Understanding Human Learning Using a Multi-agent Simulation of the Unified Learning Model

Vlad Chiriacescu, GC Image LLC, Lincoln, NE, USA

Leen-Kiat Soh, Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE, USA

Duane F. Shell, Department of Educational Psychology, University of Nebraska-Lincoln, Lincoln, NE, USA

ABSTRACT

Within cognitive science and cognitive informatics, computational modeling based on cognitive architectures has been an important approach to addressing questions of human cognition and learning. This paper reports on a multi-agent computational model based on the principles of the Unified Learning Model (ULM). Derived from a synthesis of neuroscience, cognitive science, psychology, and education, the ULM merges a statistical learning mechanism with a general learning architecture. Description of the single agent model and the multi-agent environment which translate the principles of the ULM into an integrated computational model is provided. Validation results from simulations with respect to human learning are presented. Simulation suitability for cognitive learning investigations is discussed. Multi-agent system performance results are presented. Findings support the ULM theory by documenting a viable computational simulation of the core ULM components of long-term memory, motivation, and working memory and the processes taking place among them. Implications for research into human learning, cognitive informatics, intelligent agent, and cognitive computing are presented.

Keywords: *Cognitive Modeling, Computational Simulation, Human Learning, Multi-Agent, Unified Learning Model (ULM)*

1. INTRODUCTION

Human learning in the sense of knowledge storage, exchange, and retrieval is an increasingly important topic in many areas of science. Fields such as neuroscience, cognitive science, psychology and education are engaged in the study of how humans acquire knowledge and develop skill and expertise. Modeling and understanding human learning is especially salient in the emerging fields of cognitive informatics (Wang, 2007; Wang et al., 2010; Wang, Widrow, et al., 2011) and cognitive computing (Wang, 2009a; Wang, 2011; Wang et al., 2010). *Cognitive informatics* is a transdisciplinary inquiry bringing together computer science, information sciences, cognitive science, and intelligence science to investigate and understand the in-

DOI: 10.4018/ijcini.2013100101

ternal information processing mechanisms and processes of the brain and natural intelligence (Wang, 2007). Learning is clearly central to this effort as most human thought and behavior that could be described as intelligent emerges from knowledge and behavior that was learned either directly or through experience (Wang, Kinsner, & Zhang, 2009). This learning is realized in the brain through neural plasticity which produces the micro-architecture of neuron connectivity (Kandel, Schwartz, & Jessell, 2000); Shell et al., 2010). A goal of cognitive informatics is to inform *cognitive computing*; the emerging paradigm of intelligent computing methodologies and systems based on cognitive informatics that attempts to implement computational intelligence by mimicking the mechanisms of the brain in cognitive computers (Wang, 2009a; Wang, 2011). Clearly, cognitive computers would benefit from being able to learn in ways similar to those which underlie neural plasticity.

Recently, an interdisciplinary team of researchers in psychology, education, and teaching published a comprehensive learning theory derived from a synthesis of research in cognitive neuroscience, cognitive science, and psychology: the Unified Learning Model or ULM (Shell et al., 2010). The ULM has begun to influence thinking and practice in fields such as scholarship of teaching and learning (Wilson-Doenges & Gurung, 2013), situated cognition (Durning & Artino, 2011), pedagogy (Nebesniak, 2012), and cognitive function (Wasserman, 2012).

Learning in ULM results from the interaction of three cognitive components: long-term memory, working memory, and motivation. Long-term memory (or LTM) is the relatively permanent store of knowledge possessed by a person. In the ULM, knowledge refers to the totality of what a person knows. This includes factual and conceptual knowledge sometimes referred to as declarative knowledge, cognitive and behavioral skills sometimes referred to as procedural knowledge, episodic knowledge of personal experience, and sensory or perceptual knowledge. Long-term memory for declarative and procedural knowledge resides in the cortex with procedural knowledge involving primarily the sensory-motor cortical regions and cerebellum. Sensory/perceptual, linguistic, and number knowledge generally resides in specialized modular processing areas (Kandel et al., 2000).

Working memory (or WM) is the term for the currently active part of cognition. Brain areas such as the forebrain and hippocampus have been implicated in working memory functioning (Kandel, Schwartz, & Jessell, 2000), however, working memory is better thought of as a process than an anatomical location. Two aspects of working memory affect learning. The first is capacity limitation, which is thought to be somewhere around 4-7 elements (Saults & Cowan, 2007). Elements, however, can be chunks, that increase functional working memory capacity. The second aspect is attention (Knudsen, 2007). Central to the ULM is the proposition that attention is a necessary precondition to learning. Only attended knowledge in working memory can add to or change knowledge in long-term memory.

The final ULM component is motivation. Motivation derives both from biological components like drives (e.g., hunger) and emotions and from cognitive components such as goals and beliefs (Schunk & Zimmeman, 2008; Shell et al., 2010). The ULM holds that these motivators are intimately connected to working memory and direct attention such that knowledge in working memory is attended only when there is motivation to attend to it.

Within long-term memory, connections between neurons are strengthened and weakened through neural plasticity that follows a Hebbian learning process (Kandel et al., 2000; Caporale & Dan, 2009). The basic ULM learning mechanism merges Hebbian neural plasticity with statistical learning. In the ULM, knowledge in long-term memory is built when distinct pieces of knowledge, either from sensory input or retrieved from long-term memory, that are held simultaneously in working memory are attended, connected, and stored as chunks in long-term memory. The connections in these chunks continue to strengthen or decay depending on repetition due to knowledge retrieval via pattern matching and spreading activation throughout the chunk. As with findings in neural studies (Caporale & Dan, 2009), this repetition causes knowledge chunks in long-term memory to ultimately reflect statistical regularities present in the knowledge being learned.

Within cognitive science, computational modeling has a long history as a method for testing theory about human cognition (McClelland, 2009). Although modeling cannot prove a theory, it can provide evidence that the theory is at least plausible (McClelland, 2009). The authors of the ULM argued that the core learning mechanisms of the ULM were potentially computational; but they did not derive a computational model in their work. The work reported here has been directed at creating a computational model of the ULM (called C-ULM) to test of the viability of the learning mechanisms proposed in the ULM.

We have developed a multi-agent-based simulation in which each single agent learns in accordance with the ULM model. Each single agent has a cognitive architecture that consists of the three main ULM components: long-term memory containing knowledge, working memory, and motivation. *Knowledge* in long-term memory is represented as an undirected, weighted graph where nodes indicate knowledge concepts and weighted edges—with a certainty measure on each weight—indicate a quantified connection between two concepts. *Motivation* is computed for each concept and is a function of the certainty that an agent has towards the weights for connections involving the analyzed concept and the value of the concept for solving a task. *Working memory* is the buffer that is filled with units of knowledge. Two types of units are considered: singleton concepts and concept chunks (i.e., a groups of connected concepts).

Within the C-ULM, we have also articulated several aspects of learning and teaching both conceptually and algorithmically. First, agent communication is grounded on the actions of *teaching* and *learning* and has at its core, algorithms that perform the processes of (1) allocating working memory for teaching and learning and (2) using the working memory content to update the knowledge of a learner or a teacher. Second, a feature of the learning process is represented by the *spread activation factor*, which guides how the certainty for the weights of all connections reachable from a starting connection is to be updated. The amount of change in certainty for a connection is inversely proportional to the distance between this connection and the starting connection. Third, in C-ULM, *knowledge decay* (or, simply put, forgetting) is triggered when connections do not enter working memory for a given number of simulation time steps. The decay consists in increasing the uncertainty for the involved connection weights. Fourth, agent behavior is problem solving based and directed at completing *tasks* which require specific patterns of knowledge connections. Agents must possess the requisite connected knowledge with adequate certainty to attain a task.

Our contributions can be considered from two perspectives. From the *cognitive informatics and cognitive modeling perspectives*, C-ULM advances the literature by providing the first computational simulation of learning that incorporates the ULM components of long-term memory, working memory, motivation and the relationships among them into an operative modeling framework. The C-ULM incorporates the more sophisticated ULM learning processes that are more closely tied to human neural learning than current reinforcement learning (Kawato, & Samejima, 2007), back propagation (McClelland, 2009), and Bayesian methods (Goodman, Ullman, & Tenenbaum, 2011). From the *multi-agent cognitive computing perspective*, C-ULM could benefit cognitive computing research and development at two levels. First, the intelligence of individual agent reasoning can potentially be improved by the incorporation of the learning functions and relationships among long-term memory, motivation and working memory represented in the C-ULM. Second, C-ULM can allow incorporation of human teaching and learning processes into agent-to-agent knowledge transfer leading to more efficient agent learning and human-computer interactions.

Note that this paper is an extension of a previously published conference paper with the same title (Chiriacescu, Soh, & Shell, 2013).

2. RELATED WORK

One particularly relevant work in cognitive informatics is that by Tian, Wang, Gavrilova, and Ruhe (2011). They describe and propose a formal knowledge representation system (FKRS) based on the object-attribute-relation (OAR) model and its concept algebra (Wang, Tian & Hu, 2011). It uses as a linguistic base the well-known WordNet and is comprised of three main components: concept formation, conceptual knowledge representation and knowledge visualization. FKRS and OAR are examples of semantic level symbolic models (McClelland, 2009). They model knowledge in linguistic and language terms. The C-ULM operates at a level more similar to a connectionist model. The learning processes of the ULM that are modeled in C-ULM are not language or symbol based. They reflect statistical Hebbian neural learning process. These are more elemental than symbolic language. As discussed by McClelland (2009), these approaches differ but are complementary rather than antagonistic.

The FKRS can prove helpful in obtaining a more structured representation of the knowledge that is being learned. The ULM argues that knowledge in the brain comes to reflect statistical regularities in the information being learned. FKRS provides a rigorous description of the properties of concepts. This could provide guidance as to what statistical regularities exist in the knowledge by describing specific attributes and objects pertaining to a given concept. An important connection can be established between the OAR model and the C-ULM knowledge representation. In the OAR model, there are networks of objects, attributes and relation that connect objects and attributes forming networks of objects and attributes. Of note, those objects and attributes are seen as partially connected (and not fully connected) in a similar fashion as knowledge is represented in C-ULM. Thus, the C-ULM concepts could correspond to OAR's objects and the relations between them represented by C-ULM's connections. Furthermore, C-ULM allows for a large variety of relations given the relative connection strength indicated by the connection weight value. As future work, attributes can be incorporated within C-ULM concepts or as an alternative, concepts can represent attributes that form specific chunks that in turn represent corresponding OAR objects.

Another important cognitive informatics connection can be made between the C-ULM architecture and the layered reference model of the brain (LRMB) (Wang & Chiew, 2010; Wang, Wang, Patel, & Patel, 2006). The LRMB is a formal, layered model of cognitive processes in the brain. In this model, the brain has 7 seven abstraction layers of processes with primitive processes operating at the sub-conscious level and higher cognitive functions such as learning, problem solving and decision making operating at the conscious level and relying on the mechanisms of previous levels. The distinctions between sub-conscious and conscious levels mirror other recent formulations such as Kahneman's (2011) System 1 and System 2. The LRMB is a process oriented model. The ULM (Shell et al., 2010) is a knowledge oriented model. In the ULM, all process distinctions are seen as distinctions in knowledge with knowledge including all forms of data contained in the brain from sensory information to higher-order skills. Although the ULM recognizes that different brain areas, such as sensory memory modules or the motor cortex, have different outputs similar to the abstraction layers of the LRMB, the ULM holds that within the range of what that particular area is capable of outputting, its outputs are the results of neural plasticity learned via the ULM principles. From the perspective of the ULM, the distinctions represented in the LRMB reflect differences in the types of knowledge that different parts of the brain/cognitive system are encoding. Sensory memory modules are encoding statistical regularities in low level data associated with the sense. Language modules are encoding statistical regularities in the language. The functional model of the LRMB reflects a general information processing approach to cognition. The ULM shares this approach. However, the ULM merges the LRMB functions of short-term memory and natural intelligence (NI-OS and NI-APP) into a single working memory consistent with much recent thinking (Saults & Cowan, 2007). The ULM also merges all sensory, motor, and general cognitive functions into a single long-term memory. This makes the C-ULM a much simpler computational model than LRMB. It may be that the observable outputs of the natural intelligence of the brain are better modeled by something like the LRMB and the acquisition of the knowledge that produces that intelligence is better modeled by something like C-ULM. Whether this is a fruitful approach needs to be established in future research.

Because the C-ULM architecture reflects these ULM consolidations of knowledge and working memory, many LRMB levels and processes are represented within the C-ULM. For example, Layer 1, *Sensation*, is represented by concepts received by a learning agent in C-ULM. Those stimuli enter the second layer through the short-term memory (STM), which is akin to the working memory in C-ULM. Layer 4, *Perception*, has two important modules: attention and emotions. The first module, attention is modeled within C-ULM by the use of the awareness threshold that filters what enters into short term-memory. The second module, emotions, is modeled to a certain degree in C-ULM by the motivation concept and motivation scores for concepts. Furthermore, as meta-cognition processes, we model the search module of Layer 5 (*Meta-Cognition*) when we do breadth-first search to find the appropriate concepts that will be retrieved for teaching or updated for learning. The *memorize* module of Layer 5 is further characteristically represented by the acquisition of new connections and also by the update of connection weights in C-ULM. Furthermore, the C-ULM's chunking process—an important process in ULM—leads to an ever increasing efficientization of the way STM is being used in the learning process. A chunk represents a network of concepts that are more related to each other than to other concepts. From a knowledge representation point of view, the chunk is a higher, more abstract level of knowledge that is a synthesis of individual concepts. Thus the C-ULM's concept of chunking can be related to the LRMB's modules of Abstraction and Synthesis found at Layer 5 (Meta-cognition) and Layer 6 (Meta-inference). C-ULM also models the interaction happening at the top LRMB layer, between the learning and the problem solving processes. Thus, more learning steps enhance problem solving and in turn, solved problems lead to new learning experiences (coming from the knowledge obtained by solving the task).

There are additional parallels between C-ULM and the LRMB based problem solving model proposed by Wang and Chiew (2010). Within C-ULM, problem solving happens through the process of attempting and solving a task. Just as in Wang and Chiew (2010), solving a problem requires a set of representation and search operations. Within C-ULM, the representation operations are those operations that alter the long-term memory (LTM) structure of an agent (acquiring new connections and in the latest version, also pruning extremely unused connections). On the other hand, the search operations are those operations that, taking into account agent knowledge but also task feedback update both the LTM structure and connection weight values. These series of structure and weight updates are essentially searching through the problem space in order to find the suitable configuration of connections and weights that leads to solving the task.

In relation to the cognitive informatics model of memorization proposed by Wang (2009b), the C-ULM shares a focus on repetition and connection or relation as the primary learning processes. As noted previously, the OAR model that Wang uses operates at a symbolic level and the C-ULM is a statistical based model. Also, the C-ULM in merging short-term memory into a more general working memory and merging various Sensory Buffer Memory (SBM), Conscious-Status Memory (CSM), Long-Term Memory (LTM), and Action-Buffer Memory (ABM) from Wang into a single Long-Term Memory. Wang's memorization model is intended to apply to one specific type of cognitive process from the LRMB model. The C-ULM is meant to apply to all learning of all of the knowledge included in the LRMB model, making C-ULM a more general statement of how knowledge is acquired across all brain and cognitive components.

Recent work in cognitive informatics has focused on motivational regulators that perform roles similar to C-ULM motivators. Rosales, Jaime, and Ramos (2013) introduced an emotional regulation model having two main components, i.e., emotional response and emotional regulation. When the virtual agents respond to a risk situation, their emotions could influence the decision-making process adversely. The emotional regulation process helps them to ignore, regulate or use their emotions. The regulation component consists of two modules—namely, a reappraisal module and a suppression module. When a virtual agent's average of perceived behavior and required behavior is the same as the expressed behavior indicating "emotional response", the suppression algorithm basically switches a virtual agent's attention and ignores the highly affective objects—where each object has an emotional memory, elicited in the agent that stored the object in the first place, for example—in the scene.

Cervantes et al. (2013) introduced a moral decision making (MDM) model for agents based on ethical, moral, and religious principles as well as on individuals' beliefs of right and wrong, feelings, and emotions. The computational process of this model consists of 3 phases: (1) assessment of options including filtering using a set of moral and ethical rules based on experiences, prejudices, emotions, cost-benefit analysis and moral evaluation, (2) execution of the selected option by which it is sent to the working memory and new execution plans are generated in a planning process, and (3) outcome evaluation where the executed actions are further evaluated. This MDM model provides a potential set of additional motivational considerations that could be incorporated into C-ULM. Clearly, human teaching and learning have moral and ethical dimensions. Learning and teaching of C-ULM could consider moral and ethical rules in decisions about what to teach and what not to teach, or what to learn and what not to learn. The above 3-step computational process could potentially inform C-ULM in deciding what learning and teaching tasks to perform, evaluating the outcomes, and reinforcing the decision. C-ULM only considers the knowledge being shared in a teaching interaction and the knowledge required for task completion.

In the ULM, Shell et al. (2010) propose that all motivators impact learning via motivation and attention direction in working memory. Other processes like morals, ethics, and emotions clearly impact human behavior including learning. Currently, C-ULM only models two of these motivators: self-efficacy and expectancy/task reward. These were chosen because they have consistently been found to be among the strongest motivators in prior studies (Schunk & Zimmerman, 2008; Shell et al., 2010). Also, as discussed in Shell et al. (2010), self-efficacy and expectancy/task reward have the most clear neurological foundations of the available motivational constructs. But, future work needs to expand the scope of motivational influences to include the types of moral and emotional factors noted by Cervantes et al. (2013) and Rosales et al. (2013).

Within the cognitive modeling domain, a number of computational models have been published in the last few years that integrate one or two of the three main ULM components. One of those works (Jones, Gobet, & Pine, 2008) focuses on children's developmental change that occurs by increases in long-term knowledge and working memory capacity. The Elementary Perceiver and Memorizer-Vocabulary (EPAM-VOC) is a phoneme sequence learner that takes speech in phonemic form as input and builds a hierarchical network of phoneme sequences (or "chunks") that represents long-term knowledge of the linguistic input. Learning in this model is performed by constructing directed graphs where each arrow indicates additional information that is added to the content of the source node in order to derive the content of the destination node. The model is useful in assessing the individual influence of long-term knowledge and working memory increases in child development. As compared to this model, the C-ULM also incorporates the motivation component thus obtaining a more integrative model of human knowledge evolution and exchange. Furthermore, C-ULM uses a knowledge graph that is weighted, thus enabling the representation of concepts with a varying degree of relatedness.

Another recent computational model focuses on achievement motivation for artificial agents (Merrick, 2011). It relies on Atkinson's Risk-Taking Model (RTM) and is shown to exhibit similar goal selection features to humans. In this model, the motivation to approach a task grows stronger as the probability for succeeding at the task increases. As compared to this model, the C-ULM motivation component is based on two factors: (1) an intrinsic factor that relates motivation directly to the notion of knowledge by the use of a certainty measure on each connection weight and (2) an extrinsic factor that ties motivation to the reward-based feedback obtained from solving tasks.

In C-ULM, the agent learning results in long-term memory updates that consist of changes in the connection weights and the certainty measures associated to those weights. Similar to our certainty measure update formula is the delta-rule used in Ramscar and Yarlett (2007) for updating the association strength between the semantics and phonology of a noun item. Of note, the mentioned work includes in the update amount for association strength a spread activation parameter *s* that resembles the spread activation factor that C-ULM uses in updating long-term memory certainty measures. In contrast to this work, C-ULM also includes a motivation related factor in the update formula for association strength between two concepts.

From a cognitive-theoretic viewpoint we are supporting the idea emphasized in Chater and Brown (2008) that a combination of rather simple but general cognition principles could explain apparently complex mental phenomena (such as the mental process of learning to solve complex tasks). In the case of C-ULM, these principles involve a relatively simple cognitive architecture of three primary components and application of statistical learning mechanisms.

Within the modeling (Kawato & Samejima, 2007) and multi-agent systems (Watkins, & Dayan, 1992) fields, one of the widely used paradigms is the reinforcement learning (RL) approach. One of the most important aspects of RL algorithms is the trade-off between exploration of unknown territory and exploitation of current knowledge. In the C-ULM, this trade-off is mainly exhibited by tuning the certainty measure associated to each knowledge weight through the complementary processes of learning and knowledge decay. The RL-inspired balance between exploration and exploitation is also used in the C-ULM through the process of task feedback—if an agent solves a task, the certainty measures associated to the involved knowledge connections are updated similar to the learning process (the agent learned how to solve the task); if an agent fails to solve a task, associated certainty measures are updated similar to the forgetting process (the agent starts to forget ways of attempting the task that proved unsuccessful).

Finally, although the C-ULM is based on neurological principles as described in the ULM, it is not proposed as a direct computational model or simulation of the brain or neural functions such as the Spaun project (Eliasmith et al., 2012). The C-ULM, however, is meant to be more faithful to the principles reflected in neural plasticity than a project such as Watson (Ferrucci et al., 2010). Although Watson incorporates some ULM ideas such as long-term memory, working memory, confidence, probabilistic retrieval, and motivation, Watson is not meant to model how these components work in humans. Importantly, while Watson does make new knowledge connections, those connections are created within its long-term memory; Watson does not learn or acquire its initial long-term memory knowledge; it only reconnects already present knowledge.

3. AGENT MODEL AND MULTIAGENT FRAMEWORK

In this section we present the single agent model and the multi-agent environment used in the C-ULM simulation, showing how we "translate" the ULM into an integrated computational model. In section 3.1 we present the three components, learning principles, and learning processes as they are outlined by the Unified Learning Model. The single-agent model and the relationships between long-term memory knowledge, motivation and working memory are described in section 3.2. In section 3.3 we focus on the interactions that take place among agents, i.e., the actions of teaching and learning. Finally, section 3.4 presents agent tasks and the interaction taking place between an agent and a task.

3.1. Unified Learning Model (ULM)

Central to the Unified Learning Model (ULM) is the idea that all learning takes place in three primary components: (1) *long-term memory* which contains long-term knowledge, (2) *working memory* (WM) which receives knowledge retrieved from long-term memory and processes incoming sensory input, and (3) *motivation* which directs the agent's attention within working memory. These components encompass the basic cognitive architecture of the C-ULM computational model. The interactions between these components reflect the ULM's three principles of learning: (1) *Learning is a product of working memory allocation*; (2) *Working memory's capacity for allocation is affected by prior knowledge (chunking)*; and (3) *Working memory allocation is directed by motivation*. Operations within the architecture follow three ULM learning processes: (1) *New learning requires attention*; (2) *Learning requires repetition*; and (3) *Learning is about connections*.

Taken together, these three learning processes operating within the architecture of the ULM are sufficient for creating a complete computational model of learning that generates a detailed information flow in each individual agent and in the multi-agent system as a whole. The following subsection describes in detail the computational adaptation for each of the three primary architectural components.

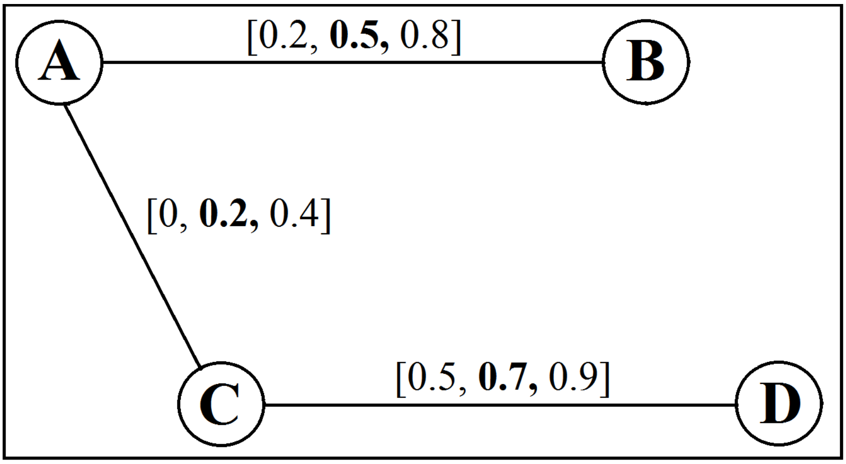
3.2. Single-Agent Model

3.2.1. Long-Term Memory

Long-term memory is modeled as an undirected, weighted graph where nodes represent knowledge concepts and weighted edges represent a quantified connection between two concepts. Initially, agents do not have the necessary knowledge to solve a task but in some cases they might have a 'vague idea' of how to solve the problem. Key to modeling of the knowledge component is measuring the vagueness for each particular edge weight. This is realized by assigning a certainty measure called *confusion interval* to each edge weight. This interval is bounded and its *length* indicates how certain is the agent regarding the associated weight. For example, if the length is very small, the agent is quite certain about the weight of the edge and it has a solid knowledge about it. When an agent has to solve a task or teach another agent about a given connection weight, the agent will use a weight randomly generated from the associated confusion interval. The *center* of this confusion interval is also the edge weight.

Figure 1 presents an example of an agent's LTM. Next to each LTM connection is the confusion interval corresponding to that connection. The second value (bolded in Figure 1) in the confusion interval represents the interval center (or midpoint) and the edge weight. The other two values represent the minimum and the maximum values of the confusion interval. The lower bound on the minimum value is 0 and the upper bound on the maximum value is 1. As discussed later in this section, both the edge weight and the length of this interval are updated during the learning process (Equations (2), (4) and (7)). Specifically, the edge weight can move in both directions, towards 0 or 1. The length of the confusion interval is

Figure 1. LTM with concepts A, B, C, D. On each edge is outlined the associated confusion interval.



shortened by the learning process (Equation (2)) and it is increased by the process of knowledge decay (Equation (7)). The confusion interval instantiates the statistical learning inherent in the ULM learning process of repetition. As in Hebbian learning for neural synapses, LTM connections in C-ULM strengthen with repetition and weaken (decay) with disuse.

3.2.2. Motivation

We use the notion of *motivational scores* to model the motivational component of the architecture. Each concept found in agent LTM has a motivational score associated with it. A higher score reflects a higher motivation for teaching or learning about the associated concept while a lower score indicates a lower motivation related to that concept. This score is a function of: 1) the underlying confusion intervals for the connections that contain the concept, and 2) the expected rewards for the tasks that use the concept, as shown in Equation (1):

$$m_X^{A(t)} = \sum_{Y \in SC_X} \left(\frac{1}{l_{XY}^{A(t)}} \right) \cdot \sum_{k \in T_X} \left(R_k \right) \quad (1)$$

where X is a concept in agent A's LTM; $m_X^{A(t)}$ is the agent A's motivational score for concept X at time step t; SC_X is the set of concepts connected to concept X; XY is the edge connecting concepts X and Y; $l_{XY}^{A(t)}$ is the length of agent A's confusion interval for edge XY at time step t; T_X is the subset of tasks that require concept X; and R_k is the reward for task k. The rationale behind this formula is to allow two types of motivators that exist at the architectural level of ULM (Shell et al., 2010): an intrinsic one that captures the notion of self-efficacy, i.e., length of confusion intervals, and an extrinsic one similar to reinforcement learning (Watkins & Dayan, 1992) that assesses the expectancy of possible rewards available when using the concept for solving tasks.

3.2.3. Working Memory (WM)

Similar to the LTM component, WM is also represented using a weighted graph. The difference is that it has a *capacity* which indicates the maximum number of concepts (or knowledge chunks) allowed in the WM graph. WM allocation is part of the learning and teaching actions and thus is a part of the agent communication protocol. In order to realize WM allocation, we introduce the concept of *awareness threshold* (*AT*). This threshold indicates how aware the agent is of external and internal stimuli. If a stimulus has an intensity that is higher than this threshold, the agent becomes aware of that stimulus and consequently it allocates a WM slot for that stimulus. In our modeling, the concepts are the stimuli, and the motivational scores represent the stimulus intensity for the associated concept. Thus, the awareness threshold dictates what is attended, within the general architectural principle that motivation directs WM allocation.

3.2.4. LTM Update and Spread Activation

After WM is allocated, the WM content indicates how to update the long-term memory of a learning or teaching agent, based on the statistical learning principles embodied in the ULM learning process of repetition. In the case of a learning agent, this step updates both the confusion interval centers of LTM connections corresponding to WM connections and the confusion interval length of the same connections. In the case of a teaching agent, only the confusion interval length is updated since a teaching agent only reinforces its existing knowledge *without* receiving new information about the task weights. The formula for ***updating a learning agent's confusion interval center*** is given by Equation (2):

$$w_{XY}^{L(t)} = \frac{cic \cdot \left[f\left(X, WM_L\right) \cdot m_X^{L(t)} + f\left(Y, WM_L\right) \cdot m_Y^{L(t)} \right] \cdot w_{XY}^{T(t)} + w_{XY}^{L(t-1)}}{cic \cdot \left[f\left(X, WM_L\right) \cdot m_X^{L(t)} + f\left(Y, WM_L\right) \cdot m_Y^{L(t)} \right] + 1} \tag{2}$$

where $w_{XY}^{L(t)}$ and $w_{XY}^{L(t-1)}$ are the learning agent confusion interval centers for edge XY during simulation time steps t and $t-1$, respectively; $m_X^{L(t)}$ and $m_Y^{L(t)}$ are the learning agent's motivational scores for concepts X and Y at time step *t*; $w_{XY}^{T(t)}$ is the instantiated weight value for edge XY communicated by the teacher via a weighted sub-graph at time step *t*; *cic* is a learning coefficient that influences how much the confusion interval's center moves towards the weight communicated by the teacher ($w_{XY}^{T(t)}$) and f is a function that returns 0 or 1 based on whether the given concept is currently present in the given WM. Function *f* is described by Equation (3) below:

$$f\left(Z, WM\right) = \begin{cases} 0, & Z \notin WM \\ 1, & Z \in WM \end{cases} \tag{3}$$

The mechanism for *updating a learning or teaching agent's confusion interval length* for a given connection *x* is given by Equations (4), (5) and (6):

$$l_x^{A(t)} = l_x^{A(t-1)} - sf \cdot mf \cdot cil \tag{4}$$

$$sf = 1 - \frac{d\left(c,\ x\right)}{D} \tag{5}$$

$$mf = f\left(X, WM\right) \cdot \left(m_X - AT\right) + f\left(Y, WM\right) \cdot \left(m_Y - AT\right) \tag{6}$$

where $l_x^{A(t)}$ and $l_x^{A(t-1)}$ are the confusion interval lengths for agent's *A* connection *x* (connected by a graph path to connection *c*) at time steps *t* and *t-1* respectively; sf is the *spread factor* (defined by Equation (5)); mf is the *motivation factor* (defined by Equation (6)); cil is a learning coefficient that influences the change in the confusion interval length during a simulation time step; $d\left(c,\ x\right)$ is the graph distance from connection *c* existent in both agent WM *and* LTM to a connection *x* existent *only* in the agent LTM; D is a *normalization factor* considered to be the upper-bound on the distance between a pair of connections in the LTM graph—that is, any distance greater than this value is set to *D*; m_X and m_Y are the motivational scores for concepts X and Y, respectively; f is the WM presence function defined by Equation (3); and AT is the awareness threshold for the learner.

These equations implement a statistical learning algorithm where both the connection center and confusion interval are repeatedly updated. As noted in the ULM (Shell et al., 2010), by virtue of the law of large numbers, this repetitive update process should lead to convergence on the actual weights of the task connections available in the environment of the simulation.

Additionally, we instantiate spreading activation which is an architectural component that results from the associative nature of human knowledge (Anderson, 1983). Spreading activation says that if a concept is activated, then this activation spreads to any connected concept. Furthermore, the activation of all connected concepts is smaller and it decreases with the distance from the initial concept. In C-ULM (Equations (4) and (5)), the update made to the confusion interval length of connection *x* reachable from connection *c* decreases as the updated connection *x* is farther from connection *c*.

3.2.5. Knowledge Decay

The ULM learning process of repetition says that repeated connections are strengthened but that non-repeated connections weaken. To accomplish this, we use a statistical learning algorithm that weakens long-term knowledge through decay. If a concept does not enter WM for a specified number of time steps, the concept is considered unused and the associated confusion intervals of all connections involving that concept are increased. The *knowledge decay mechanism for updating an agent's confusion interval length* for a connection involving an unused concept is given by Equation (7):

$$l_{XY}^{A(t)} = \begin{cases} l_{XY}^{A(t-1)} \cdot e^{r_{dec}}, u_X < u_X^{(t)} \le DF \cdot u_X \\ l_{XY}^{A(t-1)}, u_X^{(t)} \le u_X \; or \; u_X^{(t)} > DF \cdot u_X \end{cases} \quad (7)$$

where X is the unused concept, Y is a concept (used or unused) connected to concept X, $l_{XY}^{A(t)}$ and $l_{XY}^{A(t-1)}$ are the confusion interval lengths for agent's A connection XY at time steps t and $t-1$, respectively; e is the natural number; r_{dec} is the knowledge decay rate (i.e. the rate at which the confusion interval grows) and is an experimental parameter set to a constant value (between 0 and 1); u_X indicates how many time steps concept X can remain unused without triggering knowledge decay for connections involving X; $u_X^{(t)}$ is the number of time steps that concept X has been unused for at time t; $DF \cdot u_X$ is an upper-bound on the number of time steps for which knowledge decay is applied to connections involving concept X; and DF is a decay multiplication factor.

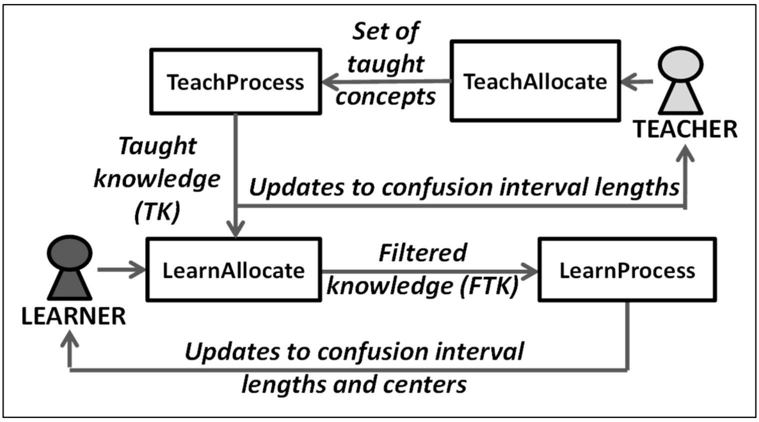
3.3. Multiagent Framework

In this section we present the agent communication and interaction protocol consisting of the actions of teaching and learning as illustrated in Figure 2. In this protocol, first, the teacher agent selects the concepts to be taught and allocates its WM for them. The concept selection process is done by the algorithm *TeachAllocate*. Then, the teacher agent produces the knowledge *TK* to be taught using *TeachProcess*. This has two effects. First, the teacher agent itself learns from the teaching as well. Thus, this leads to a shortening of confusion intervals for the connections in teacher's LTM that correspond to the connections found in *TK*. Second, correspondingly, the learner agent performs the algorithm *LearnAllocate* in order to filter the taught knowledge *TK*. The "filtered" *TK* (or *FTK*) resides in the WM of the learner agent. The learner agent then proceeds to perform *LearnProcess*, which updates the confusion interval lengths and centers according to the LTM update process described earlier in Section 3.2.

3.3.1. Teaching

TeachAllocate has two versions: *TeachAllocate-Basic* and *TeachAllocateChunking*. TeachAllocateBasic makes sure that the concepts with the highest motivation scores for the teacher will be the ones that are being taught. First, it sorts in descending order all the concepts in teacher agent's LTM by their motivation scores. Then it loops through the sorted concepts and adds all connected concepts to a concept list. The loop stops when the size of the list reaches the teacher

Figure 2. Communication protocol between a teacher and a learner agent



agent's WM capacity. Of note is that it does not add isolated concepts—concepts without even a single connection—to the concept list. The reason for this exclusion is that those concepts do not contribute with any connections to the teaching process. The concept list serves as an input to the TeachProcess algorithm.

In the TeachAllocateChunking version, the algorithm does not allocate just one concept to each WM slot but instead *allocates an entire chunk*. That is, given each top concept in the sorted list during the loop, it uses a breadth-first search (BFS) to identify the knowledge chunk for that concept in the teacher's LTM and then allocates it to the WM. Similarly, if the number of chunks is greater than the number of WM slots, we break out of the loop and the algorithm terminates.

The algorithm TeachProcess updates the confusion intervals of LTM connections that are used in teaching and creates the knowledge sub-graph that is the product of teaching. This sub-graph is "sent" to the learner and a part of it will fill the learner's WM. It loops through every connection formed with concepts found in the TeachAllocate concept list. If the two concepts are connected in teacher agent's LTM, the algorithm creates the corresponding edge in the taught sub-graph *TK*. Furthermore, it updates the confusion interval in the teacher agent's LTM. In order to compute the weight of connections that make up the taught graph *TK*, it picks up a uniformly generated random value from the teacher agent's confusion interval associated with the corresponding LTM connection. Of note here is that, in contrast to agent LTM graphs, the resulting taught graph *TK* is a weighted graph with *no* confusion intervals associated.

3.3.2. Learning

Similar to TeachAllocate, the algorithm LearnAllocate has two versions: *LearnAllocate-Basic* and *LearnAllocateChunking*. Mirroring TeachAllocateBasic, LearnAllocateBasic is used to ensure that taught concepts with a motivation score higher than the awareness threshold *AT* enter the WM of the learning agent. Again, it sorts all connections in the taught knowledge graph *TK* and then loops through the sorted connection list. At each iteration of the loop it also checks whether the number of concepts added to WM is greater than the number of WM slots. If it is, it breaks out of the loop and the algorithm terminates. Otherwise, it proceeds to check whether at least one concept of the currently analyzed connection has a motivation score greater than *AT*. If this condition is met, it adds the current connection to the WM graph. The resulting graph represents the filtered knowledge (*FTK*) mentioned in Figure 2.

Like TeachAllocateChunking, the algorithm LearnAllocateChunking allocates an entire *chunk* to a WM slot instead of just a concept. If the number of knowledge chunks is greater than the number of WM slots it breaks out of the loop and terminates.

LearnProcess performs the learning mechanism given the concepts found in the WM graph. It updates the confusion interval centers of all LTM connections corresponding to WM connections according to Equation 2 and then updates the confusion interval lengths of those connections according to Equation 4. Furthermore, it also updates the confusion interval lengths for LTM connections that have no corresponding WM connection but are connected to such LTM connections.

3.3.3. Chunking

Chunking is a basic mechanism of human memory reflecting the interconnected nature of neural structure (Shell et al., 2010). As such, in the ULM, it is an essential component of the learning process. Thus, the algorithms TeachAllocateChunking and LearnAllocateChunking in C-ULM implement the chunking mechanism. This allows us to model and test the impact of this aspect of human brain processing within the constraints of WM capacity limits.

3.4. Agent Tasks

Similar to agent LTM, a task is represented by a weighted graph consisting of nodes that represent knowledge concepts and edges that represent the connections between those concepts. In contrast to agent LTM, these connections do *not* have an associated "confusion interval". Each connection weight of a given task has to be matched within a certain margin of error by agent weights so that the agent successfully solves the task.

3.4.1. Task Attempt

Attempting a task in the C-ULM is a 3-step process. First, the algorithm checks for a structural match between agent LTM and the attempted task, i.e., all task connections have to exist in the agent's LTM. If they do, it then checks if there is enough WM for processing the task. This is done by counting the number of task chunks with the BFS algorithm and comparing this number with the WM capacity. If there is enough WM, it proceeds to the final step and checks for a weight match between the agent LTM and the task. In order to check for this type of match, the process uses uniformly generated random values from the confusion intervals of agent LTM connections corresponding to the task required connections. If all the differences between those random values and the associated task required weights are below an error margin threshold, then the task is considered solved. Otherwise, or if there is insufficient WM, the agent failed to solve the task.

3.4.2. Task Feedback

A reinforcement learning feature that we have incorporated into the overall task solving process is the task feedback. If an agent solved a task, the weight centers for the agent's LTM connections corresponding to the task connections are set to the weight values randomly picked from the associated confusion intervals and all confusion interval lengths are set to smaller values. This signifies that the agent has reached a higher level of confidence in its long-term knowledge about the connections involved in the solved task. In a similar fashion, humans also learn from accomplishing specific tasks, not only from what they are being taught by others (Shell et al., 2010; Wang et al., 2009). Correspondingly, if an agent failed to solve a task, the confusion interval lengths of the involved connections are increased. Similarly, after failing to accomplish a specific task, a person might explore other options of solving it (Shell et al., 2010; Wang & Chiew, 2010). In C-ULM, this exploration for solutions is increased by the increase of confusion interval lengths. Thus, in a way, the "rewards" for solving or failing tasks are integrated into an agent's reasoning process as "self-efficacy"—confidence in what the agent knows, as in the shortening or lengthening of confusion intervals.

4. IMPLEMENTATION

Our C-ULM simulation is built using Repast (North, Collier, & Vos, 2006). We use a time-stepped simulation execution model and each simulation run is defined by a set of parameters that consists of the number of agents, tasks and concepts existent in the environment, the agent WM capacity, the normalization factor *D*, the number of simulation time steps, and the Repast random seed value. For parallel execution of simulations, we use a cluster-based supercomputer called Tusker. Tusker is a 40 TF cluster consisting of 106 Dell R815 nodes using AMD 6272 2.1GHz processors, connected via Mellanox Quad Data Rate Infiniband and backed by approximately 350 TB of Terascala Lustre-based parallel filesystem. In order to run multiple simulations in parallel, we divided the parameter file into multiple files each of which containing a subset of the initial set of parameter configurations. Then we ran the simulation with a different parameter file for each Tusker node being used. Table 1 shows the simulation parameters used.

5. DISCUSSION OF RESULTS

In this section we present some of our results, discuss the validity and utility of the C-ULM simulation and present the implications for ULM as a theory for understanding human learning and also the implications for intelligent agent research. All figures in this section (Figures 3 – 7) present a simulation with the following characteristics: 20 agents in the multi-agent system, working memory capacity is from 3 to 7, existing tasks have at most 30 concepts, spread activation factor *D* is 5 and the chunking mechanism is used.

5.1. Validity of the C-ULM Simulation

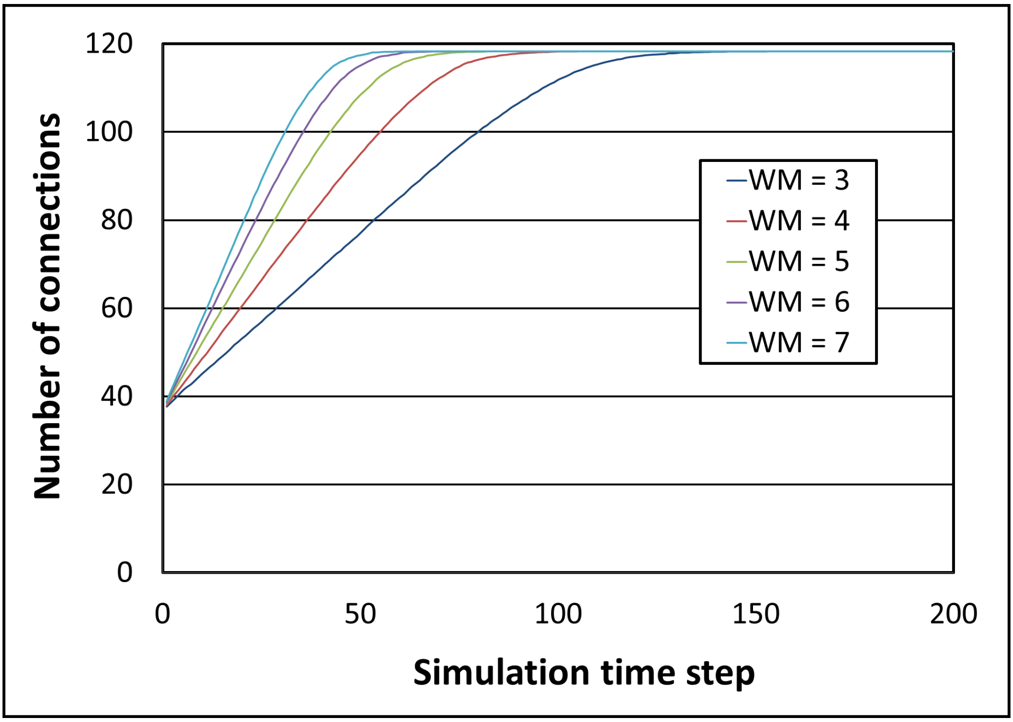
Our central research question was whether an operative computational simulation model could be created based on the ULM principles. Our answer to this question is yes. The C-ULM simulation parameters described previously have high fidelity to the principles and mechanisms described in the ULM. The next question is whether the C-ULM accurately reflects what is known about human learning.

To address this, we highlight two validations of the C-ULM simulation in Figures 3 and 4. A basic threshold for acceptance of the

Table 1. Simulation parameters

Simulation parameters	Range of values
Working memory (WM) capacity	3 – 9
Motivation factor (*mf*)	Strictly positive
Spread normalization factor (*D*)	1, 2, 3, 4, 5
Learning coefficient for the confusion interval length (*cil*)	0 – 0.01
Learning coefficient for the confusion interval center (*cic*)	0.8 – 1.2
Awareness threshold (*AT*)	0 – 1
Number of concepts	10, 30, 50, 100
Number of agents	10, 20
Number of tasks	3, 10, 30, 50
Confusion interval update amount for failed task attempt feedback (*FF*)	0.15 (constant)
Confusion interval for successful task attempt feedback (*SF*)	0.005 (constant)

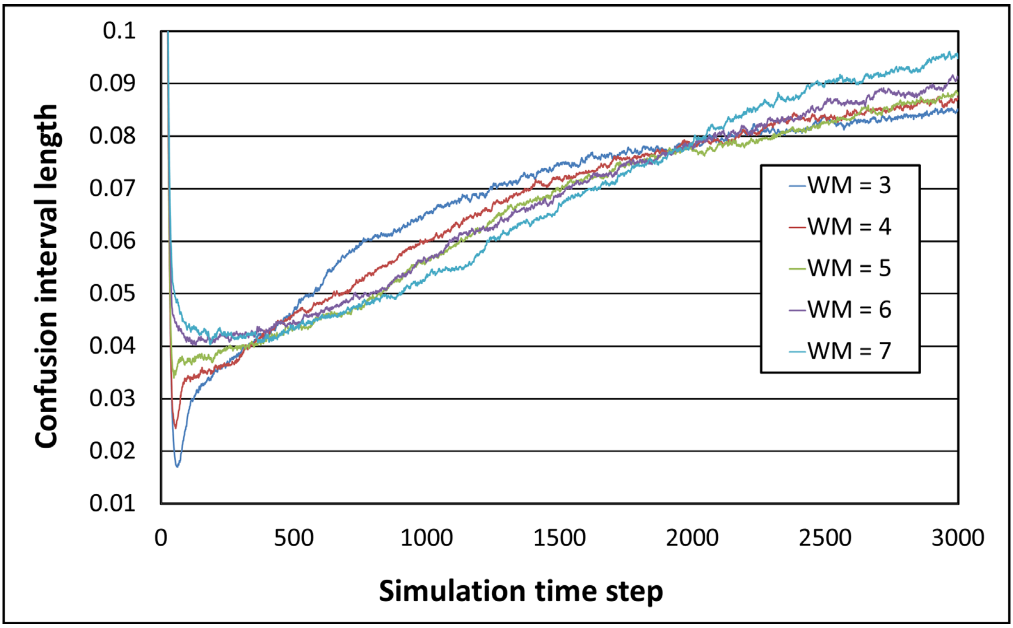
Figure 3. Number of agent connections for different working memory capacities acquired over time



C-ULM as representative of human learning is that agent learning in the C-ULM follows the asymptotic learning curve typical of learning curves observed for human declarative (Roediger III & Smith, 2012) and procedural motor learning (Wifall, McMurray, & Hazeltine, 2012). Figure 3 shows the changes in the average number of connections learned in our C-ULM simulation of human learning over the simulation time for different WM capacities (ranging from 3 to 7). The observed agent learning follows a basic learning curve corresponding to those found in human studies. Also, the shape of the learning curve is sensitive to changes in WM capacity. The slower learning associated with lower WM capacity observed is consistent with human studies (Saults, & Cowan, 2007).

Figure 4 shows the corresponding changes in the confusion interval lengths over time for different WM capacities. In particular, it shows the emergent behavior of confusion interval length dropping steeply in the beginning as agents learn when solving tasks—decreasing the uncertainty in their knowledge. However, as time progresses, their confusion starts to creep back into their knowledge base as fewer tasks are available to be solved and remaining tasks are more difficult to solve, thereby being rather unlikely to be solved. As a result, the existing knowledge decay in agents starts to factor more prominently in changing their knowledge, leading to the lengthening of confusion intervals. This pattern for the confusion interval can be viewed as indicating initial overconfidence in knowledge. This corresponds to studies showing that people exhibit overconfidence in judgments that diminishes with more experience (Hansson, Juslin, & Winman, 2008). Also, greater overconfidence has been found to be associated with shorter WM span (Hansson et al., 2008), mirroring the apparently larger initial overconfidence of agents with shorter WM spans.

Figure 4. Average confusion interval length for different working memory capacities over time



5.2. Utility of the C-ULM

The C-ULM simulation is *versatile* because of its configurability. Presently, the system can be configured along a rich set of parameters (see Table 1), including key parameters such as (1) the number of agents in the system, (2) the number of available concepts required to solve tasks, (3) the number of tasks in the environment, (4) the WM capacity of each agent, (5) the spread normalization factor *D* when knowledge update is activated from a node propagating to other connected nodes, and (6) the knowledge decay rate. Here we illustrate a small set of possible research investigations that can be conducted with C-ULM in order to better understand cognitive learning.

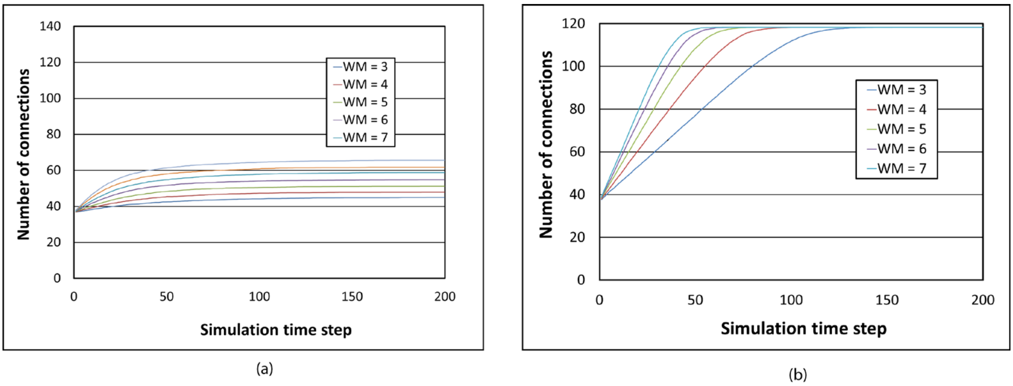
- ***What is the impact of knowledge chunking?*** Our results show that agents without the ability to chunk knowledge lead to a slower increase—as well as a reduction—in the number of agent connections (Figure 5) and also to a lower number of solved tasks (Figure 6). This reflects both the ULM principle that WM capacity is affected by prior knowledge as larger knowledge chunks lead to more knowledge being attended or retrieved through WM and corresponds to well-known findings that the greater skill and capability of experts is in large part due to knowledge chunking (Ericsson, Charness, Feltovich, & Hoffman, 2006).
- ***What is the impact of task complexity on learning?*** Our results show (Figure 7) that ULM-based agents acquire more concept connections when faced with more complex tasks. Humans also learn as they solve tasks and especially knowledge driven individuals are motivated by solving more complex tasks that can eventually lead to the acquisition of greater knowledge (Shell et al., 2010).

5.3. Implications for ULM and Cognitive Informatics

5.3.1. ULM

We believe that the findings to-date support that the C-ULM provides a working computational implementation of the core principles

Figure 5. Number of agent connections acquired over time for different working memory capacities (a) without chunking, (b) with chunking (same as Figure 3)



and mechanisms of ULM. Consistent with computational modeling as a scientific research method (McClelland, 2009), the demonstration of a viable computational model strengthens confidence in the theory of learning proposed in the ULM. The correspondence of initial results from the C-ULM with typical patterns of learning seen in human studies supports the plausibility of ULM learning mechanisms for explaining how human learning occurs. Of course no computational model can prove that a theory is correct, but as McClelland (2009) notes the purpose of a cognitive model is not to provide an exact description of the underlying cognitive or neurological processes; rather, the purpose of a model is to allow testing of the implications of theories about these processes.

A good model allows asking questions and exploring of the implications of a theory at a specific and detailed level. In the C-ULM, most agent learning parameters are adjustable. These include working memory capacity, spread of activation distance, spread of activation increment, and chunking. Also, any of the learning coefficients, decay rates, and other constants can be varied to test the implications of different values. At the global level, the number of agents, number of concepts, number of tasks, number of time steps, error margin on task solution, and task reward can be varied. The extensive variability available within the C-ULM allows for exploring a wide range of questions about human learning including the impacts of both individual differences such as working memory span and environmental influences such as task complexity and reward. Also, although we refer the nodes in a knowledge graph as concepts, they are not concepts in the everyday use of the term. The nodes can represent any level of abstraction from a neuron to an actual conceptual knowledge representation, allowing modeling at any level of the cognitive system. Similarly, while we use the language of a teacher and learner to describe the agent exchange of knowledge, the teacher need not represent another actual human teacher. The body of knowledge known to the teacher could represent the knowledge available in an environment, such as affordances.

Also, a good model of human cognition allows examination of questions that may be impractical or impossible to address in actual human studies. Because the C-ULM allows for unlimited time steps, examining the course of learning over a large number of trials is possible. This allows simulation of life-span learning and development which would be impractical to conduct with real subjects. The graph in Figure 4 suggests one possible life-span application. Although it may be true that one never forgets how to ride a bicycle, it is certainly true that one's level of proficiency decreases after a long period of disuse. One is shaky when taking up riding after a many year hiatus. The interplay of knowledge with confidence about that knowledge that can be examined with C-ULM provides an avenue for examining how proficiency is maintained over long periods, especially when use is irregular. The C-ULM also allows for examination of the learning of complex knowledge over time. It is difficult to obtain real time data, either behavioral or neurological, from people on the progress of their learning trial by trial. Most studies attempting real-time analysis examine the learning of simple knowledge, such as lists or word associates. Studying the development of meaningful expertise in a domain, which takes from 10-15 years (Ericsson et al., 2006), as a real-time phenomenon is unfeasible. The C-ULM, however, provides a means for examination of how complex knowledge is learned over a lengthy time frame, potentially shedding light on expertise development.

Although C-ULM outputs conform generally to the asymptotic learning curves associated with human learning, it is clear from C-ULM simulation runs that learning is not a smooth curve. This is somewhat apparent in Figures 3 and 4 and is more evident when individual agent curves are examined. This is particularly true for curves depicting solved tasks as in Figure 6. We often see individual agents stuck at a particular level of learned connections or solved tasks for a number of time steps followed by a jump in connections or tasks. Agents also learn connections at a much faster rate than they are able to apply them to task solution. For example, compare the time steps needed to learn the task connections in Figure 8 to the time steps needed to solve tasks in Figure 6. Many more time steps are required to implement the task connections than to acquire them. As shown in Figure 9, the learning of connection weights is not smooth as there are almost chaotic shifts within the overall trends indicated by the curves. These findings suggest nuances to learning. Knowledge connections (Figure 3) reflect what might typically be thought of as declarative or factual/conceptual knowledge (Shell et al., 2010) or concept establishment in LRMB (Wang et al., 2006). Task solution (Figure 6) reflects something more like procedural knowledge or skill (Shell et al., 2010) or higher cognitive processes like problem solving, reasoning, or decision making in LRMB (Wang et al., 2006). The C-ULM results suggest that the shift from knowing (declarative knowledge) to doing (procedural knowledge) is time consuming and may be a considerably less straight forward process than often assumed. These irregularities at the individual agent and group levels provide guidance for future investigations that can help shed light on the more micro processes of learning.

Figure 6. Number of solved tasks over time for different working memory capacities (a) without chunking (b) with chunking

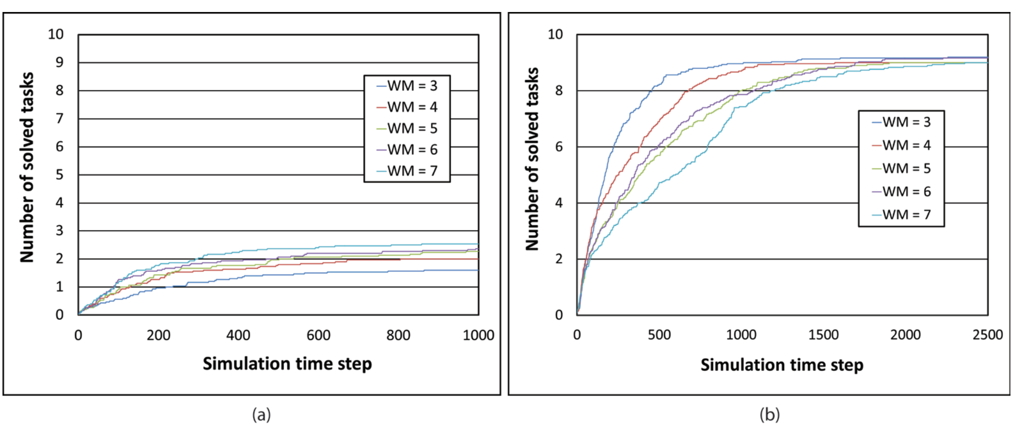
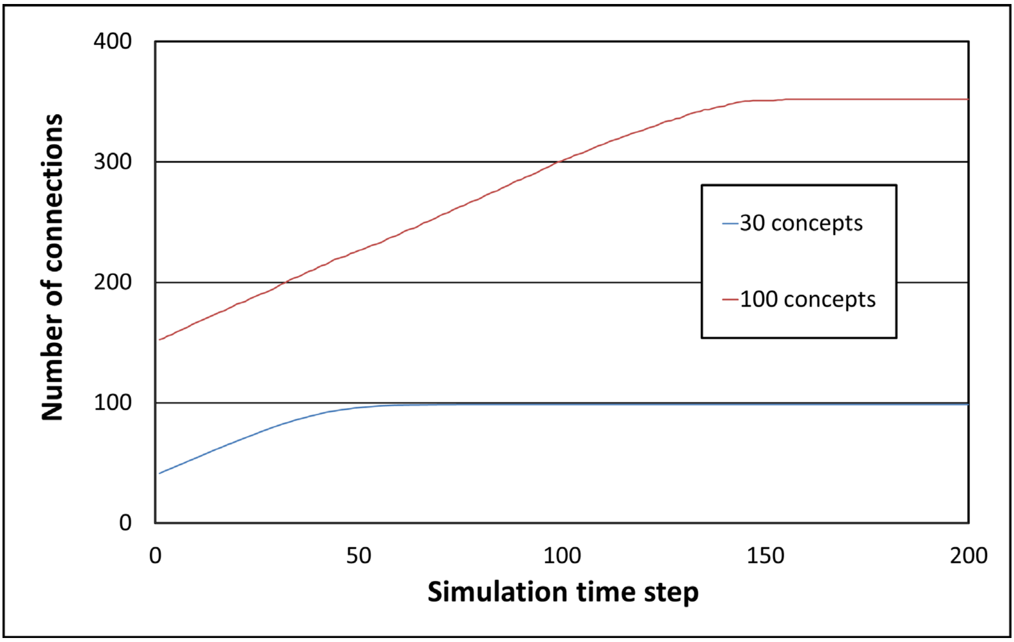


Figure 7. Number of agent connections acquired over time for different upper bounds on task complexities



5.3.2. Cognitive Informatics

Results from C-ULM suggest three potential extensions of current models in cognitive informatics (Tian et al., 2011; Wang & Chiew, 2010; Wang, Tian & Hu, 2011; Wang et al., 2006). First, C-ULM allows for a more comprehensive modeling of learning. Because C-ULM does not require specifying what the nodes in a knowledge graph represents, C-ULM can model the learning and development of any level of cognitive knowledge as represented in a model such as the layered reference model of the brain (LRMB) (Wang & Chiew, 2010; Wang et al., 2006). This potentially allows for a single unified learning mechanism to be incorporated into cognitive informatics models of brain and cognitive processes at any hierarchical level.

Second, the C-ULM learning processes do not require a-priori specification of the knowledge or problem-solutions being learned. As with neural connections, C-ULM processes expand or contract node connections and strength of connection (confusion intervals) as a function of repetition. This provides a bootstrapping capability as the C-ULM connections do not need to have any pre-programming. The ability to bootstrap has potential for developing more precise models of how humans learn from their interactions with the environment and other humans in the absence of pre-existing knowledge.

Finally, the C-ULM chunking mechanisms provide a mechanism for modeling the development of larger interconnected knowledge structures and the impacts of these larger knowledge structures on subsequent processing and storage in working memory. Although current cognitive informatics models such as OAR (Wang, 2009b; Wang et al., 2011) and projects such as Watson (Ferrucci et al., 2010) use relational and structural connections to make and expand knowledge units, they do not model chunk formation specifically. They especially don't model the prior knowledge effect that chunking has on working memory capacity and processing efficiency (Shell et

Figure 8. Average number of yet-to-be-learned connections as a performance metric, for different working memory capacities over time

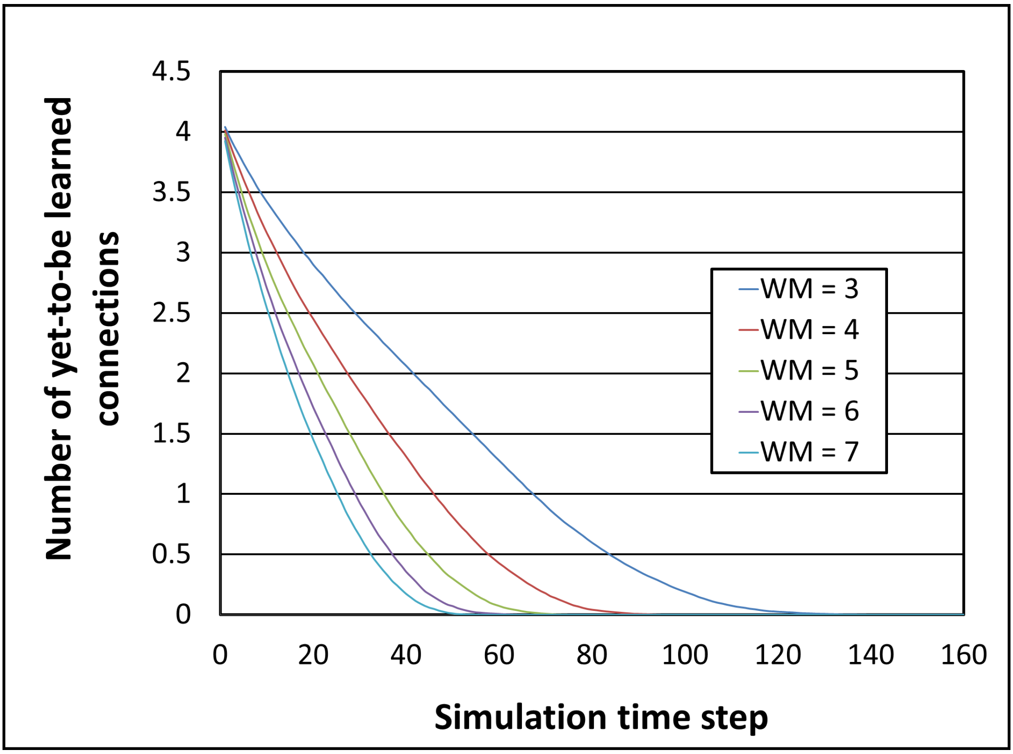
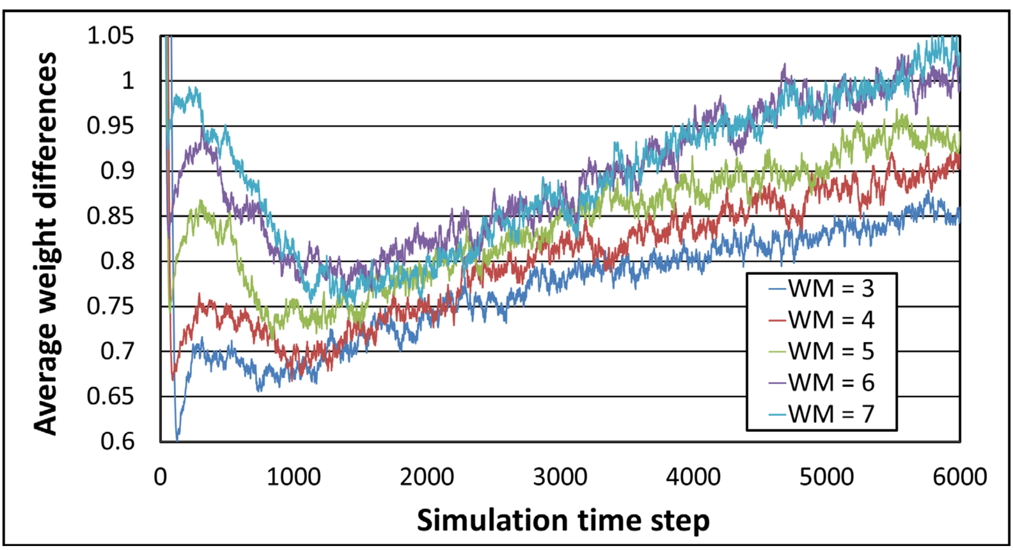


Figure 9. Average weight differences between task connections and acquired agent connections as a performance metric, for different working memory capacities over time



al., 2010). Incorporation of C-ULM chunking processes could potentially improve how larger knowledge structures are built in other cognitive informatics models.

5.4. Implications for Cognitive Computing and Agent Research

From the viewpoint of cognitive computing and computational intelligence, the contribution of the C-ULM to intelligent agent research is at two levels. One level is the modeling of individual agent reasoning inspired by the functions and relationships between the three ULM components of long-term memory knowledge, motivation and WM; and another level is the modeling of multi-agent interactions and knowledge transfer based on the principles of human teaching and learning processes. At the agent reasoning level, most multi-agent system efforts regarding modeling of human learning have been aimed at improving the performance of the agents and the multi-agent system—i.e., whether agents utilizing a particular human-based learning model improve their performance. As noted in discussions of cognitive computing (Wang, 2009a; Wang, 2011), the attractiveness of using a human-based learning model hinges upon the ability to incorporate human natural intelligence into the agent model and the intuitive abstraction of human-to-human knowledge transfer behaviors in complex situations.

From a multi-agent perspective we are more interested in the system performance at solving tasks than the similarity of the learning curves with those derived from human studies. For example, *the total number of solved tasks* of the entire system is a performance metric (Figure 6). Another example is *the average number of task connections yet to be learned* by the agents in the system (Figure 8). Since a solved task results in its concept connections being learned by the solving agents, this metric indicates the overall task solution performance. Another metric (Figure 9) is the *average weight difference* between the agent weight and the task weight corresponding to a connection between the same two concepts—that is, the difference between what the agents collectively know and what the tasks require to be solved. It measures task effectiveness but also knowledge retention and refinement. These performance metrics can be used to analyze both local, individual agent reasoning and global, emergent behaviors of the entire system. The learning and the teaching processes can be varied in order to improve both agent efficiency and effectiveness measured by these metrics. The findings from these simulation runs suggest that the C-ULM can facilitate the study of agent knowledge sharing in general and the development of utility functions involving agents that solve tasks in particular.

Incorporating the C-ULM into agent reasoning could allow researchers to investigate multiagent systems that involve human learning, either with human agents interacting with each other or artificial agents working in tandem with their human counterparts in a hybrid cognitive computing environment. Also, although we use the term ‘concept’ for nodes, the ULM learning mechanisms apply to any type of learning (Shell et al., 2010). Therefore, similar to connectionist-based models, the node can represent any level of abstraction from a neuron to a semantic concept.

Furthermore, C-ULM integrates both intrinsic and extrinsic motivation making it a more flexible solution framework for solving complex problems. C-ULM agents are extrinsically motivated by the rewards that can be obtained by solving tasks similar to common reinforcement learning methods. However, C-ULM also incorporates an intrinsic motivation component as dictated by the ULM framework, where learning is grounded a process to reduce confusion intervals of edge weights—akin to learning motivated by one’s self-efficacy, i.e., one’s confidence in one’s knowledge and expertise. Agents are motivated for acquiring knowledge, as well as for reinforcement.

For the multiagent learning and teaching field, C-ULM offers a solution on WM-level knowledge transfer between a teacher and a learner, allowing researchers to better design how agents decide on which knowledge to transfer, how to transfer, and the effectiveness of transfer. These decisions are neither arbitrary nor domain-driven; rather they are guided by the specific principles of the ULM. We believe that this has the potential of offering an alternative for modeling knowledge transfer between agents.

6. CONCLUSION

In relation to our first objective, the C-ULM provides support for the learning theory proposed in the Unified Learning Model. The C-ULM implements a viable computational simulation of the core ULM components of long-term memory, working memory, and motivation and the processes taking place among them. Our results showed that the simulation produces learning curves consistent with observed human learning and generates patterns of confusion/ confidence similar to those in human studies. As future work, we are interested in expanding and refining the C-ULM by experimenting with a larger parameter space, allowing for a variable WM and awareness threshold (Bar, 2000), experimenting with other functions such as the power law for the knowledge decay process Kahana and Adler (2002), testing against human behavioral and neurological data, and generally improving the model according to the ULM and other recent studies on human learning. We are also interested in exploring connections between C-ULM and emerging work in cognitive informatics.

From the intelligent agent perspective, the C-ULM simulation could prove useful in the research of multi-agent systems that involve human learning. Further, the C-ULM offers a general framework for knowledge transfer between agents. In the future, we are interested in exploring other types of agent interactions such as a one-to-many teaching and learning processes where a teaching agent teaches more learning agents in the same time step. These efforts will inform future developments in cognitive computers.

REFERENCES

Anderson, J. R. (1983). A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, *22*, 261–29. doi:10.1016/S0022-5371(83)90201-3

Bar, M. (2000). Conscious and nonconscious processing of visual object identity. In Y. Rossetti, & A. Revonsuo (Eds.), *Beyond dissociations: Interaction between dissociable conscious and nonconscious processing* (pp. 153–174). Amsterdam, Netherlands: John Benjamins. doi:10.1075/aicr.22.09bar

Caporale, N., & Dan, Y. (2009). Spike timing-dependent plasticity: A Hebbian learning rule. *Annual Review of Neuroscience*, *31*, 25–46. doi:10.1146/annurev.neuro.31.060407.125639 PMID:18275283

Cervantes, J. A., Rodriguez, L. F., Lopez, S., & Ramos, F. (2013). A biologically inspired computational model of moral decision making for autonomous agents. In *Proceedings of the 12th IEEE International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC'2013)* (pp. 111-117). Fordham University, NY: IEEE CS Press.

Chater, N., & Brown, C. T. (2008). From universal laws of cognition to specific cognitive models. *Cognitive Science*, *32*, 36–67. doi:10.1080/03640210701801941 PMID:21635331

Chiriacescu, V., Soh, L.-K., & Shell, D. F. (2013, July 16-18). Understanding human learning using a multi-agent simulation of the unified learning model. In *Proceedings of the 12th IEEE International Conference on Cognitive Informatics and Cognitive Computing (ICCI*CC'2013)*, Fordham University, New York City, NY (pp. 143-152).

Durning, S. J., & Artino, A. R. (2011). Situativity theory: A perspective on how participants and the environment can interact: AMEE Guide 52. *Medical Teacher*, *33*, 188–199. doi:10.3109/0142159X.2011.550965 PMID:21345059

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, C., & Rasmussen, D. (2012). A large-scale model of the functioning brain. *Science*, *338*, 1202–1205. doi:10.1126/science.1225266 PMID:23197532

Ericsson, K. A., Charness, N., Feltovich, P. J., & Hoffman, R. R. (Eds.). (2006). *The Cambridge handbook of expertise and expert performance*. Cambridge University Press. doi:10.1017/CBO9780511816796

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., & Gondek, D. et al. (2010). Building Watson: An overview of the DeepQA project. *AI Magazine*, 59–79.

Goodman, N. D., Ullman, T. D., & Tenenbaum, J. B. (2011). Learning a theory of causality. *Psychological Review*, *118*, 110–119. doi:10.1037/a0021336 PMID:21244189

Hansson, P., Juslin, P., & Winman, A. (2008). The role of short-term memory capacity and task experience for overconfidence in judgment under uncertainty. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, *34*(5), 1027–1042. doi:10.1037/a0012638 PMID:18763889

Jones, G., Gobet, F., & Pine, M. J. (2008). Computer simulations of developmental change: The contributions of working memory capacity and long-term knowledge. *Cognitive Science*, *32*, 1148–1176. doi:10.1080/03640210802073689 PMID:21585447

Kahana, M. J., & Adler, M. (2002, March). *Note on the power law of forgetting*. Waltham, MA: Brandeis University, Center for Complex Systems.

Kahneman, D. (2011). *Thinking fast and slow*. Farrar, Straus, & Giroux.

Kandel, E. R., Schwartz, J. H., & Jessell, T. M. (2000). *Principles of neural science* (Vol. 4). McGraw-Hill.

Kawato, M., & Samejima, K. (2007). Efficient reinforcement learning: Computational theories, neuroscience, and robotics. *Current Opinion in Neurobiology*, *17*, 205–212. doi:10.1016/j.conb.2007.03.004 PMID:17374483

Kiani, R., & Shadlen, M. N. (2009). Representation of confidence associated with a decision by neurons in the parietal cortex. *Science*, *324*(5928), 759–764. doi:10.1126/science.1169405 PMID:19423820

Knudsen, E. I. (2007). Fundamental components of attention. *Annual Review of Neuroscience*, *30*, 57–78. doi:10.1146/annurev.neuro.30.051606.094256 PMID:17417935

McClelland, J. L. (2009). The place of modeling in cognitive science. *Topics in Cognitive Science*, *1*, 11–38. doi:10.1111/j.1756-8765.2008.01003.x

Merrick, K. (2011). A computational model of achievement motivation for artificial agents (Extended Abstract). In *Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS'11)* (Vol. 3, pp. 1067 -1068). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Nebesniak, A. L. (2012). Effective instruction: A mathematics coach's perspective. *The Mathematics Teacher, 106*(December 2012/January 2013), 354-358.

North, M. J., Collier, N. T., & Vos, J. R. (2006). Experiences creating three implementations of the repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation*, *16*(1), 1–25. doi:10.1145/1122012.1122013

Ramscar, M., & Yarlett, D. (2007). Linguistic self-correction in the absence of feedback: A new approach to the logical problem of language acquisition. *Cognitive Science*, *31*, 927–960. doi:10.1080/03640210701703576 PMID:21635323

Roediger, H. L. III, & Smith, M. A. (2012). The "pure-study" learning curve: The learning curve without cumulative testing. *Memory & Cognition*, *40*, 989–1002. doi:10.3758/s13421-012-0213-5 PMID:22644774

Rosales, J. H., Jaime, K., & Ramos, F. (2013). An emotional regulation model with memories for virtual agents. In *Proceedings of the 12th IEEE International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC'2013)* (pp. 260-267). Fordham University, NY: IEEE CS Press.

Saults, J. S., & Cowan, N. (2007). A central capacity limit to the simultaneous storage of visual and auditory arrays in working memory. *Journal of Experimental Psychology. General*, *136*(4), 663–684. doi:10.1037/0096-3445.136.4.663 PMID:17999578

Schunk, D. H., & Zimmerman, B. J. (Eds.). (2008). Motivation and self-regulated learning: Theory, research, and applications. Erlbaum/Taylor & Francis Group.

Shell, D. F., Brooks, D. W., Trainin, G., Wilson, K., Kauffman, D. F., & Herr, L. (2010). *The Unified Learning Model: How motivational, cognitive, and neurobiological sciences inform best teaching practices*. Netherlands: Springer. doi:10.1007/978-90-481-3215-7

Tian, Y., Wang, Y., Gavrilova, M. L., & Ruhe, G. (2011). A formal knowledge representation system (fkrs) for the intelligent knowledge base of a cognitive learning engine. *International Journal of Software Science and Computational Intelligence*, *3*(4), 1–17. doi:10.4018/jssci.2011100101

Wang, Y. (2007). The theoretical framework of cognitive informatics. *International Journal of Cognitive Informatics and Natural Intelligence*, *1*(1), 1–27. doi:10.4018/jcini.2007010101

Wang, Y. (2009a). On cognitive computing. *International Journal of Software Science and Computational Intelligence*, *1*(3), 1–15. doi:10.4018/jssci.2009070101

Wang, Y. (2009b). Formal description of the cognitive process of memorization. *Transactions of Computational Science*, *5*, 81–98.

Wang, Y. (2011). Towards the synergy of cognitive informatics, neural informatics, brain informatics, and cognitive computing. *International Journal of Cognitive Informatics and Natural Intelligence*, *5*(1), 74–92. doi:10.4018/jcini.2011010105

Wang, Y., Baciu, G., Yao, T., Kinsner, W., Chan, K., & Zhang, B. et al. (2010). Perspectives on cognitive informatics and cognitive computing. *International Journal of Cognitive Informatics and Natural Intelligence*, *4*(1), 1–29. doi:10.4018/jcini.2010010101

Wang, Y., & Chiew, V. (2010). On the cognitive process of human problem solving. *Cognitive Systems Research*, *11*, 81–92. doi:10.1016/j.cogsys.2008.08.003

Wang, Y., Kinsner, W., & Zhang, D. (2009). Contemporary cybernetics and its faces of cognitive informatics and computational intelligence. [Part B]. *IEEE Transactions on Systems, Man, and Cybernetics*, *39*(4), 823–833. doi:10.1109/TSMCB.2009.2013721 PMID:19349246

Wang, Y., Tian, Y., & Hu, K. (2011). Semantic manipulations and formal ontology for machine learning based on concept algebra. *International Journal of Cognitive Informatics and Natural Intelligence*, *5*(3), 1–29. doi:10.4018/IJCINI.2011070101

Wang, Y., Wang, Y., Patel, S., & Patel, D. (2006). A layered reference model of the brain (LRMB). *IEEE Trans. on Systems, Man, and Cybernetics*. *Part C*, *36*(2), 124–133.

Wang, Y., Widrow, B., Zhang, B., Kinsner, W., Sugawara, K., & Sun, F. et al. (2011). Perspectives on the field of cognitive informatics and its future development. *International Journal of Cognitive Informatics and Natural Intelligence*, *5*(1), 1–17. doi:10.4018/jcini.2011010101

Wasserman, T. (2012). Attention, motivation, and reading coherence failure: A neuropsychological perspective. *Applied Neuropsychology: Adult*, *19*(1), 42–52. doi:10.1080/09084282.2011.643940 PMID:22385379

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*, 279–292. doi:10.1007/BF00992698

Wifall, T., McMurray, B., & Hazeltine, E. (2012). Perceptual similarity affects the learning curve (but not necessarily learning). *Journal of Experimental Psychology. General*, *2012*. doi: doi:10.1037/a0030865 PMID:23230994

Wilson-Doenges, G., & Gurung, R. A. R. (2013). Benchmarks for scholarly investigations of teaching and learning. *Australian Journal of Psychology*, *65*, 63–70. doi:10.1111/ajpy.12011

Vlad Teodor Chiriacescu is currently pursuing a MS degree in Computer Science (expected graduation date: December 2013) at the University of Nebraska. His research thesis is based on the development of the computational Unified Learning Model to investigate teaching and learning and to inform multiagent learning. Before joining the University of Nebraska, he worked as a quality assurance engineer at Axway Romania. He received his Engineer Diploma in Electrical Engineering from the Polytechnic University of Bucharest.

Leen-Kiat Soh is an Associate Professor in the Department of Computer Science and Engineering at the University of Nebraska. His current primary research interests are in multiagent systems—focusing on coalition formations and multiagent sensing and learning—and computer science education—focusing on introductory CS education, technology-enabled learning, and integration of computational and creative thinking. He has applied his research to computer-aided education systems, survey informatics, intelligent user interfaces, and multiagent modeling and simulations in cognitive learning (i.e., the computational Unified Learning Model framework) and smart grids. He has developed intelligent tutoring systems, computer-supported collaborative learning environments, and knowledge-based decision support systems. His previous research areas include image processing and decision support in the domain of remote sensing and classification of satellite sea ice images. He has published over 150 papers in peer-reviewed conferences and journals. He has been PI or Co-PI on 16 federally funded projects. He is a member of ACM, AAAI, and IEEE. Soh received his Ph.D. in Electrical Engineering with Honors from the University of Kansas.

Duane F. Shell PhD is currently Research Professor in the Department of Educational Psychology at the University of Nebraska-Lincoln. His primary research areas are learning, self-regulation and motivational influences on behavior and cognition, as these are manifest in education and public health settings. He has published over 50 refereed journal and proceedings papers and 40 additional articles and technical reports, and has presented over 80 papers and workshops at professional and scientific conferences. He has been PI or Senior Faculty on 26 funded federal, state, and foundation grants and contracts. Dr. Shell specializes in multivariate, multidimensional analyses of complex relationships among motivation, environmental factors, self-regulation, and learning. He is lead author of the Unified Learning Model: How Cognitive, Motivational, and Neurobiological Sciences inform Best Teaching Practices (ULM-Springer, 2010) and co-developer of the computational simulation (C-ULM) based on ULM principles. He has taught courses in educational psychology, cognition, learning, and history and philosophy of psychology at the University of Texas at Austin and the University of Nebraska-Lincoln and courses in computer programming at Southeast Community College Nebraska. He has worked as a computer systems analyst and programmer in business and has developed adaptive computer applications for persons with disabilities.