2013

UNDERSTANDING HUMAN LEARNING USING A MULTI-AGENT SIMULATION OF THE UNIFIED LEARNING MODEL

Vlad Chiriacescu
University of lVebraska-Lincoln, vchiriac@cse.unl.edu

Leen-Kiat Soh
University of Nebraska-Lincoln, lsoh2@unl.edu

Duane F. Shell
University of Nebraska-Lincoln, dshell2@unl.edu

Cognitive Informatics & Cognitive Computing (ICCI*CC), 2013 12th IEEE International Conference on
Year: 2013
Pages: 143 - 152, DOI: 10.1109/ICCI-CC.2013.6622237

UNDERSTANDING HUMAN LEARNING USING A MULTI-AGENT SIMULATION OF THE UNIFIED LEARNING MODEL

Vlad Chiriacescu
Department of Computer Science and Engineering University of Nebraska-Lincoln
vchiriac@cse.unl.edu

Leen-Kiat Soh
Department of Computer Science and Engineering University of Nebraska-Lincoln
lksoh@cse.unl.edu

Duane F. Shell
Department of Educational Psychology University of Nebraska-Lincoln
dshell2@unl.edu

Abstract

Within cognitive science, 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 and intelligent agents are presented.

Keywords: *Cognitive modeling; Unified Learning Model; Human Learning; Computational simulation*

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. 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 [1]. The ULM has begun to influence thinking and practice in fields such as scholarship of teaching and learning [2], situated cognition [3], pedagogy [4], and cognitive function [5].

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. Sensory/perceptual, linguistic, and number knowledge generally reside in specialized modular processing areas [6].

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 [6], 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 [7]. Elements, however, can be chunks, that increase functional working memory capacity. The second aspect is attention [8]. 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. The ULM holds that these motivators are intimately connected to working memory and

Proc. 12th IEEE Int. Conf. on Cognitive Informatics & Cognitive Computing (ICCI*CC'13)
D.F. Hsu, Y. Wang, A.R. Rao, D. Zhang, W. Kinsner, W. Pedrycz, R.C. Berwick & L.A. Zadeh (Eds.)
978-1-4799-0783-0/13/$31.00 ©2013 IEEE

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 [6, 9]. 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 [9], this repetition causes knowledge chunks in long-term memory to ultimately reflect statistical regularities present in the knowledge being learned.

Within computer science, computational modeling has a long history as a method for testing theory about human cognition [10]. Although modeling cannot prove a theory, it can provide evidence that the theory is at least plausible [10]. 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.

Our contributions can be considered from two perspectives. From the *cognitive modeling perspective*, 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 approaches to learning modeling such as reinforcement learning [11] back propagation [10], and Bayesian methods [12]. From the *multi-agent perspective*, C-ULM could benefit agent research at two levels. First, the modeling of individual agent reasoning can potentially be improved by the functions and relationships between long-term memory, motivation and working memory represented in the C-ULM. Second, C-ULM can potentially improve the modeling of agent-to-agent knowledge transfer based on the principles of human teaching and learning processes.

2. RELATED WORK

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 [13] 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 [14]. 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 confidence in knowledge accuracy by 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 [15] 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 [16] 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 [11] and multi-agent systems [17] 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 with the involved knowledge connections are updated to reflect task success (the agent learned how to solve the task); if an agent fails to solve a task, associated certainty measures are updated to reflect task failure (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 [18]. The C-ULM, however, is meant to be more faithful to the principles reflected in neural plasticity than a project such as Watson [19]. 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 can make new interconnections among the knowledge within its existing long-term memory; Watson does not learn or acquire the knowledge in its long-term memory.

3. AGENT MODEL AND MULTIAGENT FRAMEWORK

In this section we present the single agent model and the multiagent 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 rules 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. Operations within the architecture follow three ULM learning processes: (1) n***ew learning requires attention***; (2) ***learning requires repetition***; and (3) l***earning 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.

Fig. 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 Fig. 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 (Eqs. (2), (4) and (7)). Specifically, the edge weight can move in both directions, towards 0 or 1. The length of the confusion interval is shortened by the learning process (Eq. (2)) and it is increased by the process of knowledge decay (Eq. (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.

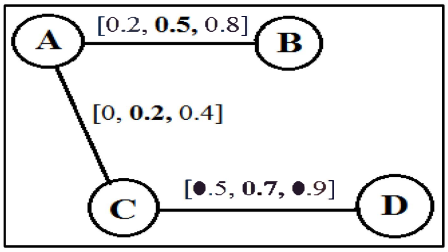

Fig. 1. LTM with concepts A, B, C, D. On each edge is outlined the associated confusion interval.

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 Eq. (1):

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

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 [1]: an intrinsic one that captures the notion of self-efficacy, i.e., length of confusion intervals, and an extrinsic one similar to reinforcement learning [17] 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 Eq. (2):

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

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 Eq. (3) below:

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

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

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

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

$$mf = f(X, WM) \cdot (m_X - AT) + f(Y, WM) \cdot (m_Y - AT) \qquad (6)$$

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 Eq. (5)); mf is the *motivation factor* (defined by Eq. (6)); cil is a learning coefficient that influences the change in the confusion interval length during a simulation time step; $d(c, x)$ 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 Eq. (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 [1], 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 [20]. 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 (Eqs. (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 Eq. (7):

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

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 Fig. 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.

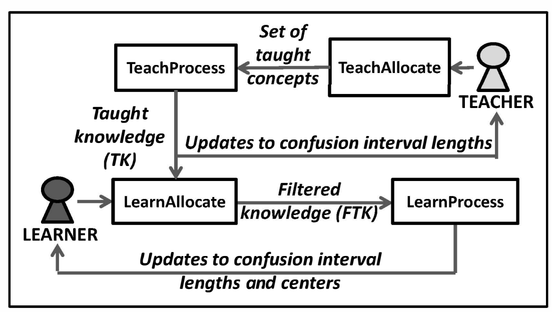

Fig. 2. Communication protocol between a teacher and a learner agent.

3.3.1. Teaching. TeachAllocate has two versions: ***TeachAllocateBasic*** 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 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: ***LearnAllocateBasic*** 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 Fig. 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 Eq. 2 and then updates the confusion interval lengths of those connections according to Eq. 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 [1] is a basic mechanism of human memory reflecting the interconnected nature of neural structure. 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 type of 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 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 [1]. 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 [1]. 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 [21]. 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 of approximately 100 nodes connected by Quad Data Rate Infiniband.

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 (figs. 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 Figs. 3 and 4. A basic threshold for acceptance of the 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 [22] and procedural motor learning [23]. Fig. 3 shows the changes in the average number of connections learned in our 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 [7].

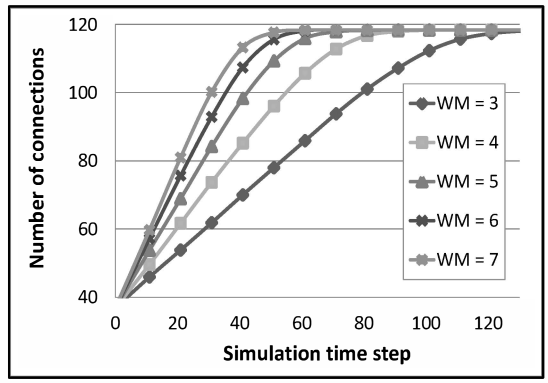

Fig. 3. Number of agent connections.

Fig. 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 and 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 [24]. Also, greater overconfidence has been found to be associated with shorter WM span [24], mirroring the apparently larger initial overconfidence of agents with shorter WM spans.

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, including key parameters such as

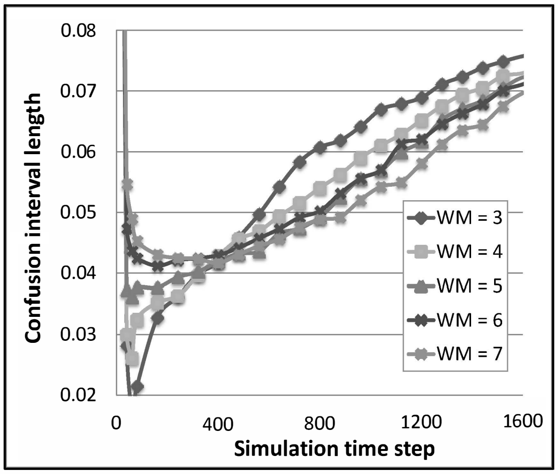

Fig. 4. Confusion interval length.

(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 in the number of agent connections and also to a lower number of solved tasks. 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 [25].

What is the impact of task complexity on learning? Our results show that ULM-based agents acquire more concept connections when faced with more complex tasks. Humans also learn as they solve tasks and individuals who are motivated by learning goals especially are motivated by solving more complex tasks that can eventually lead to the acquisition of greater knowledge [1].

5.3 Implications for ULM

We believe that the findings to-date support that the C-ULM provides a working computational implementation of the core principles and mechanisms of ULM. Consistent with computational modeling as a scientific research method [10], 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 [10] 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 Fig. 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 [25], 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.

5.4 Implications for Agent Research

From the viewpoint of 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 multi-agent systems—i.e., whether agents utilizing a particular human-based learning model improve their performance. The attractiveness of using a human-based learning model hinges upon 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 (Fig. 5). Another example is ***the average number of task connections yet to be learned*** by the agents in the system (Fig. 6). 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 (Fig. 7) 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.

6. CONCLUSIONS

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 pat-

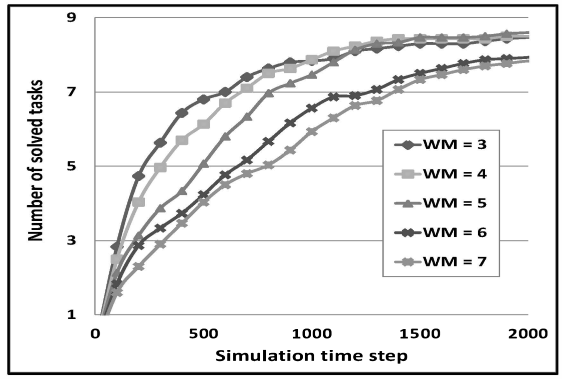

Fig. 5. Number of solved tasks.

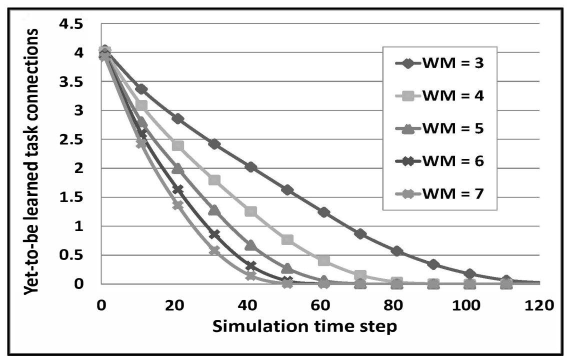

Fig. 6. Number of task connections yet-to-be learned by the agents.

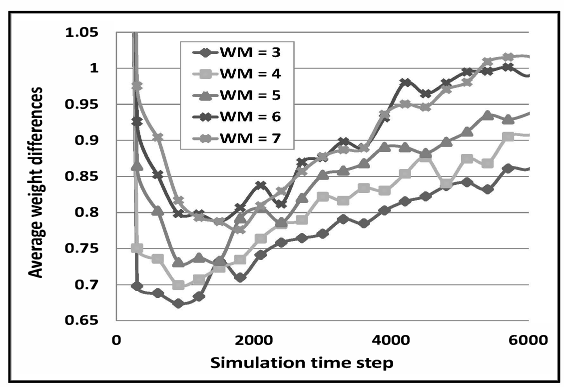

Fig. 7. Average weight differences between agent and task connections.

terns 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 [26], experimenting with other functions such as the power law for the knowledge decay process [27], testing against human behavioral and neurological data, and generally improving the model according to the ULM and other recent studies on human learning.

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.

References

[1] D. F. Shell, D. W. Brooks, G. Trainin, K. M. Wilson, D. F. Kauffman, and L. M. Herr, *The Unified Learning Model: How Motivational, Cognitive, and Neurobiological Sciences Inform Best Teaching Practices.* Netherlands, Springer, 2010.

[2] G. Wilson-Doenges and R. A. R. Gurung. "Benchmarks for scholarly investigations of teaching and learning," *Australian J. of Psych.*, vol. 65, pp. 63-70, 2013.

[3] S. J. Durning and A. R. Artino, "Situativity theory: A perspective on how participants and the environment can interact: AMEE Guide 52," *Medical Teacher*, vol. 33, pp. 188-199, 2011.

[4] A. L. Nebesniak, "Effective instruction: A mathematics coach's perspective," *The Mathematics Teacher*, vol. 106:5, pp. 354-358, December 2012/January 2013.

[5] T. Wasserman, "Attention, motivation, and reading coherence failure: A neuropsychological perspective", *Applied Neuropsych.: Adult*, vol. 19:1, pp. 42-52, 2012.

[6] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, "Principles of Neural Science", vol. 4, New York, McGraw-Hill, 2000.

[7] J. S. Saults, and N. Cowan, "A central capacity limit to the simultaneous storage of visual and auditory arrays in working memory," *J. Exper. Psych. General*, vol. 136(4): 663–684, 2007.

[8] E. I. Knudsen, "Fundamental components of attention," *Annu. Rev. Neurosci.*, vol. 30: 57–78, 2007.

[9] N. Caporale and Y. Dan, "Spike timing-dependent plasticity: A Hebbian learning rule," *Annu. Rev. Neurosci.*, vol. 31: 25-46, 2008.

[10] J. L. McClelland, "The place of modeling in cognitive science," *Topics in Cog. Sci.*, vol. 1:11-38, 2009.

[11] M. Kawato, and K. Samejima, "Efficient reinforcement learning: Computational theories, neuroscience, and robotics," *Current Opinion in Neurobiology*, vol. 17: 205-212, 2007.

[12] N. D. Goodman, T. D. Ullman, and J. B. Tenenbaum, "Learning a theory of causality," *Psych. Rev.*, vol. 118(1): 110-119, 2011.

[13] G. Jones, F. Gobet, and M. J. Pine, "Computer simulations of developmental change: The contributions of working memory capacity and long-term knowledge", *Cog. Sci.*, vol. 32(7):1148-76, 2008.

[14] K. Merrick, "A computational model of achievement motivation for artificial agents (Extended Abstract)". *Proc. AAMAS'11*, pp. 1067 -1068, 2011.

[15] M. Ramscar, and D. Yarlett, "Linguistic self-correction in the absence of feedback: A new approach to the logical problem of language acquisition." *Cog. Sci.*, vol. 31:927-960, 2007.

[16] N. Chater, and C. T. Brown, "From universal laws of cognition to specific cognitive models." *Cog. Sci.*, vol. 32: pp. 36–67, 2008.

[17] C. J. C. H. Watkins, and P. Dayan, P. "Q-learning. *Machine Learning"*, vol. 8, pp. 279-292, 1992.

[18] C. Eliasmith, T. C. Stewart, X. Choo, T. Bekolay, T. DeWolf, C. Tang, and D. Rasmussen, "A large-scale model of the functioning brain," *Science*, vol. 338: 1202-1205, 2012.

[19] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, et al., "Building Watson: An overview of the DeepQA project," *AI Magazine*, Fall 2010: 59-79, 2010.

[20] J. R. Anderson, "A spreading activation theory of memory." *J. Verb. Learning & Verb. Behavior*, vol. 22: pp. 261-295, 1983.

[21] M. J. North, N. T. Collier, and J. R. Vos, "Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit", *ACM Transactions on Modeling and Computer Simulation*, vol. 16, Issue 1, pp. 1-25, 2006.

[22] H. L. Roediger III, and M. A. Smith, "The "pure-study" learning curve: The learning curve without cumulative *testing," Mem. and Cog.*, vol. 40: 989-1002, 2012.

[23] T. Wifall, B. McMurray, and E. Hazeltine, "Perceptual similarity affects the learning curve (but not necessarily learning)," *J. Exper. Psych.: General*, Advance online publication. doi: 10.1037/a0030865, 2012.

[24] P. Hansson, P. Juslin, and A. Winman, "The role of short-term memory capacity and task experience for overconfidence in judgment under uncertainty," *J. Exper. Psych.: Learn., Mem., & Cog.*, vol. 34(5): 1027-1042, 2008.

[25] K. A. Ericsson, N. Charness, P. J. Feltovich, and R. R. Hoffman (eds.) *The Cambridge handbook of expertise and expert performance*. New York, NY: Cambridge University Press, 2006.

[26] M. Bar, "Conscious and nonconscious processing of visual object identity." In Y. Rossetti & A. Revonsuo (eds.), *Beyond dissociations: Interaction between dissociable conscious and nonconscious processing*. Amsterdam: John Benjamins, pp. 153-174, 2000.

[27] M. J. Kahana, and M. Adler, "Note on the power law of forgetting", Center for Complex Systems, Brandeis University, March 2002.