How the Core Theory of CLARION Captures Human Decision-Making

Sebastien Helie, Ron Sun

Abstract—**Some mainstream psychologists have criticized computational cognitive architectures on the issue of model complexity and parameter tweaking (i.e., the likelihood that cognitive architectures can explain any results and their opposites). This paper tries to address these criticisms by tackling the issue of model complexity in cognitive architectures. Here, we start with a well-established cognitive architecture, CLARION, and extract its core theory to explain a wide range of data. The resulting minimal model was used to provide parameter-free principled explanations for several psychological "laws" of uncertain reasoning and decision-making. This paper is concluded by a discussion of the implication of parameter-free modeling in cognitive science and psychology.**

I. Introduction

Cognitive theories are often underdetermined by data [1]. As such, different theories, with very little in common, can be used to explain the very same phenomena observed in experimental psychology [2]. According to Newell, this problem can be resolved by adding constraints to psychological theories. The most intuitive approach to adding constraints to any scientific theory is to collect more data. Newell [1] argued that more data could be used to constraint a theory if the theory was designed to explain a wider range of phenomena (both from the same and other domains). So far such 'integrative' theories have taken the form of *cognitive architectures*, and some of them have been very successful at explaining a wide range of data (e.g., [3]-[4]). However, on the down side, cognitive architectures tend to be complex, including multiple modules and many free parameters (in their computational implementations).

The problem of complexity has been recognized and acknowledged by Sun [5], who argued that a cognitive architecture should be *minimal*. Minimality in cognitive architectures needs to be attained in two senses. First, a cognitive architecture should have only minimal initial structures (i.e., modules). Second, the internal structures and representations should also be kept to a minimum while capturing human data. In this article, we study how the CLARION cognitive architecture [4], [6]-[8] can be minimized, which leads to extracting its core theory.

The remainder of this article is organized as follow. First, Section II introduces a core theory extracted from the CLARION cognitive architecture. Second, the core theory extracted in Section II is used to provide principled (and almost parameter-free) explanations of decision-making phenomena (Section III) and biases in medical diagnoses (Section IV). This article concludes with a short discussion of the implications of research on minimal cognitive architectures.

II. The CLARION Cognitive Architecture

CLARION is a cognitive architecture that is, in part, based on two basic assumptions: representational differences and learning differences of two different types of knowledge: implicit versus explicit [4], [6]-[8]. These two types of knowledge differ in terms of accessibility and attentional requirement. The top level of CLARION (as in Fig. 1) contains explicit knowledge (easily accessible, but requiring more attentional resources) whereas the bottom level contains implicit knowledge (harder to access, but mostly automatic). Because knowledge in the top and bottom levels is different, Sun and his colleagues [6]-[8] 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.

CLARION is further divided into two different subsystems: the *Action-Centered Subsystem* and the *Non-Action-Centered Subsystem*. The Action-Centered Subsystem (with both levels) contains procedural knowledge concerning actions and procedures (i.e., it serves as the long-term procedural memory), while the Non-Action-Centered Subsystem (with both levels) contains declarative knowledge (i.e., it serves as the long-term declarative memory, both semantic and episodic; [8]). The Non-Action-Centered Subsystem is controlled by the Action-Centered Subsystem. The Non-Action-Centered Subsystem is also used for various types of reasoning [6], [9].

The second assumption in CLARION concerns the existence of different learning processes in the top and bottom levels [7]-[8]. 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

Manuscript received May 5, 2011. This research was supported by a postdoctoral research fellowship from *Le Fonds Québecois de la Recherche sur la Nature et les Technologies* to the first author and research grants provided by the *Army Research Institute*, contracts DASW01-00-K-0012 and W74V8H-04-K-0002, to the second author.

S. Helie is with the Department of Psychological & Brain Sciences, University of California, Santa Barbara, CA 93106-9660 USA (phone: 805-284-9474; fax: 805-893-4303; e-mail: helie@psych.ucsb.edu).

R. Sun is with the Cognitive Science Department, Rensselaer Polytechnic Institute, Troy, NY 12180-3590 USA (e-mail: rsun@rpi.edu)

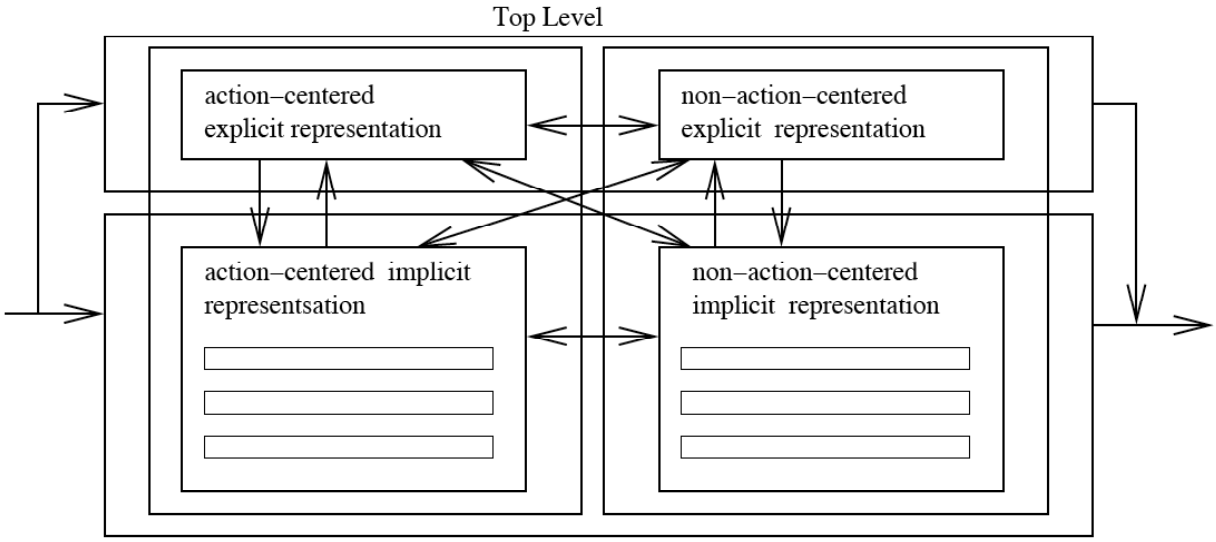

Fig. 1. A high-level representation of CLARION.

explicit) information. The inclusion of and the emphasis on bottom-up learning (i.e., the transformation of implicit knowledge into explicit knowledge) is, in part, what distinguishes CLARION from other cognitive models (for another example, see [10]).

A. *The Action-Centered Subsystem*

The Action-Centered Subsystem (ACS) is the main subsystem in CLARION [7]-[8]. In addition to filling the role of long-term procedural memory, the ACS is used to capture some executive functions (i.e., the control of some other subsystems). As such, the ACS receives all the inputs from the environment, and provides action recommendations. The description of the implementation of the ACS included in the present paper is conceptual, because technical formalities are not required to explain the range of phenomena accounted for in this paper. Readers interested in the technical aspects of the ACS are referred to [7]-[8].

1) Top Level: In the top level of the ACS (the *Action Rule Store*), explicit knowledge is represented using condition and action chunks. Condition chunks can be activated by the environment (e.g., a stimulus) or other CLARION subsystems (e.g., working memory). Action chunks can represent motor programs (i.e., a response) or queries to other CLARION subsystems. In particular, an action recommendation of the ACS can be used to query the Non-Action-Centered Subsystem with a round of reasoning (as detailed later). In this case, the Non-Action-Centered Subsystem can return one or several chunks resulting from the round of reasoning, which can be used in the ACS as action recommendations or as conditions for computation at a future time step.

Chunks returned by the Non-Action-Centered Subsystem are accompanied by their *internal confidence levels* (i.e., activations). The internal confidence level is for estimating the confidence in the answer returned to the ACS. This measure is important because the ACS does not have direct access to the processing that led to this chunk being returned. The ACS can use a threshold (i.e., ψ) on the internal confidence level to decide on accepting/rejecting the result of NACS processing. Also, the internal confidence level can be used to estimate confidence in a produced response.

Both condition and action chunks are individually represented by single nodes in a connectionist network and have a clear conceptual meaning (i.e., localist representations). The chunks in the top level of the ACS are linked to implement rules of the form "Condition chunks" → "Action chunks". These rules can be simply represented by connections weights, thus forming a linear connectionist network. These explicit procedural rules, and the chunks involved, can be learned bottom-up (via Rule-Extraction-Refinement; [7]), by explicit hypothesis testing (via Independent Rule Learning), or be fixed (e.g., by experimental instructions; Fixed Rules). In all cases, top-level rules are learned in a "one-shot" fashion [4].

2) Bottom Level: The bottom level of the ACS (*Implicit Decision Networks*) uses feature-based representations to capture implicit procedural knowledge. Each top-level chunk is represented by a set of features in the bottom level (i.e., distributed representations). The chunk features (in the bottom level) are connected to the chunks (in the top level) so that they are usually activated together through bottom-up activation (when the features are activated first) or top-down activation (when the chunks are activated first).

The features of the condition and action chunks are connected in the bottom level using several specialized multilayer nonlinear connectionist networks. Each network can be thought of as a highly efficient routine (once properly trained) that can be used to accomplish a particular task. Training of the bottom-level networks is iterative and done using backpropagation implementing Q-learning [7].

B. *The Non-Action-Centered Subsystem*

The Non-Action-Centered Subsystem (NACS) of CLARION is a slave-system used to capture the declarative (both semantic and episodic) long-term memory [4]. The inputs and outputs of this subsystem usually come from another subsystem in CLARION, namely the ACS. In

addition, the NACS is used to capture several forms of reasoning [6], [9]. Here, a technical description of the core processes of the NACS is provided below. (The reader interested in the complete description is referred to [4].)

1) Top Level: In the top level of the NACS (the *General Knowledge Store*), explicit knowledge is represented by chunks (as in the ACS top-level). However, unlike in the ACS, NACS chunks are not divided into condition and action chunks: all chunks represent concepts that can be used as a condition or a conclusion in rule application. Each chunk can be activated by: (a) an ACS query, (b) its association with another chunk (via an associative rule), or (c) its similarity to another chunk (via a similarity measure). When a NACS chunk is activated by an ACS query, its activation is generally set to unity (i.e., $s_j^{ACS} = 1$). However, the other two sources of activation can have smaller (positive) values.

NACS chunks can be linked together to represent 'associative' rules (similar to a semantic network). 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 j following the application of an associative rule, s_i is the activation of chunk i, and w_{ij}^r is the strength of the associative rule between chunks i and j (by default, $w_{ij}^r = 1/n$, where n is the number of chunks in the condition of the associative rule).[1] The application of (1) is referred to here as *rule-based reasoning* [11].

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*. More precisely,

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

where s_j^s is the activation of chunk 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 i. The similarity metric ($s_{ci\sim cj}$) is defined in the bottom level of the NACS and is detailed in the following subsection (see (6) below).

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

$$s_j = Max(s_j^{ACS}, \beta_1 \times s_j^r, \beta_2 \times s_j^s) \tag{3}$$

where s_j is the overall activation of chunk j, and β_1 and β_2 are scaling parameters quantifying the weights of rule-based and similarity-based reasoning respectively.[2] By default, $\beta_1 = \beta_2 = 1$.

Regardless of the activation source, chunks that are inferred (activated) in the NACS may be sent to the ACS for consideration in action decision-making. Every chunk that is sent back to the ACS is accompanied by an internal confidence level (activation, as in (3)).

When only one chunk is to be returned to the ACS, a chunk is stochastically selected using a Boltzmann distribution:

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

where P(chunk j) is the probability that chunk j is selected to be returned to the ACS, s_j is the activation of chunk j (3), and α is a free parameter representing the degree of randomness (temperature). In cases where only one chunk is returned to the ACS, this normalized activation is used as the internal confidence level (instead of the 'raw' activation of (3)).

In addition to the above-mentioned activation, each chunk has a base-level activation 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 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 of the chunk. This measure has an exponential decay and corresponds to the odds of needing chunk j based on past experiences [12]. When the base-level activation of a chunk 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 mostly for capturing forgetting (using the density parameter).[3]

Like in the ACS, chunks in the NACS can be learned by explicitly encoding given information (using, e.g., Fixed Rules) and by explicitly encoding knowledge bottom-up from the bottom levels of CLARION (both from the ACS and the NACS; e.g., by using Rule-Extraction-Refinement). In addition, each item in working memory has probability p of being encoded in the NACS as a chunk at every time step (for details on working memory, see [4]).

2) Bottom Level: As in the ACS, the bottom level of the NACS (i.e., the *Associative Memory Networks*) uses feature-based representations to (often redundantly) encode the top-level chunks with distributed representations [6]. The features are connected to the top-level chunks so that, when a chunk is activated, its corresponding bottom-level feature-based representation (if exists) is also activated and vice-

[1] 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.

[2] It should be noted that, mathematically, β_1 and β_2 add a single degree of freedom to the model. However, separate free parameters were used here to facilitate the interpretation of the parameter values.

[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).

versa. Alternatively, any bottom-level feature in the NACS can be directly activated by an ACS query.

The connections between top-level chunks and their feature-based representations allow for a natural computation of similarity (2):

$$s_{c_i \sim c_j} = \frac{n_{c_i \cap c_j}}{f\left(n_{c_j}\right)} \tag{6}$$

where n_{cj} represents the number features in chunk j, $n_{ci \cap cj}$ is the feature overlap between chunks i and j, and $f(x)$ is a slightly super-linear positive function (by default, $f(x) = x^{1.1}$). Thus, similarity-based reasoning in CLARION is naturally accomplished using (a) top-down activation by chunks of their feature-based representations, (b) calculation of feature overlap between any two chunks (6), and (c) bottom-up activation of the top-level chunks (2).

One important form of similarity-based reasoning is *inheritance-based* inference. In CLARION, this is done using the *reverse containment principle* [11]. 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) feature-based description 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 the emulation of a hierarchy of concepts (in the ideal case; [11]).

III. Decision-Making

Decision-making in psychology is concerned with choices and preferences. Preferences in decision-making have been modeled very early in psychology and economics (e.g., [13]). When only two choices are available, important phenomena from the psychological literature include strong stochastic transitivity, independence of irrelevant alternatives, and regularity in binary choices (as reviewed in [14]). When more than two choices are available, the main phenomena observed are the similarity effect, the attraction effect, the compromise effect, and the complex interactions between these effects (as reviewed in [15]).

Throughout the years, several models of decision-making have been proposed (e.g., elimination by aspect, Thurstone preferential model, additive utility models, etc; for a review, see [16]). While these models usually have the desirable property of being amenable to analytical solutions, they cannot account for all the afore-mentioned effects simultaneously. To our knowledge, 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 [16]. CLARION embodies such a model in its NACS.

A. Capturing and Further Enhancing Decision Field Theory within CLARION

We now examine the role of decision field theory (DFT) in the CLARION cognitive architecture. The reader not familiar with DFT and/or its terminology is referred to the *Appendix* for a summary description. First, decision-making is carried out in the CLARION NACS [4]. Because all the intermediate results of DFT are numerical and fuzzy, DFT is likely to be mainly carried out in the bottom level of the NACS. In addition, the dynamic in DFT is driven mainly by similarity-based inhibition (matrix **S** in the Appendix), which is also naturally carried out in the bottom level of the NACS. Similar to the ACS, the bottom level of the NACS is composed of several specialized networks (e.g., backpropagation networks). Hence, a special module in the bottom level of the NACS of CLARION, devoted to such decision-making, can include a connectionist network implementing DFT (as proposed by [15]).[4]

Before describing the network implementation, two points need to be emphasized. First, including a DFT network in the bottom level of the NACS does not increase the number of free parameters in CLARION, and only one free parameter is varied to explain the decision-making phenomena (i.e., ψ, the threshold on the internal confidence level). Second, each option from the DFT network is redundantly represented as a chunk in the top level of the NACS, and the activations of the option chunks are equal to the outputs from the fourth layer of the DFT network (i.e., the preferences of the options; as detailed next).

B. A Connectionist Implementation of Decision Field Theory

A connectionist network implementing decision field theory (DFT) within the bottom-level NACS of CLARION is presented in Fig. 2. As can be seen, the matrix formulation of DFT (as presented in the Appendix) allows for a natural representation in a four-layer connectionist network. In connectionist terminology, the personal evaluation of each option attribute (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**) represents the (fixed) weight connections between the second and third layers of the network. The activation in the third layer represents the *valence* of each option (i.e., 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*. Note that the connections between the third and fourth layers are direct (i.e., they do not carry a weight). A more complete description of the network can be found in [15].

[4] 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 valence and preference inhibition matrices could not be easily represented in the top level.

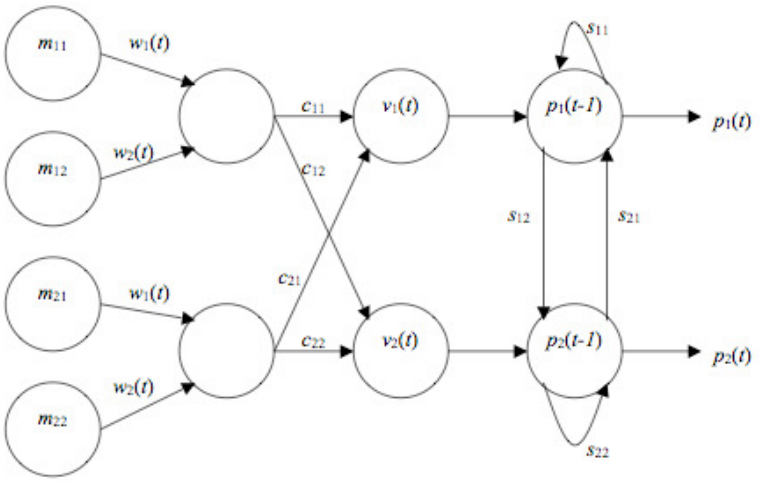

Fig. 2. A connectionist implementation of decision field theory.

1) 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 chunks representing the options (and the option chunk activations are equal to the preferences).

The chunks in the top level representing the options are retrieved by the ACS, and their activations are used to estimate 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, because the chunk activations in the top level *are* the preferences of the corresponding options in the bottom level DFT network). 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.[5]

2) Advantages: Including the DFT network in the bottom level of the NACS enhances both the decision-making capabilities of CLARION and the generality of the phenomena that DFT can account for. For instance, the top-level chunks in CLARION are connected with other chunks that can allow for rule-based reasoning and similarity-based reasoning to be carried out [20], which could not have been carried out within DFT alone. The rules in the top level can also be used to validate the option (chunk) chosen by the DFT network. The validation rules can include, e.g., moral imperatives, cultural conventions, behavioral norms, etc. (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., [17], regarding internalization as top-down learning). By capturing the duality and the complex interaction of explicit and implicit processes, CLARION adds new dimensions to DFT.

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, the inhibition matrix is automatically redefined because changing the features considered changes the similarity relations between the chunks (options) in CLARION (see (6)). In all cases, when the set of features or options is modified, the diffusion process in the DFT network is re-initialized.

Similarity-based reasoning in CLARION also plays an important role in decision-making. Specifically, the similarity between the chunks representing the options in the top level of CLARION is used to define the inhibition matrix:

$$\forall_{i \neq j},\ s_{ij} = -\left(s_{c_i \sim c_j}\right) = -\frac{n_{c_i \cap c_j}}{f(n_{c_j})} \tag{7}$$

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 can simply be set to one.) This definition satisfies the constraint on the inhibition matrix as defined in DFT (i.e., that inhibition is negatively related to similarity: [14]-[15]). 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.

In addition, 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; e.g., [18]). This can account

[5] 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).

for the (sometimes arbitrary) attention selection process underspecified in DFT.

Finally, the initial state of the diffusion process [initial preferences; $\mathbf{p}(0)$] can be initialized using the base-level activations of the chunks representing the options (5). This can be used to 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.

C. Discussion

It was shown in [16] that DFT can account simultaneously for a number of important phenomena in human cognition: 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 [14]. In multi-alternative choices, DFT can account for similarity effects, the attraction effect, the compromise effect, and the complex interaction between these phenomena [15]. In this work, we have shown how DFT can be enhanced by its inclusion in the bottom level of the NACS. CLARION eliminates all the free parameters in DFT by defining the inhibition matrix using similarity-based reasoning and the random-walk boundary with the ACS threshold on internal confidence levels. Also, rule-based reasoning in CLARION can be used to select options and attributes based on different external rules, including a validation process. Because CLARION includes a DFT network, it can also account for all the empirical phenomena discussed above, as well as more complex sociocultural decision-making situations [19]. Detailed explanations for the empirical phenomena discussed 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.

IV. Decision Making in Medical Diagnostics

CLARION has been used to model several biases in decision-making and uncertain reasoning found in psychology experiments (see [20]). However, such research also has several real-life applications. One example application is medical diagnosis [21]. Here we present two example biases that can be accounted for in CLARION.

A. Unpacking Principle

Physicians tend to stop gathering information about patient symptoms when they are fairly certain of their diagnosis [21]. This can leave out important information that could affect the final diagnosis. For instance, a physician could make a diagnosis before learning about a patient full medical history and not consider some possible diagnoses that may be relevant (so that these additional possible diagnoses are not even considered as possibilities). In CLARION, this is represented by a parameter that acts as a threshold (ψ) on the internal confidence level received by the ACS from the NACS. When the threshold is crossed by the internal confidence level, the model stops gathering information and the ACS outputs the response. When the threshold is not crossed, the ACS is not satisfied with the chunk(s) received from the NACS and initiates another round of processing (by gathering more information and sending a new request to the NACS). The lower the value assigned to ψ, the more likely that some relevant information is missed and that some relevant chunk will never be considered in the decision. Hence, CLARION can account for the unpacking principle by setting the threshold on the internal confidence level to a low value.

B. Ascertainment Bias

Another common bias in medicine is the ascertainment bias [21]. Simply put, this is a form of stereotyping that induces the physician to base its diagnosis on prior beliefs (related to, e.g., gender, race, etc.) instead of symptoms. For instance, a physician may be reluctant to diagnose depression to a man, because he believes that man cannot suffer from depression (which is obviously incorrect). In CLARION, this bias is a special case of similarity-based reasoning. The disease and its symptoms would all be represented by chunks in the top level of the NACS. Rules in the top level would associate symptoms with the disease. These chunks would also be represented by a set of features in the bottom level. Specifically, the features of the disease chunk would include the features of a typical patient diagnosed with this disease (e.g., a woman over 50 years-old). When the physician meets the patient, demographic information is collected and questions are asked about his/her symptoms. The symptoms activate symptom chunks in the top level of the NACS (for rule-based reasoning, see (1)), and the patient's description activates features in the bottom level of the NACS (i.e., for similarity-based reasoning, see (6)). These two sources of activation are propagated to the disease chunks and the results are integrated using the *Max* function (as in (3)). If β_2 is set to a high value (i.e., $\beta_2 >> \beta_1$), the disease chunk activation is going to be determined mostly by stereotyping, thus ignoring the patient's actual symptoms. Because the final diagnosis is based on the chunk disease activation, this provides an intuitive explanation for the ascertainment bias in medical diagnoses.

C. Discussion

This section described two example biases in medical science that can be accounted for by CLARION. Interestingly, these biases were explained using the very core assumptions of CLARION, namely the ACS control over the NACS processing and similarity-based processing in the NACS bottom level. While explaining these two biases does not require the full complexity of DFT, it is fully compatible with it. The unpacking principle was explained by the threshold on decision-making, which plays an important role in DFT [14]-[16]. The ascertainment bias was explained with similarity-based processing, which also plays an important role in DFT (for response inhibition [22]). Hence, explaining

these medical biases did not increase the complexity of the CLARION core theory included in this paper.

V. Conclusion

This article explored whether it is possible to reduce the complexity in a cognitive architecture while maintaining its generality. Here, we extracted the core theory of CLARION [4], [6]-[8] and showed that it was possible to explain cognitive/psychological phenomena in decision-making. These phenomena were mainly explained using the Non-Action-Centered Subsystem (NACS) in CLARION, and the free parameters were not used in most cases (most were parameter-free explanations, while in others only some general constraints were listed). Many other phenomena have also been explained with the core theory of CLARION [20]. This exercise in minimality is important, because cognitive architectures have generally avoided the question of model complexity by responding with generality criteria (e.g., [5]). On the other end of the spectrum, simpler cognitive/psychological models are usually applicable only to a very limited set of tasks, which can lead to the problem of complex tasks being explained by an aggregate of several simple models that are mutually incompatible. As such, this work is an important first step in bridging the gap between mathematically simple models and general cognitive architectures.

Appendix

Decision Field Theory (DFT) is a prominent theory in decision-making [14]-[16]. More specifically, DFT is a diffusion model that allows for the modeling of the decision process instead of only focusing on the end-state. DFT can be understood using two fundamental notions: valence and preference [15]. The *valance* of an option is the momentary advantage/disadvantage of an option in relation to the other options being considered. 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 economy can be used as attributes when considering buying options for a car [15].

The second component of valence is attentional weight allocation. At every time step, one might concentrate on a different attribute of the options. For instance, one might be thinking specifically about car quality at the present moment, and switch her attention to economy at the following time step. Attention allocation is represented by the vector $\mathbf{w}(t)$. This notation highlights the dependency of attentional allocation on time. By default, only one of the attributes is attended to at any moment and attention is switched either randomly or using a Markov process [15].[6] 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 weighed values of 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 economy attribute. Hence, a different weighed value would be assigned to the car at time t depending on which attribute received more attention.

The third and final component of valence is a comparison process that contrasts the weighed value of each option with the weighed values of 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:

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

where $\mathbf{v}(t)$ is the valence vector at time t (A1) and $\mathbf{S} = [s_{ij}]$ is a $n \times n$ inhibition matrix. The preference vector represents a n-dimensional random walk 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.[7]

From (A2), we see that 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 $\mathbf{S}$ matrix contains a set of 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 stronger options

[6] Note that attention can be distributed across attributes according to DFT, but this additional flexibility is rarely used.

[7] Note that a lower bound can also be used to eliminate options.

inhibit the weaker options. While the exact values assigned to the free parameters is not crucial, 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 each other 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 [22].

REFERENCES

[1] A. Newell, *Unified Theories of Cognition*. Cambridge: Harvard University Press, 1990.

[2] S. Roberts, and H. Pashler, "How persuasive is a good fit? A comment on theory testing," *Psychological Review*, vol. 107, pp. 358-367, 2000.

[3] J. R. Anderson, and C. Lebiere. *The Atomic Components of Thought*. Mahwah: Erlbaum, 1998.

[4] R. Sun, Duality of the Mind: A Bottom-Up Approach Toward Cognition. Mahwah: Lawrence Erlbaum Associates, 2002.

[5] R. Sun, "Desiderata for cognitive architectures," *Philosophical Psychology*, vol. 17, pp. 341-373, 2004.

[6] S. Helie and R. Sun, "Incubation, insight, and creative problem solving: A unified theory and a connectionist model," *Psychological Review*, vol. 117, pp. 994-1024, 2010.

[7] R. Sun, E. Merrill, and T. Peterson, "From implicit skills to explicit knowledge: A bottom-up model of skill learning," *Cognitive Science*, vol. 25, pp. 203-244, 2001.

[8] R. Sun, P. Slusarz, and C. Terry, "The interaction of the explicit and the implicit in skill learning: A dual-process approach," *Psychological Review*, vol. 112, pp. 159-192, 2005.

[9] R. Sun and X. Zhang, "Accounting for a variety of reasoning data within a cognitive architecture," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 18, pp. 169-191, 2006.

[10] S. Helie, R. Proulx, and B. Lefebvre, "Bottom-up learning of explicit knowledge using a Bayesian algorithm and a new Hebbian learning rule," *Neural Networks*, vol. 24, pp. 219-232, 2011.

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

[12] J. R. Anderson, *The Adaptive Character of Thought*. Hillsdale: Erlbaum, 1990.

[13] L. L. Thurstone, "A law of comparative judgement," *Psychological Review*, vol. 34, pp. 273-286, 1927.

[14] J. R. Busemeyer, and A. Diederich, "Survey of decision field theory," *Mathematical Social Sciences*, vol. 43, pp. 345-370, 2002.

[15] R. M. Roe, J. R. Busemeyer, and J. T. Townsend, J.T., "Multialternative decision field theory: A dynamic connectionist model of decision making," *Psychological Review*, vol. 108, pp. 370-392, 2001.

[16] J. R. Busemeyer, and J. G. Johnson, "Micro-process models of decision making," in *The Cambridge Handbook of Computational Psychology*, R. Sun, ed. Cambridge University Press, 2008, pp. 302-321.

[17] R. Sun, X. Zhang, and R. Mathews, "Capturing human data in a letter counting task: Accessibility and action-centeredness in representing cognitive skills," *Neural Networks*, vol. 22, pp. 15-29, 2009.

[18] C. Bundesen, T. Habekost, and S. Kyllingsbæk, "A neural theory of visual attention: Bridging cognition and neurophysiology," *Psychological Review,* vol. 112, pp. 291-328, 2005.

[19] R. Sun, Ed., *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*. Cambridge University Press, 2006.

[20] S. Helie, and R. Sun, "Accounting for memory, categorization,and reasoning phenomena using the CLARION cognitive architecture," unpublished.

[21] P. Croskerry, "The importance of cognitive errors in diagnosis and strategies to minimize them," *Academic Medecine*, vol. 78, pp. 775-780, 2003.

[22] M. Usher, and J. L. McClelland, "Loss aversion and inhibition in dynamical models of multialternative choice," *Psychological Review*, vol. 111, pp. 757-769, 2004.