Terry, 2005). CLARION assumes the distinction between explicit and implicit knowledge, as well as the distinction of action-centered and non-action-centered knowledge. Like a number of other cognitive architectures, CLARION is focused on the explanation of human behavior and has been successful in capturing a wide range of psychological data and phenomena (e.g., Helie and Sun, 2010; Sun, 2002; Sun et al., 2001, 2005; Sun and Zhang, 2006).

It is worth noting that cognitive architectures are the antithesis of "expert systems": Instead of focusing on capturing performance in narrow domains, they are aimed at providing a broad coverage of a wide variety of domains (Langley and Laird, 2003). Applications of intelligent systems (especially intelligent agents) increasingly require broadly scoped systems that are capable of a wide range of 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.

Cognitive architectures tend to be complex, including multiple modules and many free parameters (in their computational implementations). Because each module in a cognitive architecture usually includes its share of free parameters, increasing the number of modules in a cognitive architecture usually increases the number of free parameters in its computational implementation. Increasing the number of (independent) free parameters in a model adds to model complexity (Pitt and Myung, 2002; Roberts and Pashler, 2000). In cognitive architectures, this can result not only from the number of modules (as explained above), but also from the complexity of within-module processing. The number of free parameters within a module can be reduced by making the modules highly specialized, but this can only be achieved at the cost of adding more modules. Likewise, the number of modules can be reduced by making very general modules, which, however, usually have more free parameters.

Is it possible to create a cognitive architecture that can act as a *unified* theory and yet constrain the model complexity? The problem has been discussed in Sun (2004), which argued that a cognitive architecture should be *minimal*. In this paper, we study how the CLARION cognitive architecture can be minimal, exploring its core theory as the basis for building better cognitive agents.

The remainder of this paper is organized as follow. First, a general discussion of CLARION is presented. Second, the core theory of CLARION is expressed as a mathematical model. Third, a number of cognitive/psychological regularities are reviewed, along with mathematical and/or conceptual explanations of the phenomena using the core theory of CLARION (e.g., concerning categorization, uncertain reasoning, and decision-making). This presentation is followed by a general discussion.

Although many papers have been published on CLARION before, in this work, unlike previous work, we address accounting for generic psychological quasi-laws, rather than dealing with specific tasks and specific data sets. In other words, it is not about building specialized "expert systems" each of which can address a specific task, but is about building a general-purpose model that can simultaneously account for a variety of psychological regularities. This approach distinguishes this work from previous work.

Similarly, this work is not about new learning algorithms, or new computational techniques, but an architecture for structuring multiple algorithms, multiple representations, and so on, to achieve maximal effects (i.e., accounting for a maximal range of cognitive phenomena) with minimum mechanisms, as well as synergy from these mechanisms (as in human cognition).

The CLARION cognitive architecture

Overview

CLARION is, in part, based on two basic assumptions: representational differences and learning differences of two different types of knowledge: implicit versus explicit (Helie and Sun, 2010; Sun, 2002; Sun et al., 2001, 2005). These two types of knowledge differ in terms of accessibility and attentional requirement. The top level of CLARION (as in Figure 1) contains explicit knowledge (easily accessible, requiring more attentional resources) whereas the bottom level contains implicit knowledge (harder to access, more automatic). Because knowledge in the top and bottom levels is different, Sun et al. (2001, 2005) have shown that it is justified to integrate the results of top- and

bottom-level processing in order to capture the interaction of implicit and explicit processing in humans.

Insert Figure 1 about here

As can be seen in Figure 1, CLARION is divided into different subsystems, such as the *Action-Centered Subsystem*, the *Non-Action-Centered Subsystem*, as well as the *motivational* and *meta-cognitive* subsystems. The Action-Centered Subsystem (with both levels) contains procedural knowledge concerning actions and procedures (i.e., it serves as the procedural memory; Sun et al., 2005), while the Non-Action-Centered Subsystem (with both levels) contains declarative knowledge (i.e., non-action-centered knowledge; thus it serves as the declarative memory, both semantic and episodic; Helie and Sun, 2010). This division corresponds to the first assumption in CLARION.

With the Action-Centered Subsystem, CLARION captures, for example, sequence learning and sequential skills (Sun et al., 2001; Sun and Peterson, 1998). The Non-Action-Centered Subsystem of CLARION has been used to capture human reasoning (e.g., Sun and Zhang, 2006). For example, rule-based reasoning is captured by top-level processes (explicit) whereas similarity-based reasoning is captured through the interaction of top-level and bottom-level processes (implicit).

The second assumption in CLARION concerns the existence of different learning processes in the top and bottom levels, respectively (Sun et al., 2001, 2005). In the bottom level, implicit associations are learned through gradual trial-and-error learning. In contrast, learning of explicit rules in the top level is often “one-shot” and represents the abrupt availability of explicit knowledge following “explicitation” of implicit knowledge or new acquisition of linguistic (or otherwise explicit) information.

The following subsections present a more complete description of the subsystems. For clarity, only the details necessary to capture the psychological quasi-laws included in this paper are covered. So they provide a formal description of the non-action-centered subsystem, but only an abstract description of the other subsystems (however, more formal descriptions can be found in Sun, 2002, 2003, and Helie and Sun, 2010).

An abstract description of the Action-Centered Subsystem

The Action-Centered Subsystem (ACS) is the main subsystem in CLARION (Sun et al., 2001, 2005). In addition to being the long-term procedural memory, the ACS captures some executive functions (i.e., the control of some other subsystems). The ACS receives the inputs from the environment, and provides action recommendations.

In the top level of the ACS, explicit knowledge is represented using condition and action chunk nodes. In CLARION, each chunk represents a concept (for conditions or actions in the ACS). In the top level, each chunk is denoted by a chunk node (each chunk also has a distributed (micro)feature-based representation in the bottom level, as described next). Condition chunk nodes can be activated by the environment (e.g., a stimulus) or other CLARION components (e.g., working memory). Action chunk nodes can represent motor programs (i.e., a response) or queries/commands to other CLARION subsystems. Both condition and action chunks are individually represented by single nodes at the top level and have clear conceptual meaning (i.e., a localist representation). Chunk nodes can be linked to form rules (more later).

In particular, an action recommendation of the ACS can query the Non-Action-Centered Subsystem (for a cycle of reasoning; as detailed later). In this case, the Non-Action-Centered Subsystem can return one or several chunks resulting from reasoning

(which can be used in the ACS as action recommendations or as conditions for computation). Chunks returned by the Non-Action-Centered Subsystem are accompanied by their *internal confidence levels* (normalized activations). The internal confidence level is for estimating the confidence in the answer returned to the ACS. This measure is useful because the ACS does not have direct access to the processing that led to this chunk being returned. The ACS can use a threshold on the internal confidence level (ψ) to decide on accepting/rejecting the result of NACS processing. The internal confidence level estimates subjective confidence in a produced response.

The chunk nodes in the top level of the ACS are linked to represent rules of the form "Condition chunk node → Action chunk node". These rules can be simply represented by weights on the links, thus forming a linear connectionist network.[1] For example, a rule may be "if raining, then run". The concepts "raining" and "run" are represented by chunks. For "raining", there is a chunk node at the top level denoting it, and at the same time its (micro)features are represented at the bottom level (more later). The same goes for "run". There is a link from the chunk node for "raining" to the chunk node for "run" capturing the rule mentioned above.

These explicit procedural rules, and the chunk nodes involved, can be learned bottom-up (via the Rule-Extraction-Refinement algorithm; Sun et al., 2001), by explicit hypothesis testing (via the Independent Rule Learning algorithm), or be fixed (e.g., by experimental instructions; as described in Sun, 2003). In all cases, top-level rules are learned in a "one-shot" fashion. We will not get into details here.

The bottom level of the ACS uses (micro)feature-based representations to capture implicit procedural knowledge (note that we use "(micro)feature" to denote features that

may or may not be interpretable, as discussed in connectionist theorizing; sometimes we simply use "feature" instead of the more cumbersome "(micro)feature"). Each top-level chunk node is represented by a set of (micro)features in the bottom level (i.e., a distributed representation). The (micro)features (in the bottom level) are connected to the chunk nodes (in the top level) so that they can be activated together through button-up activation (when the features are activated first) or top-down activation (when the chunk nodes are activated first). Therefore, in general, a chunk is represented by both levels: using a chunk node at the top level and distributed feature representation at the bottom level; the distributed and localist representation together (along with their interaction) form a chunk.

The condition and action features are connected in the bottom level using several specialized multilayer nonlinear connectionist networks (MLPs). Each network can be thought of as a highly efficient behavior routine (once properly trained) that can be used to accomplish a particular type of task. Training of the bottom-level networks is iterative and done using, for example, backpropagation implementing Q-learning (Sun et al., 2001; Watkins, 1989).

A detailed description of the Non-Action-Centered Subsystem

The Non-Action-Centered Subsystem (NACS) of CLARION captures the declarative (both semantic and episodic) memory (Sun, 2002). The inputs and outputs of this subsystem usually come from the ACS. In addition, the NACS captures several forms of reasoning (Helie and Sun, 2010; Sun, 1994; Sun and Zhang, 2006). A technical description of the core processes of the NACS is provided below (the reader interested in the complete specification is referred to Sun, 2002, 2003).

Top level

In the top level of the NACS, explicit knowledge is represented by chunk nodes (as in the ACS top level). Unlike in the ACS, in the NACS, chunks are not divided into condition and action chunks: all chunks represent concepts that can be used either as a condition or a conclusion in rule application. Each chunk node at the top level can be activated by: (1) an ACS query, (2) its association with another chunk (via an associative rule), or (3) its similarity to another chunk (via similarity matching). When a NACS chunk node is activated by an ACS query, the information is transmitted exactly (Helie & Sun, 2009), so the activation of the NACS chunk is set to full activation (i.e., $s_j^{ACS} = 1$). However, the other two sources of activation can have smaller (positive) values.

NACS chunk nodes can be linked together to represent 'associative' rules (similar to a semantic network in a way; Quillian, 1968). In the simplest case, by representing the associative rules using connection weights, the top level of the NACS can be represented by a linear connectionist network:

$$s_j^r = \sum_i s_i \times w_{ij}^r \tag{1}$$

where s_j^r is the activation of chunk node j following the application of an associative rule, s_i is the activation of chunk node i, and w_{ij}^r is the strength of the associative rule between chunk nodes i and j (by default, $w_{ij}^r = 1/n$, where n is the number of chunk nodes in the condition of the associative rule).[2] The application of Eq. 1 is referred to as *rule-based reasoning* (Sun, 1994).

NACS chunks also share a relationship through similarity, which enables reasoning by similarity. In CLARION, the activation of a chunk caused by its similarity to other chunks is termed *similarity-based reasoning*. Specifically,

$$s_j^s = s_{c_i \sim c_j} \times s_i \tag{2}$$

where s_j^s is the activation of chunk node j caused by its similarity to other chunks, $s_{ci\sim cj}$ is the similarity from chunk i to chunk j, and s_i is the activation of chunk node i. The similarity measure ($s_{ci\sim cj}$) is carried out through the interaction of the bottom level and the top level of the NACS and is detailed in the following subsection (see Eq. 6 below).

Overall, the activation of each chunk node in the top level of the NACS is equal to the maximum activation it receives from the three previously mentioned sources, i.e.:

$$s_j = Max(s_j^{ACS}, s_j^r, s_j^s) \tag{3}$$

where s_j is the overall activation of chunk node j.

Chunks that are inferred (activated) in the NACS may be sent to the ACS for consideration in action decision-making (via chunk nodes). Every chunk that is sent back to the ACS is accompanied by an internal confidence level. The internal confidence level is calculated by transforming the chunk activations (Eq. 3) into retrieval probabilities (using a Boltzmann distribution, which is common in psychology):

$$P(j) = \frac{e^{s_j/\alpha}}{\sum_i e^{s_i/\alpha}} \tag{4}$$

where $P(j)$ is the probability that chunk j is chosen to be returned to the ACS, s_j is the activation of chunk node j (Eq. 3), and α is a free parameter representing the degree of randomness (temperature). This normalized activation is used as the internal confidence level. If only one chunk is to be returned to the ACS, a chunk is stochastically selected using Eq. 4.

In addition to the above-mentioned activation, each chunk node has a base-level activation, for representing priming effects, defined as:

$$b_j^c = ib_j^c + c\sum_{l=1}^{n} t_l^{-d} \tag{5}$$

where b_j^c is the base-level activation of chunk node j, ib_j^c is the initial base-level activation (by default, $ib_j^c = 0$), c is the amplitude (by default, $c = 2$), d is the decay rate (by default, $d = 0.5$), and t_l is the lth use/coding of the chunk node. This measure decays exponentially and corresponds to the odds of needing chunk node j based on past experiences (see Anderson, 1990 for further justifications). When the base-level activation of a chunk node falls below a "density" parameter (d_c), the chunk is no longer available for reasoning (rule-based or similarity-based). In the NACS, base-level activations are used for capturing forgetting (with the density parameter) and for computing the retrieval time of a chunk (whereby retrieval time is inversely proportional to base-level activation).[3]

Like in the ACS, chunk nodes in the NACS can be learned by explicitly encoding given information (using, e.g., "fixed rules"; Sun, 2003) and by acquiring explicit knowledge bottom-up from the bottom levels of CLARION (both from the ACS and the NACS; e.g., by using the Rule-Extraction-Refinement algorithm; see Sun et al., 2001). In addition, each item experienced by the agent has a probability p of being encoded in the NACS as a (semantic and/or episodic) chunk at every time step.

Bottom level

As in the ACS, the bottom level of the NACS uses (micro)feature-based representations to encode the chunk (which is also indicated by corresponding top-level chunk nodes) with distributed representations (Helie and Sun, 2010). The (micro)features are connected to the top-level chunk nodes so that, when a chunk node is activated, its

corresponding bottom-level feature-based representation (if exists) is also activated and vice versa.

The connections between top-level chunk nodes and their corresponding bottom-level (micro)feature-based representations allow for a natural computation of similarity (Eq. 2):

$$s_{c_i \sim c_j} = \frac{n_{c_i \cap c_j}}{f\left(n_{c_j}\right)} = \frac{\sum_k w_k^{c_j} h_k\left(c_i, c_j\right)}{f\left(\sum_k w_k^{c_j}\right)} \tag{6}$$

where w_k^{cj} is the weight of feature k in chunk j (by default, $w_k^{cj} = 1$ for all ks), $h_k(c_i, c_j) = 1$ if chunks i and j share feature k and 0 otherwise, and $f(\bullet)$ is a slightly supralinear, positive, monotonically increasing function [by default, $f(x) = x^{1.1}$]. This is because similarity-based activation is not exact, and thus should not fully activate a chunk (therefore rules can always be used to cancel the result of similarity-based reasoning; see the *Uncertain Reasoning* section below). By default, $n_{ci \cap cj}$ counts the number of features shared by chunks i and j (i.e., the feature overlap) and n_{cj} counts the number of features in chunk j. However, the feature weights can be varied to account for prior knowledge or the context. Thus, similarity-based reasoning in CLARION is naturally accomplished using (1) top-down activation by chunk nodes of their corresponding feature-based representations, (2) calculation of feature overlap between any two chunks (as in Eq. 6), and (3) bottom-up activation of the top-level chunk nodes (Eq. 2).

Here is a simple example of how Eq. 6 may be used. Consider the two chunks ‘fire engine’ (with features ‘red’, ‘has wheels’, and ‘has a ladder’) and ‘tomato’ (with

features ‘red’, ’grows on trees’, and ‘edible’). We may calculate the similarity-based activation of ‘tomato’ from ‘fire engine’ as follows. First, for simplicity, assume that all three features of ‘tomato’ are equally important (i.e., $w_k^{cj} = 1$ for all k’s). So, the denominator of Eq. 6 is $3^{1.1} \approx 3.35$. Next. $h_k($‘fire engine’, ‘tomato’$)$ is calculated for each feature of ‘tomato’ and multiplied by the corresponding weight (all weights were assumed to be one). Only the first feature of ‘tomato’ is shared by ‘fire engine’ (i.e., ‘red’). Thus the numerator of Eq. 6 is 1.So, Eq. 6 produces 1 / 3.35 ≈ 0.30, reflecting the fact that that fire engines and tomatoes are not highly similar. The result of Eq. 6 is then multiplied by the original activation of the chunk node ‘fire engine’ (as stated by Eq. 2), which produces the similarity-based activation of the chunk node ‘tomato’.

One interesting application of similarity-based reasoning is based on the ‘reverse containment principle’. According to the reverse containment principle, if chunk i represents a category that is a superset of the category represented by chunk j, all the (bottom-level) features of chunk i are included in the (bottom-level) features of chunk j (i.e., $n_{ci \cap cj} = n_{ci}$). For instance, chunk i could represent the category ‘bird’ while chunk j could represent the category ‘sparrow’. In this case, the feature-based description of ‘sparrow’ would include the feature-based description of ‘bird’ (plus additional features unique to sparrows). The reverse containment principle allows for a natural explanation of inheritance-based inference (more later).

In addition to enabling the calculation of the similarity measure, the bottom level of the NACS captures implicit non-action-centered processing. Implicit processing is accomplished by using a number of networks connecting distributed (micro)feature-based representations (as in the bottom level of the ACS). Some of these networks are auto-

associative, allowing retrieval using partial match, while some others are hetero-associative, allowing retrieval of input-output mappings (more details later).

Insert Figure 2 about here

Auto-associative networks can be implemented using Hopfield-type networks (Anderson, Silverstein, Ritz, and Jones, 1977; Hopfield, 1982). In particular, the *Nonlinear Dynamic Recurrent Associative Memory* (NDRAM; Chartier and Proulx, 2005) has been used in the NACS of CLARION (see, e.g., Hélie and Sun, 2010). The NDRAM (as used in the bottom level of the NACS) is a synchronous Hopfield-type model that allows for learning continuous-valued patterns and minimizes the number of spurious memories in the model (Chartier and Proulx, 2005). The network is shown in Figure 2. The transmission within the network is as follows:

$$x_{i[t+1]} = g\left(\sum_{j=1}^{N} w_{ij} x_{j[t]}\right) \tag{7}$$

$$g(y) = \begin{cases} +1 & , \text{ if } \; y > 1 \\ (\delta+1)y - \delta y^3 & , \text{ if } \; -1 \le y \le 1 \\ -1 & , \text{ if } \; y < -1 \end{cases} \tag{8}$$

where N is the number of nodes in the network, $x_{i[t]}$ is the state of the ith node in the network at time t, $\delta > 0$ is a free parameter representing the slope of the transmission function (by default, $\delta = 0.4$), and $\mathbf{W} = [w_{ij}]$ is the $N \times N$ weight matrix, which may be learned online using:

$$w_{ij[k+1]} = \zeta w_{ij[k]} + \eta\left(\bar{x}_i \bar{x}_j - x_{i[p]} x_{j[p]}\right) \tag{9}$$

where $w_{ij[k]}$ is the connection weight between nodes i and j at time k ($w_{ij[0]} = 0$), $x_{i[p]}$ is the activation of node i after p applications of Eqs. 7 and 8 (by default, $p = 1$), η is the

learning rate (by default, $\eta = 0.001$), ζ is a memory efficiency parameter (by default, ζ = 0.9999), and $\bar{x}_i$ is the output of the vigilance module:

$$\bar{x}_i = z\frac{(\mu x_{i[0]} + x_{i[p]})}{1+\mu} + (1-z)x_{i[0]} \tag{10}$$

where $x_{i[0]}$ is the initial activation of node i (before the p applications of Eqs. 7 and 8), μ is a free parameter that quantifies the effect of the initial activation in $\bar{x}_i$ (by default, μ = 0.01), and z is defined by:

$$z = \begin{cases} 1, \text{ if } \sum_{i=1}^{N} x_{i[0]} x_{i[p]} \left(\sum_{j=1}^{N} x_{i[0]}^2 \sum_{j=1}^{N} x_{i[p]}^2 \right)^{-1/2} > \rho \\ 0, \text{ Otherwise} \end{cases} \tag{11}$$

where $0 \leq \rho \leq 1$ is the vigilance parameter (Grossberg, 1976). In words, z = 1 if the correlation between the initial activation and the final activation is higher than ρ and zero otherwise. Hence, the initial activation is learned when the correlation between the initial and final activations (as determined by the *if* clause in Eq. 11) is low (which suggests a new activation pattern), and a weighed average between the initial activation and the final activation is learned when the correlation is high (which suggests a variant of an already learned activation pattern).[4] It should be noted that learning is online; that is, learning occurs each time a stimulus is presented to the model (Chartier and Proulx, 2005).[5]

The weight space of Hopfield-type neural networks can be interpreted as a phase space in a dynamical system (e.g., Anderson et al., 1977; Haykin, 2009). The learned patterns are represented by fixed-point attractors, and all the trajectories (initiated by activation patterns) have been shown to converge to a fixed-point attractor in NDRAM (Hélie, 2008).[6] This interpretation of Hopfield-type neural networks can be used to draw

a useful analogy between the phase space in the system and psychological spaces (Shepard, 1987; Tenenbaum and Griffiths, 2001): Concepts/categories in psychological space are represented by fixed-point attractors in the phase space, and new stimuli can be learned (i.e., added to the psychological space, which corresponds to creating a new fixed-point attractor in the phase space), and existing items can be retrieved from the psychological space (i.e., through convergence of the transmission to an already existing fixed-point attractor in the phase space).

A quick sketch of the motivational and meta-cognitive subsystems

Supervisory processes over the operations of the ACS and the NACS are based on two subsystems in CLARION: the motivational subsystem and the meta-cognitive subsystem (see Figure 1).

The motivational subsystem (the MS) is concerned with "drives" behind actions (Toates, 1986). That is, it is concerned with why an agent does what it does. Drives help to determine explicit, specific goals as well as reinforcements (rewards) for the ACS specifically and thus, indirectly, the actions of the agent (Sun, 2002, 2003).

The meta-cognitive subsystem (the MCS) monitors, controls, and regulates cognitive processes for the sake of improving cognitive performance in achieving goals (Ackerman and Kanfer, 2004; Reder, 1996) on the basis of motivational states (from the MS). Control and regulation may be carried out through setting reinforcement (reward) functions for the ACS on the basis of drive activations (Sun, 2009). Control and regulation may also be in the forms of setting goals for the ACS, setting essential parameters of the ACS and the NACS, interrupting and changing on-going processes in the ACS and the NACS, and so on. Since these two subsystems are not essential to the

exposition below, we will not go into further details of them here (but see Sun, 2009 for full details).

Capturing Human Cognition: Accounting for Psychological Quasi-Law

In this section, we show how CLARION captures and explains some basic psychological quasi-laws (statistical patterns and regularities). In each subsection, several quasi-laws within a particular psychological domain are explained. For many additional psychological laws or quasi-laws captured by CLARION, not explained here, see Helie and Sun (2009).

It should be noted that the goal here is not to simulate specific psychological data; instead, it is to show that the core CLARION theory can account for these psychological quasi-laws, as intrinsic to the theory, and most of the following explanations are parameter-free. In this way, CLARION is shown to be a psychologically realistic model of cognitive agents.

Categorization

Categorization is one of the most important cognitive functions (Harnad, 2005; Kruschke, 2008). Major psychological phenomena concerning categorization include similarity effects, frequency effects, and size effects (see, e.g., Kruschke, 2008). These phenomena are captured within the NACS of CLARION. Note that only one free parameter is varied (i.e., ρ, the vigilance parameter).

Similarity in categorization

Feature similarity. Human experiments in psychology have shown that the similarity between two stimuli is affected by the number of matching features, as well as by the number of non-matching features (as argued, for example, by Tversky, 1977).

CLARION captures this. In CLARION (in the NACS in particular), feature-based similarity is mostly computed through the interaction of the bottom level and the top level (Eq. 6). The effect of matching features is represented by the numerator in similarity calculation, while the effect of non-matching features is represented in the denominator:

$$s_{c_i \sim c_j} = \frac{n_{c_i \cap c_j}}{f\left(n_{c_j}\right)} = \frac{n_{c_i \cap c_j}}{f\left(n_{c_j \cap c_i} + n_{c_j \cap \neg c_i}\right)}$$

where n_{cj} is the number of features representing chunk j, $n_{ci \cap cj}$ is the size of the feature overlap of chunks i and j, $n_{cj \cap \neg ci}$ is the number of features in chunk j that are not part of chunk i, and $f(\bullet)$ is a monotonically increasing, slightly supralinear, positive function. As can be seen, the numerator increases with the number of matching features, which leads to higher similarity in similarity calculation. The denominator increases with the number of non-matching features, which reduces the similarity. This captures qualitatively the effect found in the empirical data as mentioned earlier (e.g., Tversky, 1977).

Asymmetry of similarity. Human experiments have shown that subjective similarity is not always symmetric, and in fact, often asymmetric (see Kruschke, 2008; Tversky, 1977).

This empirical result is readily accounted for by the similarity measure (as used in CLARION):

$$s_{c_i \sim c_j} = \frac{n_{c_i \cap c_j}}{f\left(n_{c_j}\right)}$$

$$s_{c_j \sim c_i} = \frac{n_{c_j \cap c_i}}{f\left(n_{c_i}\right)}$$

Here, the numerators are the same (assuming that the feature weights are the same, $n_{ci\cap cj} = n_{cj\cap ci}$; see Eq. 6) but the denominators differ (unless $n_{ci} = n_{cj}$). Hence, subjective similarity is symmetric if and only if the two stimuli have the same number of features and the same feature weights; otherwise, similarity is asymmetric in CLARION, as in empirical data from human experiments.

Frequency effects

Reliability of categorization. Human experiments have shown that stimuli that are more frequent are easier to categorize correctly; see, for example, Nosofsky (1988) for such experimental demonstrations.

This can be accounted for, again, by the NACS in CLARION, which is in part implemented by an attractor neural network (in the bottom level, as described earlier). The attractors represent category prototypes that have been learned previously using exemplars (previous stimuli). It was shown that the position (in the phase space) of the attractors learned by the model converged toward the arithmetic mean of the stimuli used to train an attractor network (Hélie, Chartier, and Proulx, 2006). The stimuli that appeared more frequent (during learning) were learned more often and thus weighed more heavily in the determination of the position of the attractor (prototype). Hence, the

stimuli that are more frequent are closer (i.e., more similar) to their category prototype, that is, the attractor and thus more easily settled into the attractor. Therefore, they are easier to categorize correctly.

Fan effect. Features that are consistently associated with a category facilitate categorical decisions, as demonstrated by, for example, Reder and Ross (1983).

In CLARION, the fan effect in categorization has a similar explanation as the *reliability of categorization* effect. In the NACS, the features that are more frequently associated with a category affect more the position of the attractor that represents the category prototype, because they are learned more often and are thus weighed more heavily in determining the attractor position. Hence, features that are associated more often with a category during training facilitate categorization in a test phase when they are present during the test phase, because they make the test stimulus closer to the attractor representing the prototype.

Size of categories effect

Base rate effects. Generally speaking, human subjects are more likely to assign a new stimulus to larger existing categories (i.e., those with more exemplars; see, e.g., Homa and Vosburgh, 1976).

This characteristic of human categorization may be explained by the "density estimation" capabilities (Hélie et al., 2006) of the NACS of CLARION. In CLARION, the categories are represented by attractors in the bottom level (as well as chunk nodes at the top level), and more frequent categories (encountered during training) have larger attractor fields. Hence, new stimuli are more likely to be categorized in those more frequent categories because they are more likely to initiate trajectories in the attractor

fields of the attractors representing more frequent categories (for a formal explanation, see Anderson et al., 1977).

Variability. Psychological experiments have shown that human subjects are more likely to categorize a new stimulus in a category with higher variance among its existing members (as demonstrated by, e.g., Nisbett, Krantz, Jepson, and Kunda, 1983).

In CLARION, this result can be accounted for by the vigilance parameter in the attractor neural network in the NACS (see, e.g., Grossberg, 1976). The variability of the categories may be captured using parameter ρ, which represents the minimal correlation required between the stimulus and an existing attractor for them to be categorized together. Setting $\rho = 1$ creates extremely small categories (each stimulus is considered a category prototype) while setting $\rho = 0$ results in all stimuli being put into the same category. Categories are represented by attractors in the bottom level (and chunk nodes at the top level) and each attractor may have a different value for ρ: More variable categories have a lower vigilance value, whereas less variable categories have a higher vigilance value. This would correspond to what has been observed in human data, where more variable categories are more likely to be used to categorize new stimuli. In CLARION, the vigilance parameters may be learned during category learning with reinforcement learning algorithms.

Uncertain Reasoning

Reasoning is an important cognitive capability that allows the generation of predictions based on observations (e.g., induction) and also the application of general rules to particular cases (i.e., deduction). Research has shown that human reasoning is often a mixture of rule-based and similarity-based processes (for reviews and arguments,

see Sun, 1994). Under many circumstances, human reasoning is uncertain (i.e., not known for sure and not guaranteed to be correct but plausible). Several cases of human uncertain reasoning have been identified in Sun (1994). The core theory of CLARION can capture them (with the NACS of CLARION).

The present subsection presents a conceptual description of the cases and their explanations in CLARION. Detailed mathematical proofs can be found in Helie and Sun (2009). Note that all the explanations and derivations below are parameter free.

Uncertain information

For humans, when information regarding the premise of a rule is not known with certainty, a conclusion can still be reached albeit with a certain amount of uncertainty (Collins and Michalski, 1989; Sun, 1994).

In CLARION, this phenomenon can be explained by rule-based reasoning within the NACS (see Eq. 1). Uncertainty of information is captured by partial activation (< 1; as opposed to full activation). In CLARION, if the premise chunk node is partially activated, the conclusion chunk node is also partially activated, as indicated by Eq. 1, proportional to the activation of the premise chunk node.

Incomplete information

When a rule has more than one premises, a conclusion can be reached (with uncertainty) even if only some of the premises are known (Sun, 1994). For example, one could have a rule: “If Peyton Manning and Reggie Wayne play, the Colts are going to win the game”. If it is known that Peyton Manning plays but the playing status of Reggie

Wayne is unknown, the conclusion that the Colts is going to win can still be made (with some uncertainty).

In CLARION, this phenomenon is accounted for by rule-based reasoning in the NACS (see Eq. 1). Each premise in a rule has a weight, and the weights of the premises add to one. Hence, when not all the premise chunk nodes are activated, the conclusion chunk node is partially activated, proportional to the number of activated premise chunk nodes. Therefore, a partially certain conclusion may be reached within CLARION.

Similarity matching

When no rule allows for answering a question directly, one can make an inference based on similarity to other known facts (Sun, 1994). For example, when asked: "Is the Chaco a cattle country?", one answered: "It is like western Texas, so in some sense I guess it's a cattle country" (Collins and Michalski, 1989; Sun, 1994). That is, an answer may be based on similarity matching.

In CLARION, this phenomenon is explained by similarity-based reasoning (through the interaction of the bottom level and the top level; see Eq. 2). When two chunks share a subset of features, the activation of one chunk is partially transmitted to the other. Here, the activation of a chunk node representing "Chaco" activates (partially) the chunk node representing "western Texas" (through top-down and bottom-up activation flow, which completes similarity calculation), which in turn (partially) activates all the rules associated with western Texas (e.g., "western Texas is a cattle country"). Hence, activating the chunk node representing "Chaco" automatically activates (partially) the chunk node representing "cattle country", proportional to the similarity between the two.

Superclass-to-subclass inheritance

In inheritance, one uses properties from the superclass to answer a question about a subclass (for human data, see Collins and Quillian 1969; Sun, 1994). For example, when asked if Snoopy the dog has four legs, one may respond "yes" because the generic (prototypical) dog has four legs.

In CLARION, inheritance is captured as a special case of similarity-based reasoning, because the bottom-level feature representations along with the inter-level interaction in CLARION can capture a categorical hierarchy (as discussed before, and as shown in Sun, 1994, 2003). Chunks representing subclasses (subcategories; e.g., Snoopy) have all the features of the chunk representing their superclass (e.g., dogs), plus additional features making them unique (i.e., the reverse containment principle; see *A detailed description of the Non-Action-Centered Subsystem*). Hence, superclass-to-subclass inheritance is naturally explained in CLARION by similarity-based reasoning applied to superclass-subclass relations (Eq. 6).[7]

Cancellation of superclass-to-subclass inheritance

According to CLARION, superclass-to-subclass inheritance is the most reliable (certain) form of similarity-based reasoning (Eq. 6). However, it is not as reliable (certain) as rule-based reasoning (Eq. 1). Hence, known rules (at the top level) can be used to cancel such inheritance, thus capturing exceptions. For instance, one can infer that Snoopy the dog has four legs, because the prototypical (generic) dog has four legs (i.e., superclass-to-subclass inheritance). However, a rule might also be present (at the top level), stating that Snoopy the dog does not have four legs, because he was in an accident (cancellation of inheritance).

In CLARION, similarity-matching alone cannot fully activate a chunk node. This is because the denominator in Eq. 6 is supralinear, as explained earlier. In contrast, rule-based reasoning can fully activate a chunk node (Eq. 1). Hence, rules can be used to cancel conclusions reached by similarity-based reasoning, thus leading to canceling inheritance.

The reverse inheritance (from subclass to superclass) as well as its cancellation can be explained in a similar fashion (see Sun, 1994).

Mixed rules and similarities

In addition, rule-based and similarity-based reasoning can be chained in many different ways. For instance, as explained before, a chunk node can be activated by similarity matching, and the newly inferred chunk node can fire a rule. The opposite can also happen: inferring a chunk node using rule-based reasoning and activating another chunk node by similarity to the inferred chunk node. There are many such cases that can be explained by chaining the explanations of the preceding cases. Different mixtures of rule-based and similarity-based reasoning are explained in Sun (1994, 2003) and Helie and Sun (2009).

Decision-making

Decision-making is concerned with choices and preferences, as studied in psychology and economics (e.g., Thurstone, 1927). When only two choices are available, important cognitive phenomena as documented in the psychological literature include: strong stochastic transitivity, independence of irrelevant alternatives, and regularity in binary choices (e.g., as reviewed in Busemeyer and Diederich, 2002). When more than

two choices are available, cognitive phenomena observed include: the similarity effect, the attraction effect, the compromise effect, and the complex interactions between these effects (e.g., as reviewed in Roe, Busemeyer, and Townsend, 2001).

Several psychological models of decision-making have been proposed (e.g., elimination by aspect, Thurstone preferential model, additive utility models, etc; for a review, see Busemeyer and Johnson, 2008). They cannot account for all the afore-mentioned effects simultaneously. The only exception is *decision field theory* (DFT), which can account simultaneously for all the afore-mentioned phenomena, is amenable to analytical solution, and can be captured by a connectionist model (Busemeyer and Johnson, 2008; Helie and Sun, 2009). CLARION embodies such a model in its NACS.

Capturing and enhancing Decision Field Theory within CLARION

The role of decision field theory (DFT) in the CLARION cognitive architecture is examined below. Those not familiar with DFT and/or its terminology are referred to Appendix for a brief description.

First, complex decision-making (such as as captured in DFT) is carried out in the CLARION NACS (Sun, 2002). As explained earlier, the bottom level of the NACS is composed of several specialized networks that can be either auto-associative (e.g., NDRAM: Chartier and Proulx, 2005) or hetero-associative (e.g., backpropagation networks or MLPs: Haykin, 2009). A special module in the bottom level of the NACS of CLARION, devoted to decision-making, includes a hetero-associative connectionist network implementing DFT.[8] Only one free parameter is varied to explain the decision-making phenomena (i.e., ψ, the threshold on the internal confidence level). Each option from the DFT network is also (redundantly) represented as a chunk node in the top level

of the NACS, and the activations of the option chunk nodes are the same as the outputs from the fourth layer of the DFT network (detailed next).

Connectionist implementation of Decision Field Theory

A connectionist network implementing decision field theory (DFT) within the bottom-level NACS of CLARION is presented in Figure 3. The matrix formulation of DFT (as explained in the Appendix) allows for a natural implementation in a four-layer connectionist network.

In connectionist terminology, the evaluation of each option attribute (as captured by the **M** matrix) represents the stimulus (i.e., the input pattern), and attention allocation [the $\mathbf{w}(t)$ vector] is used to filter the input so that only attended dimensions reach the second layer of the network. The contrast matrix (**C**) is captured by the weight connections between the second and third layers of the network. The activation in the third layer represents the *valence* of each option (e.g., the momentary advantage/disadvantage of an option in relation to the other options). The fourth layer represents the network dynamics (with the **S** matrix; i.e., the trajectory of the decision process). Finally, the output activation of the fourth layer represents the *preference*. The connections between the third and fourth layers are direct and one-to-one. [The activations of the option chunks at the top level are the same as the fourth layer. A more complete description of the network can be found in Roe et al. (2001).]

Insert Figure 3 about here

Decision process. First, the ACS sends a request to the NACS for considering a particular decision. This request activates the option attribute nodes (the first layer of the DFT network in the bottom level) to represent the personal evaluation of these attributes.

This activation is then propagated throughout the bottom-level network (as described in the Appendix). The output of the last layer of the bottom-level DFT network (i.e., the preferences) are transferred to the top-level chunk nodes representing the options (and the chunk nodes activations are equal to the fourth-layer preferences).

The chunk nodes in the top level representing the options are retrieved by the ACS, and their activations lead to the internal confidence level. The threshold used in the ACS on the internal confidence level is equivalent to the upper boundary in the DFT diffusion process (i.e., it controls when a decision is output). When the internal confidence level of one of the options crosses this boundary, a decision is made. Otherwise, the decision process continues for another iteration in the NACS.[9]

Advantages. The DFT network in the bottom level of the NACS enhances both the decision-making capabilities of CLARION and the range of the phenomena that DFT can account for. By capturing the duality and the complex interaction of explicit and implicit processes, CLARION adds new dimensions to DFT.

First, top-level chunk nodes in CLARION may be connected to other chunk nodes, enabling rule-based reasoning and similarity-based reasoning to be carried out (see the *Uncertain reasoning* subsection), which could not have been carried out within DFT alone.

Second, the rules in the top level can also be used to validate the option chosen by the DFT network. The validation rules can include, for example, moral imperatives, cultural conventions, behavioral norms, and so on, because top-level rules can take precedence over bottom-level activation/recommendation. These rules may not have been internalized sufficiently to be reflected in the valences in the bottom-level DFT network

(see, e.g., Sun, Zhang, and Mathews, 2009 and Sun, 2003 regarding internalization through top-down learning).

Third, one consequence of the presence of rule-based reasoning in CLARION is that options or features can be eliminated from consideration. If some features are added/eliminated by rule-based reasoning (from the ACS or the MCS), the inhibition matrix is automatically redefined because changing the features considered changes the similarity relations between the chunk nodes (options) in CLARION (see Eq. 6). When the set of features or options is modified, the diffusion process in the DFT network is re-initialized.

Fourth, similarity-based reasoning in CLARION also plays an important role in decision-making. Specifically, the similarity between chunk nodes representing options in the top level of CLARION defines the inhibition matrix of DFT:

$$\begin{aligned} \forall_{i \neq j},\ s_{ij} &= -\left(s_{c_i \sim c_j}\right) \\ &= -\frac{n_{c_i \cap c_j}}{f(n_{c_j})} \end{aligned} \tag{12}$$

where $\mathbf{S} = [s_{ij}]$ is the inhibition matrix and $s_{ci\sim cj}$ is the similarity between chunks i and j (the diagonal terms of the $\mathbf{S}$ matrix is set to one[10]). This definition corresponds to the constraint on the inhibition matrix as defined in DFT (i.e., that inhibition is negatively related to similarity; Busemeyer and Diederich, 2002; Roe et al., 2001). This definition of the inhibition matrix eliminates all the free parameters in DFT: Similarity in CLARION is defined by the overlap of the chunk features, which is sufficient to define the inhibition matrix in DFT.

Fifth, when the ACS queries the NACS, it can specify the set of attributes and options to be considered in the decision. It can also provide a dynamical model of

attention switching and focus the decision on one or several attributes of the options (e.g., by changing the focus on every iteration, as humans often do; see, e.g., Bundesen, Habekost, Kyllingsbæk, 2005). This can account for the attention selection process in DFT.

Finally, the initial state of the diffusion process [initial preferences; $\mathbf{p}(0)$] can be initialized by the ACS, for example, by using the base-level activations of the chunk nodes representing the options (Eq. 5), which can represent decision biases based on previous choices made in similar situations. Alternatively, an effort to be objective can be made by the decision-maker and the diffusion process can be unbiased by setting $\mathbf{p}(0) = \mathbf{0}$ (and ignoring the base-level activations; through ACS actions).

Accounting for psychological phenomena

In Busemeyer and Johnson (2008), it was shown that DFT can account simultaneously for the following phenomena: violation of independence, stochastic dominance, preference reversals, and context dependent preferences. DFT can also account for: stochastic transitivity, speed-accuracy trade-offs, preference reversal under time pressure, and decision times (Busemeyer and Diederich, 2002). In multi-alternative choices, DFT can account for similarity effects, the attraction effect, the compromise effect, and the complex interaction between these phenomena (Roe et al., 2001).

Because CLARION includes a DFT network in the bottom level of the NACS, it can also account for all the empirical phenomena above. Detailed explanations for the empirical phenomena above are essentially the same as in DFT, so they are not repeated here (the reader is referred to the cited papers on DFT for details).

In this work, we have shown how DFT can be enhanced by its inclusion in the bottom level of the NACS of CLARION. Inclusion in CLARION eliminates all the free parameters in DFT by defining the inhibition matrix using similarity-based reasoning in CLARION and the random-walk boundary using the ACS threshold on internal confidence levels of CLARION. Furthermore, rule-based reasoning in CLARION can be used to select options and attributes, including the possibility of an additional validation process. So CLARION can also account for these additional phenomena. In particular, CLARION can account for more complex sociocultural decision-making situations by human subjects (Sun, 2006), for example, by the use of explicit cultural rules within the subject.

General Discussions

In this article, we utilized the core theory of the CLARION cognitive architecture to explain data in multiple psychological domains, such as categorization, uncertain reasoning, and decision-making. In particular, this core CLARION theory was able to provide rigorous almost parameter-free explanations.[11] In this article, we review a range of psychological phenomena, and present mathematical/conceptual explanations of the phenomena using the CLARION NACS. The list of phenomena is not exhaustive, because the phenomena were selected based on their historical importance (e.g., as evidenced by inclusion in major reviews of the fields) and empirical reproducibility.

Because we believe that the interaction between explicit and implicit declarative memories is essential to the explanation of these psychological phenomena, the CLARION cognitive architecture is a natural candidate architectural theory. First, the interaction between explicit and implicit processing is one of the main principles

underlying the development of this cognitive architecture (Helie & Sun, 2010; Sun, 2002; Sun et al., 2005). Second, the interaction between explicit and implicit processing in CLARION has already been shown to yield synergistic learning and performance (Helie & Sun, 2010; Sun, 2002; Sun et al., 2005). Third, the interaction between explicit and implicit processing in the NACS has been shown to be essential to model similarity effects in human reasoning (Sun & Zhang, 2006) and decision-making (Helie & Sun, in press). For all these reasons, the CLARION cognitive architecture was used to explore the interaction of explicit and implicit knowledge in many psychological domains.

Note that the CLARION cognitive architecture is not about computational techniques (although there have been many innovations in this regard, as published by the authors in the late 1990's and early 2000's); the focus here is on providing a theoretical contribution to integration. CLARION is promising as a psychologically realistic model for cognitive agents.

One possible objection would be that a cognitive architecture necessarily involves rather complex interactions among components and therefore properties of one component (such as being able to account for a psychological quasi-law) may not hold after the interactions are taken into consideration. To address this objection, we should note the fact that psychological phenomena (psychological quasi-laws) are known to vary with contexts----prior experiences, individual circumstances, environmental conditions, instructions, task demands, other individuals, and so on. Although some psychological phenomena are more generic and constant, all are subject to the influences of contexts. One can only identify regularities within certain contexts (generic or specific), and account for them within the same contexts. From this perspective, CLARION is indeed

capable of accounting for these quasi-laws despite the existence of interactions among components. Given the context for any of these quasi-laws, the interactions within CLARION would be limited and fully identifiable, and therefore can be taken into consideration when accounting for the phenomenon.

This work is not about throwing together a large set of small computational models so that the resulting system can do all of whatever each of these models are capable of, which seems rather pointless. On the contrary, a major point of the work on CLARION has been about selectively including a minimum set of mechanisms, structured in a parsimonious but effective way, to account for a maximum set of psychological data and phenomena. The key focus here lies in: (1) minimal mechanisms, (2) maximal scope and coverage, and (3) effective integration (which may lead to synergy effects of various forms, such as those described by Sun et al., 2005 and Sun and Zhang, 2006). Accounting for general psychological quasi-laws serves to demonstrate this point.

It should be mentioned that, in the past, CLARION has been successful in simulating, accounting for, and explaining a wide variety of specific 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. Among them, some tasks are typical implicit learning tasks (mainly involving implicit reactive routines), while some others are high-level cognitive skill acquisition tasks (with significant presence of explicit processes). In addition, extensive work has been done modeling a complex and realistic minefield navigation task, which involves complex sequential decision-making (Sun et al., 2001). We have also worked on

reasoning tasks, creative problem solving tasks, social simulation tasks, as well as meta-cognitive and motivational tasks. While accounting for various psychological data, CLARION provides explanations that shed new light on cognitive processes (see, e.g., Helie and Sun, 2010; Sun et al., 2001; Sun et al., 2005; Sun, 2002; Sun and Zhang, 2006).

We may compare the work on the CLARION cognitive architecture with other general approaches towards psychological realism. For instance, Soar is a symbolic cognitive architecture that was proposed as the original example of unified cognitive theory (Laird et al., 1986; Lewis, 2001; Newell, 1990). One of the main themes in Soar is that all cognitive tasks can be represented by problem spaces that are searched by production rules grouped into operators (Newell, 1990). These production rules are fired in parallel to produce reasoning cycles. Soar has been used to simulate several psychological tasks (for a review, see Lewis, 2001). However, CLARION includes a distinction between action-centered and non-action-centered (procedural and declarative) knowledge, and an additional distinction between explicit and implicit knowledge (Cleeremans and Dienes, 2008; Reber, 1989; Seger, 1994). Besides adding psychological plausibility (as demonstrated in many psychological tasks; for a review, see Sun et al., 2005), the inclusion of implicit knowledge in the bottom level of CLARION allows for a natural account of similarity-based reasoning. It was shown in the "*Capturing human cognition: Accounting for Psychological Quasi-laws"* section earlier that similarity-based reasoning plays an important role in categorization, uncertain reasoning, and decision-making. While Soar might be tweaked to reproduce these results (because it is Turing-equivalent), an explicit (and *ad hoc*) representation of similarity can become cumbersome

when a large number of representations are used. CLARION thus provides a more intuitive and elegant account of the psychological quasi-laws.

As another example, ACT-R is a production-based cognitive architecture aimed at explaining psychological processes (Anderson and Lebiere, 1998). ACT-R is composed of four modules: the perceptual-motor module, the goal module, the declarative module, and the procedural module. ACT-R has been applied to a large number of psychological tasks (as reviewed in, e.g., Anderson and Lebiere, 1998). ACT-R distinguishes between procedural and declarative memories. In addition, ACT-R has a (somewhat crude) representation of the dichotomy of explicit and implicit memories: explicit memory is represented by the memory structures (i.e., chunks and production rules) while implicit memory is represented by the activation of the memory structures. In contrast, the distinction between explicit and implicit memories in CLARION is one of the main focuses of the architecture, and a more detailed representation of implicit knowledge has allowed for natural similarity-based reasoning as well as natural simulation of many psychological data sets (e.g., Helie and Sun, 2010; Sun et al., 2005). In addition, CLARION separates action-centered rules from non-action-centered rules (while in ACT-R, all the rules are included in the procedural module), as well as action-centered implicit knowledge from non-action-centered implicit knowledge. The distinction between action-centered and non-action-centered knowledge has proven to be essential in many simulations (e.g., Sun, 2002; Sun et al., 2009).

As yet another example, Bayesian models interpret the human mind as performing statistical inference. When several variables are involved or the statistical dependencies are complex, graphical models (Bayesian networks; Pearl, 1988) can be useful tools to

clarify the relations. Bayesian models have been used to address the issues of animal learning, human inductive learning and generalization, visual scene perception, motor control, semantic memory, language processing and acquisition, and social cognition (Gopnik and Glymour, 2006; Griffiths et al., 2008). The structure of a Bayesian network is similar to that of a connectionist network: both are composed of nodes and edges (Gopnik and Glymour, 2006). In a Bayesian network, the representations are localist (i.e., each variable is represented by one node), and there is no distinction between input and output nodes. This is akin to the network representing the top level of the NACS in CLARION (Helie and Sun, 2010; Sun, 2002; 2003). However, the parameters of the networks are different. In a connectionist network, parameters are connection weights, and each edge has a single parameter. In a Bayesian network, the parameters are conditional probabilities, and each edge has several parameters. Information propagation is also different (i.e., weighed sum vs. Bayesian updating). The similarities between Bayesian networks and connectionist networks enable these models to explain many of the same phenomena. However, the explanations provided differ. Gopnik and Glymour (2006) argued that Bayesian and connectionist models offer complementary explanations of the human mind: Bayesian models offer intuitive explanation of some of the afore-mentioned psychological quasi-laws, but would have to be unnaturally twisted to account for other phenomena (e.g., deduction; see Griffiths et al., 2008). Also, similarity has to be represented by probabilities in Bayesian computation. The literature on uncertain reasoning has shown that this approach is not always adequate (Tversky and Kahneman, 1974). In CLARION, similarity is naturally captured by feature-based representations in the bottom level. For these reasons, we believe that CLARION offers a more natural,

more complete, and more algorithmic explanation of the psychological quasi-laws than Bayesian models alone (see Helie and Sun, 2009 for further details).

We may also compare CLARION with the BDI approach: which asserts that desires lead to intention and intentions lead to actions based on beliefs (knowledge). BDI is basically folk psychology: The framework may be a useful engineering framework. But it is not a scientific framework concerning human cognition/psychology at any reasonably detailed level. This is because this framework is not psychologically detailed and realistic: For example, it does not capture important fine-grained details (processes and mechanisms) of human motivations and their relations to actions (see, e.g., Sun, 2009). As a result of the above, it has thus far no significant impact on cognitive science or psychology. However, we may nevertheless identify the following rough correspondence between the BDI framework and the (more detailed, more process-oriented) CLARION theory: (a) desires = drives; (b) intentions = goals; (c) beliefs = knowledge in the ACS/NACS (both implicit and explicit). In CLARION, given a specific context, drive activations lead to the setting of goals, which in turn lead to actions on the basis of existing knowledge, in rough correspondence with BDI.

In sum, it appears that CLARION provides a viable alternative for building psychologically realistic cognitive agents, which can be either for theoretical understanding of human cognition, or for developing cognitively oriented intelligent systems for applications.

Concluding Remarks

Langley and Laird (2003) proposed a set of criteria for evaluating cognitive architectures in the context of AI and cognitive agents, including: (1) generality,

versatility, and task ability, (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. It was claimed that future promise of cognitive architectures for building models of cognitive agents lies in addressing these challenges (see, e.g., Sun, 2009; Sun, 2006). One way to work on these criteria is to focus on building psychologically realistic cognitive architectures, because human intelligence shows a proper balance in meeting these criteria.

This paper explored the core theory of CLARION and showed how it captured many psychological quasi-laws related to categorization, uncertain reasoning, decision-making, and so on (see also Helie and Sun, 2009 for more such quasi-laws). These psychological phenomena were captured within the Non-Action-Centered Subsystem (NACS) of CLARION. This work is a step in bridging the gap between artificial agents and psychologically realistic (brain/mind-inspired) models, and that between mathematically simple cognitive models and general cognitive architectures. Hence, it serves as an approach towards building psychologically realistic cognitive agents that can start addressing the challenges above.

References

Anderson, J.A., Silverstein, J.W., Ritz, S.A., and Jones, R.S. (1977). Distinctive features, categorical perception, and probability learning: Applications of a neural model. *Psychological Review*, *84*, 413-451.

Anderson, J. R. (1990). *The Adaptive Character of Thought*. Hillsdale, NJ: Erlbaum.

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

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

Beer, R.D. (2000). Dynamical approaches to cognitive science. *Trends in Cognitive Sciences*, *4*, 91-99.

Berry, D.C. and Broadbent, D.E. (1988). Interactive tasks and the implicit – explicit distinction. *British Journal of Psychology*, *79*, 251-272.

Bundesen, C., Habekost, T., and Kyllingsbæk, S. (2005). A neural theory of visual attention. Bridging cognition and neurophysiology. *Psychological Review, 112*, 291-328.

Busemeyer, J.R. and Diederich, A. (2002). Survey of decision field theory. *Mathematical Social Sciences*, *43*, 345-370.

Busemeyer, J.R. and Johnson, J.G. (2008). Micro-process models of decision making. In R. Sun (Ed.) *The Cambridge Handbook of Computational Psychology* (pp. 302-321). Cambridge University Press.

Chartier, S. and Proulx, R. (2005). NDRAM: A Nonlinear Dynamic Recurrent Associative Memory for learning bipolar and nonbipolar correlated patterns. *IEEE Transactions on Neural Networks*, *16*, 1393-1400.

Cleeremans, A., and Dienes, Z. (2008). Computational models of implicit learning. In R. Sun (Ed.) *The Cambridge Handbook of Computational Psychology* (pp. 396-421). Cambridge University Press.

Cohen, M. A. and Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-13*, 815-826.

Collins, A. and R. Michalski, (1989). The logic of plausible reasoning. *Cognitive Science*, 13(1), 1-49.

Collins, A. M., and Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal Behavior and Verbal Learning, 8*, 432-438.

Crutchfield, J.P. (1998). Dynamical embodiements of computation in cognitive processes. *Behavioral and Brain Sciences*, *21*, 635.

Fodor, J. and Pylyshyn, Z. (1988). Connectionism and cognitive architecture: A critique. *Cognition*, *28*, 3-71.

Gopnik, A. and Glymour, C. (2006). A brand new ball game: Bayes net and neural net learning mechanisms in young children. In Y. Munakata and M. Johnson (Eds.) *Processes of Change in Brain and Cognitive Development: Attention and Performance XXI*. Oxford: Oxford University Press.

Griffin, D., and Tversky, A. (1992). The weighing of evidence and the determinants of confidence. *Cognitive Psychology*, *24*, 411-435.

Griffiths, T.L., Kemp, C., Tenenbaum, J.B. (2008). Bayesian models of cognition. In R. Sun (Ed.) *The Cambridge handbook of computational cognitive modeling* (pp. 59-100). Cambridge University Press.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. Biological Cybernetics, 23, 121-134.

Grossberg, S. (1988). Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, *1*, 17-61.

Harnad, S. (2005). To cognize is to categorize: Cognition is Categorisation. In C. Lefebvre and H. Cohen (Eds.) *Handbook of Categorization in Cognitive Science* (pp. 20-45). Oxford: Elsevier.

Haykin, S. (2009). *Neural Networks and Learning Machines. 3rd Edition*. Upper Saddle River, NJ: Prentice-Hall.

Hélie, S. (2008). Energy Minimization in the Nonlinear Dynamic Recurrent Associative Memory. *Neural Networks*, *21*, 1041-1044.

Hélie, S., Chartier, S., and Proulx, R. (2006). Are unsupervised neural networks ignorant? Sizing the effect of environmental distributions on unsupervised learning. *Cognitive Systems Research*, *7*, 357-371.

Hélie, S. and Sun, R. (2010). Incubation, insight, and creative problem solving: A unified theory and a connectionist model. *Psychological Review*.

Helie, S. and Sun, R. (2009). A unified account of psychological laws. Technical Report, RPI, Troy, NY.

Homa, D., and Vosburgh, R. (1976). Category breadth and abstraction of prototypical information. *Journal of Experimental Psychology: Human Learning and Memory*, *2*, 322-330.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*, 2554-2558.

Kieras, D.E. and Meyer, D.E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction*, *12*, 391-438.

Kruschke, J.K. (2008). Models of categorization. In R. Sun (Ed.) *The Cambridge Handbook of Computational Psychology* (pp. 267-301). Cambridge University Press.

Laird, J.E., Newell, A., and Rosenbloom, P.S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, *33*, 1-64.

Lewis, R.L. (2001). Cognitive theory, Soar. In N.J. Smelser and P.B. Baltes (Eds.) *International Encyclopedia of the Social and Behavioral Sciences*. Elsevier.

Mathews, R.C., Buss, R.R., Stanley, W.B., Blanchard-Fields, F., Cho, J.R., and Druhan, B. (1989). Role of implicit and explicit processes in learning from examples: A synergistic effect. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *15*, 1083-1100.

Meyer, D. and Kieras, D. (1997). A computational theory of executive cognitive processes and human multiple-task performance: Part 1, basic mechanisms. *Psychological Review*, *104*, 3-65.

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

Newell, A. and Rosenbloom P. (1981). Mechanisms of skill acquisition and the law of practice. In J.R. Anderson (Ed). *Cognitive Skills and Their Acquisition* (pp. 1-55). Hillsdale, NJ: Erlbaum Associates.

Nisbett, R.E., Krantz, D.H., Jepson, C., and Kunda, Z. (1983). The use of statistical heuristics in everyday inductive reasoning. *Psychological Review*, *90*, 339-363.

Nissen, M., and Bullemer, P. (1987). Attentional requirements of learning: Evidence from performance measures. *Cognitive Psychology*, *19*, 1–32.

Nosofsky, R.M. (1988). Similarity, frequency, and category representations. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *14*, 54-65.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference*. San Mateo, CA: Morgan Kaufmann Publishers.

Pitt, M.A. and Myung, I.J. (2002). When a good fit can be bad. *Trends in Cognitive Sciences*, *6*, 421-425.

Quillian, M.R. (1968). Semantic memory. In M. Minsky (Ed.) *Semantic Information Processing* (pp. 216-270). Cambridge, MA: MIT Press.

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

Roberts, S. and Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review*, *107*, 358-367.

Roe, R.M., Busemeyer, J.R., and Townsend, J.T. (2001). Multialternative decision field theory: A dynamic connectionist model of decision making. *Psychological Review*, *108*, 370-392.

Rosch, E. and Mervis, C.B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, *7*, 573-605.

Schyns, P.G., Goldstone, R.L., and Thibault, J.P. (1998). The development of features in object concepts. *Behavioral and Brain Sciences*, *21*, 1-54.

Schyns, P. and Rodet, L. (1997). Categorization creates functional features. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *23*, 681, 696.

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

Shepard, R.N. (1987). Toward a universal law of generalization for psychological science. *Science*, *237*, 1317-1323.

Sloman, S.A. (1996). The empirical case for two systems of reasoning. *Psychological Bulletin*, *119*, 3-22.

Sloman, S. (1998). Categorical inference is not a tree: The myth of inheritance hierarchies. *Cognitive Psychology*, *35*, 1-33.

Stadler, M. (1995). Role of attention in implicit learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *21*, 674 – 685.

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

Sun, R. (1992). On variable binding in connectionist networks. *Connection Science, 4*, 93-124.

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

Sun, R. (2002). *Duality of the Mind: A Bottom-up Approach Toward Cognition*. Mahwah, NJ: Lawrence Erlbaum Associates.

Sun (2003). *A Tutorial on CLARION 5.0.* Technical Report, Department of Cognitive Science, Rensselaer Polytechnic Institute, Troy, NY.

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

Sun, R. (2006). *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*. Cambridge University Press.

Sun, R. (2007). The motivational and metacognitive control in CLARION. In W. Gray (ed.) *Modeling Integrated Cognitive Systems* (pp. 63-75). New York: Oxford University Press.

Sun, R. (2009). Motivational representations within a computational cognitive architecture. *Cognitive Computation*, 1 (1), 91-103.

Sun, R., Merrill, E., and Peterson, T. (2001). From implicit skills to explicit knowledge: A bottom-up model of skill learning. *Cognitive Science, 25*, 203-244.

Sun, R. and Peterson, T. (1998). Autonomous learning of sequential tasks: Experiments and analyses. *IEEE Transactions on Neural Networks*, *9*, 1217-1234.

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

Sun, R. and Zhang, X. (2006). Accounting for a variety of reasoning data within a cognitive architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, *18*, 169-191.

Sun, R., Zhang, X., and Mathews, R. (2009). Capturing human data in a letter counting task: Accessibility and action-centeredness in representing cognitive skills. *Neural Networks*, *22*, 15-29.

Szymanski, K., and MacLeod, C. (1996). Manipulation of attention at study affects an explicit but not an implicit test of memory. *Consciousness and Cognition*, *5*, 165–175.

Tenenbaum, J.B. and Griffiths, T.L. (2001). Generalization, similarity, and Bayesian inference. *Behavioral and Brain Sciences, 24,*629-641.

Thurstone, L.L. (1927). A law of comparative judgement. *Psychological Review*, *34*, 273-286.

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

Townsend, J.T. (1992). Unified theories and theories that mimic each other's predictions. *Behvaioral and Brain Sciences*, *15*, 458-459.

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*, 327-352.

Usher, M. and McClelland, J.L. (2004). Loss aversion and inhibition in dynamical models of multialternative choice. *Psychological Review*, *111,* 757-769.

Watkins, C. (1989). *Learning From Delayed Rewards*. Doctoral Dissertation, Cambridge University, Cambridge, UK.

Weiner, B. (1992). *Human Motivation: Metaphors, Theories, and Research.* Sage, Newbury Park, CA.

Appendix: Decision Field Theory

Decision Field Theory (DFT) is a prominent theory of human decision-making (Busemeyer and Diederich, 2002; Busemeyer and Johnson, 2008; Roe et al., 2001). DFT models the decision *process* instead of focusing on the end-state. DFT can be understood using two fundamental notions: valence and preference (Roe et al., 2001). The *valance* of an option is the momentary advantage/disadvantage of an option in relation to the other options. In contrast, the *preference* of an option refers to the accumulation of all the valences that this option has received in the past.

First, we explain how to compute the *valence*. The valence of option i at time t, denoted $v_i(t)$, is a function of three components. The first component, $\mathbf{M} = [m_{ij}]$ is a matrix representing the personal evaluation of each of the option attributes (m_{ij} is the value of option i on attribute j). For instance, quality and price can be used as attributes when considering options for buying a car (Roe et al., 2001).

The second component of valence is attentional weight allocation. At every time step, one might focus on a different attribute of the options. For instance, one might be thinking specifically about car quality at the present moment, and switch attention to price at the following time step. Attention allocation is represented by the vector $\mathbf{w}(t)$. This notation highlights the temporal dimension of attentional allocation. By default, only one of the attributes is attended to at any moment and attention is switched either randomly or using a Markov process (Roe et al., 2001). At this point, it is useful to note that the matrix product $\mathbf{Mw}(t)$ represents the weighed value of each option at time t (independent from the other options in the set). For instance, a particular car (option) might look very interesting when focusing on the quality attribute but not so much when

focusing on the price attribute. Hence, a different weighed value would be assigned to the car at time t depending on which attribute received more attention at time t.

The third and final component of valence is a comparison process that contrasts the weighed value of each option with the other options in the set (e.g., comparing the appeal of car A and car B at time t). This is defined by the $n \times n$ contrast matrix $\mathbf{C} = [c_{ij}]$, where $c_{ii} = 1$ and $c_{ij} = -1/(n-1)$ for $i \neq j$ (where n is the number of options simultaneously considered). Intuitively, the contrast matrix subtracts from each option's weighed value the average weighed value of the other options.

Overall, the valence vector $\mathbf{v}(t) = \{v_1(t), v_2(t), \ldots, v_n(t)\}$ is:

$$\mathbf{v}(t) = \mathbf{CMw}(t) \tag{A1}$$

where the symbols are as previously defined.

The second fundamental notion in DFT is *preference*. By accumulating the valences through time, each option is assigned a preference for each time step. The n-dimensional vector $\mathbf{p}(t)$ representing the preferences is defined by:

$$\mathbf{p}(t) = \mathbf{Sp}(t-1) + \mathbf{v}(t) \tag{A2}$$

where $\mathbf{v}(t)$ is the valence vector at time t (Eq. A1) and $\mathbf{S} = [s_{ij}]$ is a $n \times n$ inhibition matrix. The preference vector represents a process that accumulates valences across time for each option. A decision is made when the preference of one of the options crosses the upper bound or the time limit is reached. In the latter case, the maximum preference is chosen.

From Eq. A2, the dynamics of the decision process is determined by two factors: the initial state $\mathbf{p}(0)$ and the inhibition matrix $\mathbf{S}$. The initial state is usually set to be unbiased [$\mathbf{p}(0) = \mathbf{0}$]. However, a bias can be included to reflect the success of previous

choices. The **S** matrix contains n^2 parameters defining the inhibition between the options. In most applications, this large number of free parameters is reduced by making the inhibition matrix symmetric. Hence, DFT models usually have $n(n+1)/2$ free parameters. The diagonal elements s_{ii} represent the memory of the previous preferences. When $s_{ii} = 1$, the model has perfect memory whereas $s_{ii} = 0$ implies that the model has no memory whatsoever. The off-diagonal terms represent lateral inhibition of the options. If $s_{ij} = 0$ for $i \neq j$, there is no competition among the options and the preference of each option grows independently from the preferences of the other options. In contrast, if $s_{ij} < 0$ for $i \neq j$, the options inhibit each other. A general principle must be respected: the inhibition resulting from the preference of an option is a negative function of its similarity to the other options. Hence, two options that are very similar strongly inhibit one another whereas two options that are dissimilar only weakly inhibit one another. Respecting this principle when assigning the values to the free parameters is essential to the success of DFT (Usher and McClelland, 2004).

Acknowledgments

This research was supported in part by the *Army Research Institute* grant W74V8H-05-K-0002 and the *Office of Naval Research* grant N00014-08-1-0068, as well as by a postdoctoral research fellowship from *Le Fonds Québecois de la Recherche sur la Nature et les Technologies*.

Footnotes

[1] More complex rules can also be implemented in CLARION, including conjunctive rules, disjunctive rules, variable binding, etc. However, these are not part of the core theory of CLARION and are not included here. The interested reader is referred to Sun (1992, 2002).

[2] It should be noted that all rules fire in parallel in the NACS of CLARION. As such, a chunk can receive activation by more than one associative rules. In this case, the maximum rule-based activation is used.

[3] Alternatively, the density parameter (d_c) can be interpreted as a stopping criterion at which one stops searching for a chunk and assumes that it is not available in memory (i.e., forgotten).

[4] Absolute correlation is often used as a measure of proximity in Hopfield-type neural networks (Anderson et al., 1977; Haykin, 2009; Hopfield, 1982). High correlation is interpreted as short distance while low correlation is interpreted as long distance.

[5] This Hebbian learning rule (Eq. 9) allows NDRAM to learn a set of real-valued patterns in single-layered neural networks (Chartier and Proulx, 2005), bidirectional associative memories, and to perform density estimation (Hélie, Chartier, and Proulx, 2006). The convergence of this learning rule has been proven in previous papers (e.g., Chartier and Proulx, 2005).

[6] While this convergence property has been shown for many Hopfield-type neural networks (e.g., Cohen and Grossberg, 1983; Hopfield, 1982), NDRAM is one of the few networks that allows for the iterative learning of real-valued correlated attractrors (Chartier and Proulx, 2005).

[7] This case is a "weak" form of deduction, because rule-based reasoning is not involved, and similarity-based reasoning can converge toward full activation but never reaches it (unlike rule-based reasoning, which exactly transmits activation).

[8] Because all the intermediate results of DFT are numerical and "fuzzy", DFT has to be mainly carried out in the bottom level. In contrast, the top level would have difficulties representing the valence of the options, because top-level activation is usually binary/crisp and rule-based. Also, connection weights are usually non-negative in the top-level; hence, the preference inhibition matrix could not be easily represented in the top level. Finally, the dynamic in DFT is driven mainly by similarity-based "inhibition" (matrix **S** in Appendix), which is naturally carried out in the bottom level of the NACS.

[9] Alternatively, the ACS can choose to make a decision and halt the diffusion process, without considering the threshold. In this case, the option with the maximum internal confidence level is chosen (as in DFT).

[10] Alternatively, and consistent with previous parameter definitions, the diagonal terms of **S** can be set to ς, i.e., the memory efficiency parameter used for the attractor neural network in the bottom level of the NACS.

[11] ρ was used to explain phenomena in the *Categorization* subsection (vigilance), and ψ was used in the *Decision-making* subsection (the threshold on the internal confidence levels).

Figure captions

Figure 1. A high-level representation of CLARION.

Figure 2. An example auto-associative network in the bottom-level of the CLARION NACS. $\boldsymbol{x}_{[0]}$ is the initial activation of the bottom-level nodes, $\boldsymbol{x}_{[p]}$ is the final activation (to be sent bottom-up to the top level), $\overline{\mathbf{x}}$ is the output of the vigilance module, and $\mathbf{W}$ is the weight matrix.

Figure 3. A connectionist implementation of decision field theory.

Figure 1

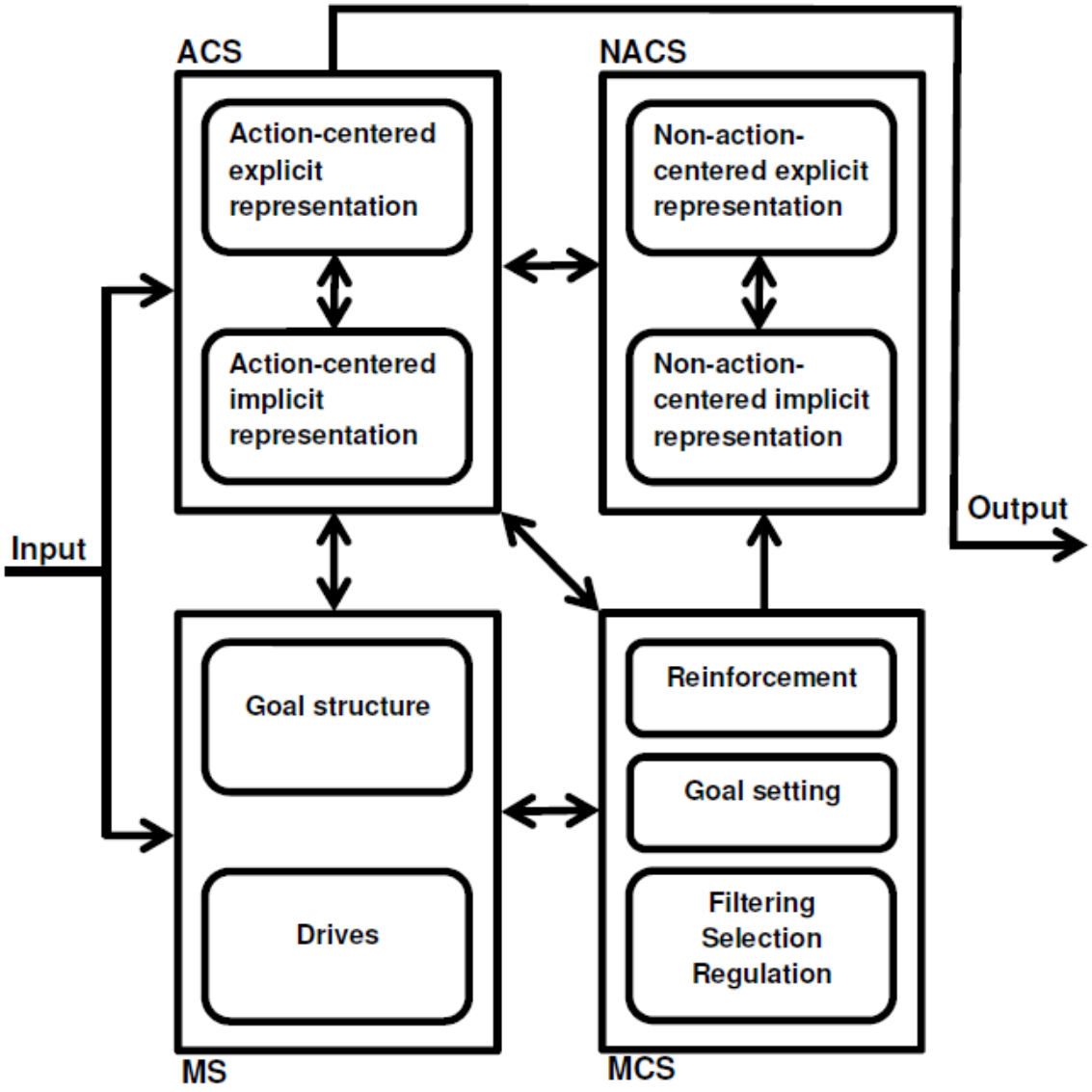

ACS
NACS
Action-centered explicit representation
Action-centered implicit representation
Non-action-centered explicit representation
Non-action-centered implicit representation
Input
Output
Goal structure
Drives
Reinforcement
Goal setting
Filtering
Selection
Regulation
MS
MCS

Figure 2

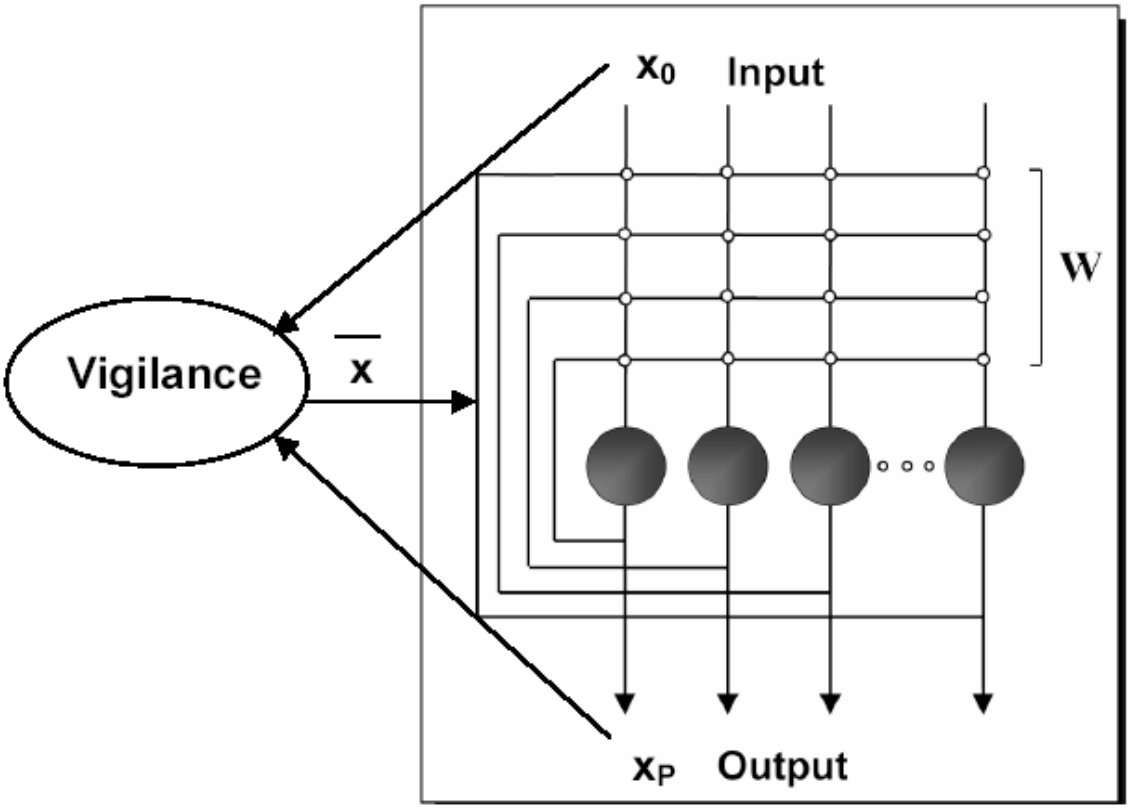
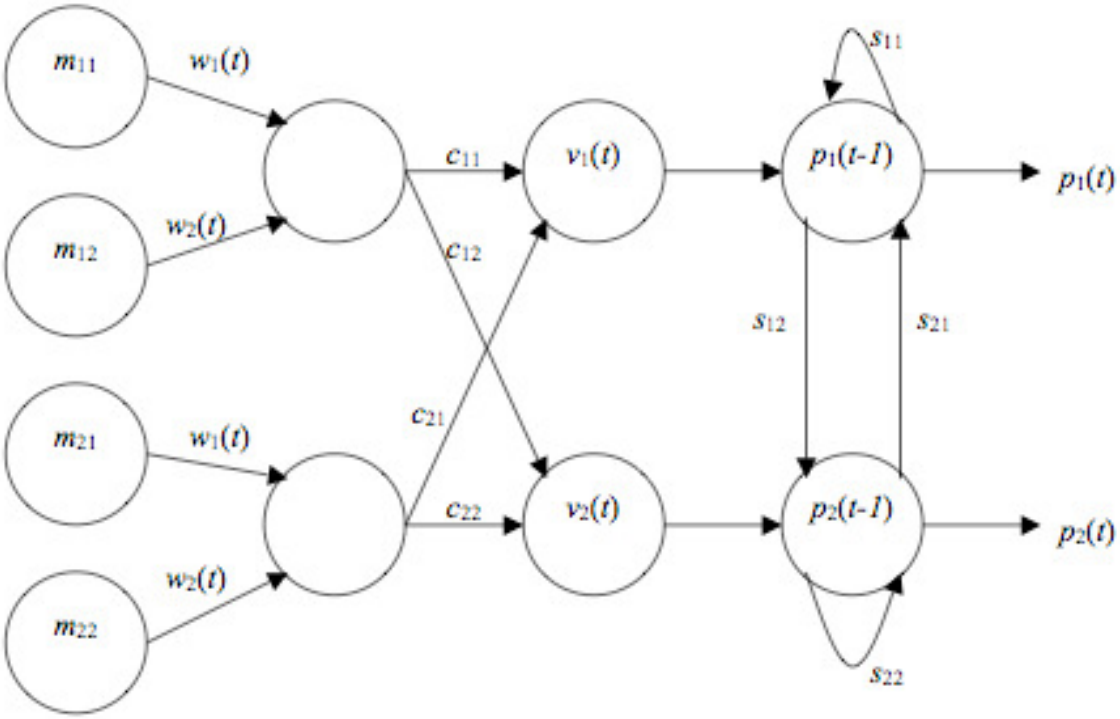


Figure 3


m_{11}
$w_1(t)$
m_{12}
$w_2(t)$
m_{21}
$w_1(t)$
m_{22}
$w_2(t)$
c_{11}
c_{12}
c_{21}
c_{22}
$v_1(t)$
$v_2(t)$
s_{11}
$p_1(t-1)$
$p_1(t)$
s_{12}
s_{21}
$p_2(t-1)$
$p_2(t)$
s_{22}

****

****

****