

# Extending the Influence of Contextual Information in ACT-R using Buffer Decay

Robert Thomson (thomsonr@andrew.cmu.edu)

Stefano Bennati (sbennati@andrew.cmu.edu)

Christian Lebiere (cl@cmu.edu)

Department of Psychology, Carnegie Mellon University

5000 Forbes Avenue, Pittsburgh, PA, 15213, USA

## Abstract

In this paper, we describe an extension of the theory of short-term memory decay for the ACT-R cognitive architecture. By including a short-term decay for elements recently cleared from active memory, we have extended the functionality of spreading activation as a source of implicit contextual information for the model. In ACT-R models of serial memory and decision-making, contextual information has generally been modeled using either explicit markers (e.g., positional indices) or fixed-length windows of prior elements (e.g., a lag-based representation). While markers and fixed-length windows do capture some patterns of human errors, they are inflexible, are set by the modeler and not the model, and are not psychologically-plausible representations of contextual information. In conjunction with our associative learning mechanism (Thomson & Lebiere, 2013), we show how buffer decay can provide more flexible and implicit contextual information which explains refraction, positional confusion errors, and repetition facilitation and inhibition.

**Keywords:** cognitive architectures; human memory; context

## Introduction

Over the last 50 years there has been a substantial body of literature describing how contextual information interacts with memory encoding and recall. More specifically, when the context changes between encoding and retrieval time, recall is relatively reduced compared to when retrieval occurs in the same context as encoding. Beginning with Godden and Baddeley's (1975; 1980) seminal work on context-dependent recall in natural environments (see also Smith & Vela, 2001), research has shown both internal-state (e.g., physiological) and external-cue (e.g., environmental) dependence on recall (Eich, 1980). For instance, Godden and Baddeley found that when deep sea divers learned a list underwater, they experienced reduced list recall when recalling this list on the surface as compared to recall while underwater again. Other examples of context-dependence include mood-dependence (Eich, Macaulay, & Ryan, 1994), language-dependence (Marian & Neisser, 2000), and motivation-dependence (Delgado, Stenger, & Fiez, 2004).

While context is a complex real-world phenomenon, it is much more constrained in a cognitive architecture such as ACT-R, where the information flow between a model and the environment is abstracted to a set of symbolic elements (see Anderson, Bothell, Byrne, et al., 2004). In this sense context is limited to the information available in the buffer system and the spread of activation from those items currently in working memory. Before delving more deeply into the role of contextual information in ACT-R we will present a brief overview of ACT-R and describe how context has previously been modeled in tasks involving serial memory and decision-making.

## An Overview of the ACT-R Architecture

ACT-R is a cognitive architecture defined as a set of modules which are integrated and coordinated through a centralized production system (see Figure 1). Modules access information from buffers, while the production system only responds to the contents of the buffers and not the internal processing of the modules themselves. The set of buffers therefore implicitly constitute the working memory of the architecture. The declarative memory and production system modules store and retrieve information that corresponds to *declarative* and *procedural* knowledge, respectively. Declarative knowledge is the kind of knowledge that a person can attend to, reflect upon, and usually articulate in some way, while procedural knowledge consists of the skills we display in our behavior, generally without conscious awareness.

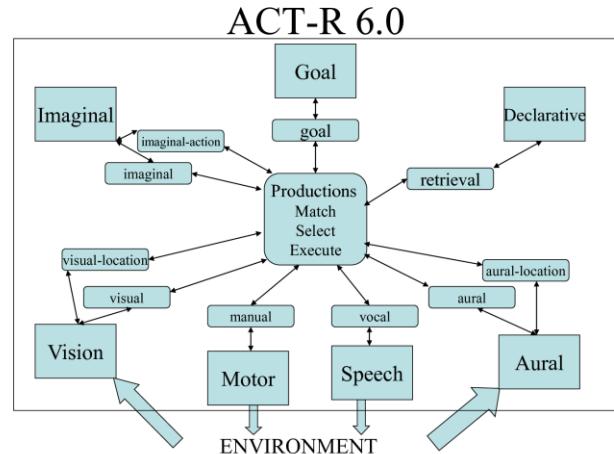


Figure 1. An overview of ACT-R's modules and their dependent buffers.

Declarative knowledge in ACT-R is represented formally in terms of chunks, which corresponds to the episodic and semantic knowledge that promotes long-term coherence in behavior. Chunks have an explicit type, and consist of a set of slot-value pairs of information. Chunks are retrieved from declarative memory (DM) by an activation process. When a retrieval request is made the most active matching chunk is returned, where activation is computed as the sum of base-level activation, spreading activation, mismatch penalty and stochastic noise. Base-level activation reflects a chunk's recency and frequency of occurrence. Activation spreads from the current focus of attention through associations among chunks in declarative memory. These associations are built up from experience, and reflect how chunks co-occur in cognitive processing. Chunks are also compared to

the desired retrieval pattern using a partial matching mechanism that subtracts from the activation of a chunk its degree of mismatch, additive for each component of the pattern and corresponding chunk value. Finally, noise is added to chunk activations to make retrieval a probabilistic process governed by a Boltzmann (softmax) distribution.

While the most active chunk is usually retrieved, a blending process (Lebiere, 1999) can also be applied that returns a derived output reflecting the similarity between the values of the contents of all chunks, weighted by their retrieval probabilities reflecting their activations. This blending process is used intensively in models of decision-making since it provides a tractable way to generalize decisions in continuous domains such as probability space.

The flow of information is controlled in ACT-R by a production system, which operates on the contents of the buffers. Each production consists of if-then condition-action pairs. Conditions are typically criteria for buffer matches, while the actions are typically changes to the contents of buffers that might trigger operations in the associated modules. The production with the highest utility is selected to fire from among the eligible productions. Please see Anderson and Lebiere (1998) and Anderson et al. (2004) for a more complete account of ACT-R mechanisms.

Before proceeding it is important to review the process of spreading activation in detail, as it is the primary mechanism for capturing implicit contextual information in ACT-R.

### Spreading Activation in ACT-R

The standard mechanism for spreading activation in ACT-R is derived from the fan effect (Anderson & Reder, 1999). The fan effect is an interference-based account of memory, where a chunk  $j$  has its spread of activation ( $S_{ji}$ ) diluted based on the number of contexts in which it has been experienced:

$$S_{ji} = s_{max} - \ln(fan_{ji})$$

$s_{max}$  is a maximum spread of association from chunk  $j$  to  $i$ , and  $fan_{ji}$  is the number of times chunk  $j$  is a slot value in all chunks in memory. The  $S_{ji}$  term is also multiplied by the weight of buffer  $k$  from where it is a source of activation, divided by the number of chunks in that buffer:

$$S_i = \sum_k \sum_j W_{kj} S_{ji}$$

The total spread of activation is summed across all the chunks in all the buffers.

There are some limitations to the current implementation of spreading activation. First, the maximum amount of spread is set by the  $s_{max}$  parameter as opposed to being learned from the environment. In Thomson and Lebiere (2013), we describe an associative learning mechanism where the strengths of association are learned from statistical regularities in the environment. Second, there is only a limited context window available when using spreading activation because activation only spreads from chunks currently in buffers. In ACT-R, once a chunk is cleared from a buffer (i.e., removed from working memory) then all the residual activation of the chunk is also removed.

As such, there is no decay of  $S_{ji}$  from the buffer, and thus context can only be spread to temporally abutting buffer contents (e.g., it only spreads proximal contextual information). These limitations restrict the kinds of contextual information which ACT-R can use.

### Prior Models Applying Context in Serial Recall

In serial memory recall, relevant contextual information includes elements recently perceived and/or recalled in the current list, and elements that have been recalled in similar positions of similar lists. The activation that spreads from contextual elements to recall candidates is described as either priming or interference based on whether it facilitates or inhibits correct recall. Patterns of errors due to contextual information – e.g., proactive and retroactive interference – include transposition errors (switching the position of two elements), omission errors (skipping an element), and intrusion errors (adding an extra element). In addition, there are context-specific effects of repetition, namely repetition facilitation when repeated items are close together, and repetition inhibition when items are farther apart.

Three related theories of memory include *chaining theory* (Ebbinghaus, 1964), *positional theory* (Conrad, 1965), and *ordinal theory* (Estes, 1972). Chaining theory assumes that order information is expressed by pairwise associations between items in memory. A limitation of chaining models is that high item-similarity and repetition cause much higher than expected confusion errors during recall (Henson, 1996). Positional theory assumes that successive items are stored in ordered slots (e.g., bins), and that these slots are implicit mechanisms which cue retrieval of item information. Positional theory, however, cannot explain context-driven human behavior such as positional confusion and repetition inhibition. Ordinal theory assumes that the position of a list item is stored as a relative value along a continuous property based on the history (c.f., prior context) of the items, or in short, that position is derived from the context of items. An advantage of ordinal theory is that no explicit positional information needs to be encoded for successful recall.

In the canonical ACT-R model of list memory (Anderson, Bothell, Lebiere, & Matessa, 1998), the model used a variation on chaining and positional theories to capture contextual information by explicitly encoding absolute serial position in the chunk representation:

```
(item-one
  ISA      item
  name    "1"
  parent   group1
  position first)
```

This model was most similar to position theory by explicitly encoding positions as a slot in the chunk. It also, however, has similarities to chaining theory as the flow of production firings attempt to recall chunks by incrementing by position, e.g., by recalling a chunk with position *first* then recalling a chunk with position *second*, and so on. The effect of context in this model was limited to positional confusion, which was accomplished by explicitly setting similarities between

position chunks. Errors of omission and intrusion are due to the base-level activation of the *item* chunks (i.e., their recency and frequency of use).

While successfully fitting overall accuracy and positional confusion (which was to be expected as it was hand-coded), explicitly representing context using positional information was not psychologically-plausible nor did it take advantage of more implicit mechanisms such as spreading activation. This explicit representation was necessary, however, because of the limitations of spreading activation not providing the necessary implicit structures to plausibly capture the effects of context on retrievals.

### Prior Models Applying Context in Decision-Making

Most models of decision-making in ACT-R use a variant of instance-based learning theory (IBL; Gonzalez, Lerch, & Lebiere, 2003). The main claim of IBL is that knowledge is generated through the creation of instances. These instances are represented in chunks with slots containing the conditions (e.g., a set of contextual cues), the decision made (e.g., an action), and the outcome of the decision (e.g., the utility of the decision). When including more than a representation of the current features available to the model, contextual cues – similar to models of serial memory – are represented explicitly as either a sequence or history of previous choices, bringing in a temporal aspect to context as well. Instances store this history as a fixed window of prior decisions – usually the two most recent decisions. By storing these prior decision instances as part of their initial context, the models can then match against them using the current context. This generates the most likely expected outcome. The model then selects the best outcome based on the dynamics of chunk retrieval and blended retrievals.

Specific examples using a lag-based context representation include a model of how batters predict baseball pitch speed (Lebiere, Gray, Salvucci & West, 2003), a model of sequence learning (Lebiere & Wallach, 2001), and a model of playing Paper Rock Scissors (West & Lebiere, 2001). While the previous models do not all use spreading activation to further support contextual effects in decision-making, it has been applied in an instance-based model of sequential diagnostic reasoning (Mehlhorn, Taatgen, Lebiere, & Krems, 2011). In this model, activation spreads from the set of symptoms to possible diagnoses, with context-based interference due to fan effects (i.e., the set of diagnoses to which a given symptom is associated).

There are some issues with using explicit fixed-length context windows and the current ACT-R implementation of spreading activation as the basis for generating contextual information. Fixed-length windows are representationally rigid and cannot account for the dynamic chunking that occurs during sequence learning, such as the kind of hierarchically-organized pattern-matching that occurs in chess mastery (Chase & Simon, 1973). They have proven effective in the previously-mentioned examples simply because they are the right level of abstraction to capture the limited range of human behaviors on abstract decision-making tasks such as repeated binary choice.

On the other hand, spreading activation is a more implicit mechanism that should capture many of context-driven effects (e.g., priming and interference). The current implementation of spreading activation, however, has some limitations that we have previously discussed. In particular, activations are not learned but are a fixed about set by a parameter and reduced by the log of the fan, that is, by the number of elements that contain the source chunk as a slot value. In addition, spreading activation becomes unstable with a fairly small amount of symbolic overlap, resulting in models which have many instances having the spread of activation become inhibitory. With a default spread (i.e., the *smax* parameter) of 2 (Lebiere, 1999), a chunk can appear as a value in 7 chunks before becoming inhibitory. This may be reasonable for modeling a single psychology experiment, but in a complex decision-making task a model using IBL will have its  $S_{ji}$  become inhibitory for most instance chunks.

### Prior Attempts to Capture Context Effects

In ACT-R 4, strengths of association ( $S_{ji}$ ) were learned by the model by experiencing environmental context using a form of associative learning (Schooler & Anderson, 1993). Spreading activation denoted the log-likelihood that chunk  $N_i$  was relevant given context  $C_j$ :

$$S_{ji} = \ln \left( \frac{P(N_i|C_j)}{P(\bar{N}_i|C_j)} \right) = \frac{P(N_i)}{P(\bar{N}_i)} \prod_j \frac{P(C_j|N_i)}{P(C_j|\bar{N}_i)}$$

Spread was computed as the log-likelihood of the number of times a chunk was in context (i.e., in the goal buffer) over times the chunk was retrieved in total. This function was deprecated due to “catastrophic” instabilities in the mechanism. When  $C_j$  is usually not in the context when  $N_i$  is needed  $P(N_i|C_j)$  is much smaller than  $P(\bar{N}_i|C_j)$ , thus the  $S_{ji}$  become negative as the log-likelihood approaches 0. This issue is due to the global context term  $C_j$  which alters the  $S_{ji}$  whenever a chunk is added and/or production fires, and is magnified by the asymmetrical log-likelihood calculation which penalizes the context ratio in long-running models.

To reconcile the difficulties in previous implementations of associative learning, Thomson and Lebiere (2013) derived a Hebbian-inspired associative learning rule influenced by spike-timing dependent plasticity (STDP; Caporale & Dan, 2008) and Rescorla-Wagner (1972) models of classical conditioning. Unlike traditional Hebbian implementations which simply increment associations so long as both pre-synaptic and post-synaptic neurons fire within a temporal window, in STDP if the pre-synaptic neuron fires before the post-synaptic then the association is strengthened, while if the post-synaptic neuron fires before the pre-synaptic then the association is inhibited. We assume that the set of chunks in all buffers when a retrieval request is initiated is loosely analogous to pre-synaptic neuronal firings, and the set of chunks in the buffers when the retrieval request is completed is loosely analogous to post-synaptic firings.

Associations are created and updated when the set of context chunks  $C$  (and its slots) are associated to a successfully requested chunk  $N$  (and its slots) according to a variant of the Rescorla-Wagner learning rule:

$$S_{\{j\} \cup \{j \in C\} \setminus \{i \mid i \in N\}} = \left( (1 - \frac{\alpha}{\omega_i}) S_{ji}^{\text{prior}} \right) + \frac{\sigma \beta_j}{\omega_i}$$

where  $\alpha$  is an interference rate that determines how much the prior  $S_{ji}$  is squashed,  $\sigma$  is a learning rate equivalent to the  $smax$  parameter,  $\beta_j$  is the weight of the buffer containing  $j$  and is equivalent to  $W_{kj}$ , and  $\omega_i$  is the number of valid slots in the requested chunk. The  $\sigma$  value is positive for chunks that were in buffers at the time the request was initiated (hebbian learning) and negative for chunks that were in buffers at the time the request was completed (anti-hebbian learning). Also, associations do not decay over time.

This equation summarizes the set of  $S_{ji}$  as the sum of positively associating the chunks in buffers when a request is initiated with the successfully requested chunk, and negatively associating chunks in buffers when the request is completed with the successfully requested chunk. The successfully requested chunk is included in the request-completed context, causing the chunk to become negatively associated with itself, leading to a natural refractory firing period analogous to base-level inhibition (Lebiere & Best, 2009). This avoids the kind of self-activation feedback loops that lead to model instability.

To provide an example, assume source  $j$  has just made a positive association with target  $I$  and has a prior  $S_{ji}$  value of 3, the default  $\alpha$  squashing value of .2, learning rate  $\sigma$  value of 1, and buffer weight  $\beta_j$  of 1, then the new  $S_{ji}$  would be:

$$S_{ji} = ((1 - .2) * 3) + 1 = 3.4$$

Due to the influence of the interference rate on the prior  $S_{ji}$ , there was a modest .4 increase in the association strength.

This mechanism is bounded, stable, and symmetrical. Compared to the Anderson et al., (1998) model of serial memory previously discussed, it was able to predict human accuracy without any parameter adjustment,  $r^2 = .977$ , and also without the need for any explicit contextual information to be encoded in the chunk structure:

```
(item-one
  ISA      item
  name     "1"
```

While this mechanism captured human accuracy, the associations that were formed were essentially still a fixed lag-1 window (albeit being learned automatically and implicit) since there was no residual spreading activation of a chunk once it was cleared from a buffer. Thus, the model was only able to capture very proximal associations ( $\pm 1$  position). As such, it did not capture the more distal positional confusion or repetition facilitation and inhibition. What was needed was a way to flexibly (and implicitly) expand the context window, or in other words, what was needed was a mechanism to handle buffer decay.

## A Buffer Decay Mechanism for ACT-R

The buffer decay module adds a short-term decay to the activation of chunks recently cleared from buffers. At the present, chunks still in buffers do not decay, as they are still in *active* memory. This short-term decay is used by the associative learning module and spreading activation to learn and spread associations whose strength is mediated

according to the remaining short-term activation. When a chunk's short-term activation falls below a given threshold then the residual activation of the chunk is considered to have fully decayed from memory. A final point is that each time a chunk is cleared from memory it is given a separate decaying short-term activation. Thus, repetition is treated as two separate traces, each with an individual decay activation.

In this paper we present two possibilities for decay functions, both of which are based on the existing base-level decay equations standard in ACT-R. This is only a first-pass using existing (and well-justified) activation equations, and may not reflect the best decay profile after further justification against human performance. The first function we present is a fixed decay rate, and works similar to the optimized base-level learning equation. The second is a dynamically-generated decay rate, and sets decay based on the length of time that the chunk remained in the buffer before being cleared. This dynamically-generated decay rate simulates the effect of encoding and elaboration theorized to take place while the chunk remains in the buffer over time.

### Fixed-Strength Decay

As previously mentioned, short-term decay is derived from the optimized base-level learning equation  $B_i$ :

$$B_i = \ln(n/(1 - d)) - d * \ln(1 + L)$$

where  $n$  is the number of presentations of chunk  $i$ ,  $L$  is the time since the creation of chunk  $i$ , and  $d$  is the decay-rate obtained from the `:bll` parameter. Since the number of presentations  $n$  of chunk  $i$  will always be 1, we can simplify the short-term decay equation  $D_i$  to:

$$D_i = 1 - d(\ln(1 + L))$$

where  $d$  is the decay rate, and  $L$  is the time since the chunk has been cleared from the buffer.

Classic memory literature has argued that the effective duration of short-term memory without the ability to rehearse is between 8-18 seconds (Peterson & Peterson, 1959; Waugh & Norman, 1965). As seen in Figure 2, this corresponds to a  $d$  between .3 and .5 (which is the default range of acceptable values in ACT-R). With a  $d$  of .3 it takes 27 seconds for the chunk to fully decay (however its influence is negligible after 18-20 seconds), with a  $d$  of .4 it takes 11 seconds for the chunk to decay, and with a  $d$  of .5 it takes 6 seconds for the chunk to decay.

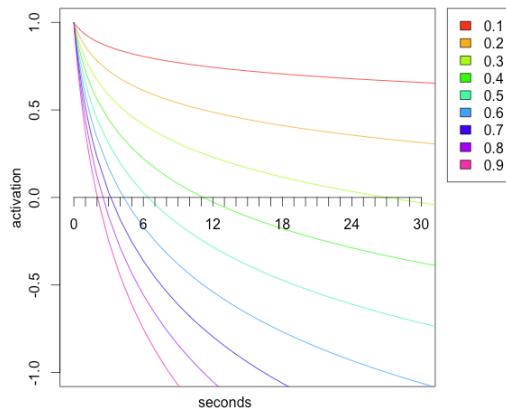


Figure 2. Activations based on different base-level decay rates.

## Dynamically-Generated Decay Strength

The process of encoding, consolidation, and elaboration of information is not an all-or-nothing process, yet it is abstracted to such in ACT-R where a chunk and its contents are either available in a buffer or they are not. Some researchers (e.g., Nyamsuren, 2012) have attempted to model perceptual encoding, where slots from chunks in the visual buffer become available probabilistically based on attentional constraints. We instead approach the processes of consolidation and elaboration by basing the short-term decay rate based on the length of time a chunk was present in a buffer. While the core of the decay function remains the same as the fixed-rate decay, dynamically-generated decay strength uses an exponential function to replace the  $d$  short-term decay parameter in the decay function:

$$d \mapsto \left( \frac{e^{-t} + \alpha}{1 + \alpha} \right)$$

which is substituted into the short-term decay equation:

$$D_i = 1 - \left( \frac{e^{-t} + \alpha}{1 + \alpha} \right) \ln(1 + L)$$

where  $t$  is the time the chunk was in the buffer,  $L$  is the time since the chunk has been cleared, and  $\alpha$  is a parameter to control the asymptotic decay rate (i.e., the slowest possible decay rate). Table 1 shows the decay rate based on the time a chunk remains in a buffer for three values of  $\alpha$ .

**Table 1.**  $d$  values based on time in buffer.

time (s)	$d, \alpha = .3$	$d, \alpha = .4$	$d, \alpha = .5$
<b>0.05</b>	0.962	0.965	0.967
<b>0.10</b>	0.927	0.932	0.937
<b>0.20</b>	0.861	0.871	0.879
<b>0.50</b>	0.697	0.719	0.738
<b>1.0</b>	0.514	0.548	0.579
<b>1.5</b>	0.402	0.445	0.482
<b>2.0</b>	0.335	0.382	0.424
<b>2.5</b>	0.294	0.344	0.388
<b>3.0</b>	0.269	0.321	0.367
<b>3.5</b>	0.254	0.307	0.353
<b>4.0</b>	0.245	0.299	0.346
<b>5.0</b>	0.236	0.291	0.338

When a chunk remains in a buffer for only 50 ms (i.e., the length of a single production) then it has a base-level decay rate greater than .9, which approximates the complete decay of its residual spreading activation in 2 seconds. This is coincidentally the time-course of visuospatial and auditory short-term memory. However, once a chunk has been in a buffer for longer than 2 to 3 seconds then it is near its asymptotic decay rate, which reflects the chunk being fully-consolidated into memory. Thus, the residual spreading activation from the chunk's strength of association will remain maximally influential on later requests.

A consequence of adopting this dynamically-generated decay rate is that rehearsal strategy and the time-course of production firings become much more important. For instance, it is possible to quickly retrieve chunks and clear them from a buffer, which results in chunks with potentially higher base-level activations, but with more rapid short-term decay, resulting in a sparser context from which to

learn (i.e., having a smaller context window). This strategy is analogous to the effect of *cramming* while studying as it reflects rapidly encoding data without making elaborating links to the surrounding context. Crammers will also have high proximal proactive interference from items recently cleared. Another phenotype are *elaborators* who let chunks remain in buffers for a longer time (up to several seconds), which would result in potentially lower base-level activation but a richer context (i.e., a broader context window) from which to learn associations and spread activation from.

## Associative Learning from Buffer Decay

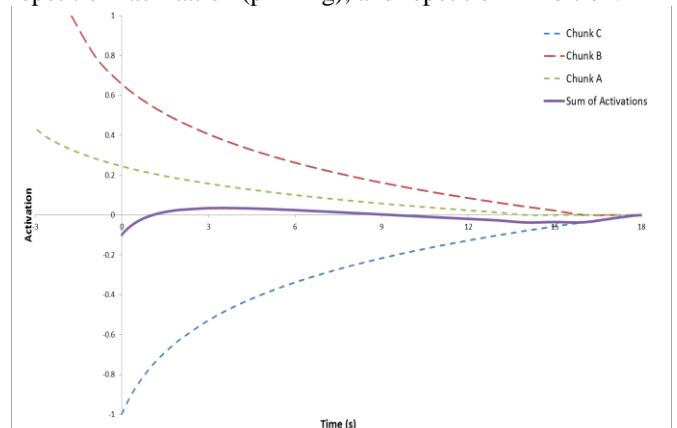
Learning associations from decaying chunks uses the same formula described previously with an extra decay term:

$$S_{\{j|v_j \in C\} \{i|v_i \in N\}} = \left( (1 - \frac{\alpha D_j}{\omega_i}) S_{ji}^{\text{prior}} \right) + \frac{\sigma \beta_j D_j}{\omega_i}$$

where  $D_j$  is the remaining short-term activation from source  $j$  when the chunk is currently decaying. If the chunk is still in a buffer then  $D_j$  is automatically set to a value of 1.

## Applications of Buffer Decay

When buffer decay is used in conjunction with associative learning, several emergent contextual effects become apparent. Imagine beginning to recall the sequence A-B-C one letter at a time. Over time, what influence would the prior context (chunks A, B, and C) exhibit on subsequent attempts to recall chunk C again? Figure 3 shows the effect of having multiple decaying chunks (A, B, and C) in the context (the dotted lines) and their summed activation (the solid line) over time. Based on the fixed-decay function, the solid line shows a net inhibitory effect for several hundred milliseconds followed by a period of priming, and finally a period of inhibition decaying to a neutral state of activation once all the contextual elements have decayed. These states, respectively, correspond to the time-course of refractory, repetition facilitation (priming), and repetition inhibition.



**Figure 3.** Activations based on recent context using associative learning and buffer decay. The interval between presented each chunk was 1.75 seconds, and the  $d$  was set to .35. The association strength from B to C was set to 1, the association strength from A to C was set to .5, and C auto-associated -1 to itself.

While the net effect of this contextual spread is only marginal, it represents a change in accuracy of several

percent and is in the rough order of magnitude found in the literature (Henson, 1996). This effect also represents only an initial implementation of the buffer decay equation, and represents an emergent behavior based on the implicit strengths of association and the temporal dynamics (e.g., presentations rate) of the task environment.

Of greater interest, this profile of inhibition followed by priming followed by inhibition is not set in stone. Changes in either the strengths of association (from prior experience) or the task environment (e.g., faster presentation rates) could result in a longer net priming period or a long net inhibitory period. The story becomes even more interesting when considering the dynamically-generated decay rate. Dynamically-generated decay provides a mechanism for additional individual differences by determining short-term decay rates based on the how long the model consolidates each chunk (i.e., how long it remained in a buffer).

## Conclusion

In this paper we have presented an initial mechanism for buffer decay in ACT-R which, in conjunction with our associative learning mechanism, provides a unified mechanism for flexible, implicit, and behaviorally-plausible context-based learning. This mechanism explains such context-based memory effects such as refraction, positional confusion errors, and repetition facilitation and inhibition.

## Acknowledgments

This work was conducted through collaboration in the Robotics Consortium sponsored by the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016; and by Intelligence Advanced Research Projects Activity via DOI contract number D10PC20021. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI, or the U.S. Government.

## References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., Qin, Y. (2004). An integrated theory of Mind. *Psychological Review*, 111, 1036-1060.

Anderson, J. R., & Betz, J. (2001). A hybrid model of categorization. *Psychonomic Bulletin & Review*, 8, 629-647.

Anderson, J. R. & Reder, L. M. (1999). The fan effect: New results and new theories. *Journal of Experimental Psychology: General*, 128, 186-197.

Anderson, J. R., Bothell, D., Lebiere, C. & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, 38, 341-380.

Anderson, J. R., and Lebiere, C. (1998). *The atomic components of thought*, Erlbaum, Mahwah, NJ.

Caporale, N., & Dan, Y. (2008). Spike Timing-Dependent Plasticity: A Hebbian Learning Rule. *Annual Review of Neuroscience*, 31, 25-46.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive psychology*, 4 (1), 55-81.

Conrad, R. (1965). Order error in immediate recall of sequences. *Journal of Verbal Learning and Verbal Behavior*, 4, 161-169.

Delgado, M. R., Stenger, V. A., & Fiez, J. A. (2004). Motivation-dependent responses in the human caudate nucleus. *Cerebral Cortex*, 14 (9), 1022-1030.

Ebbinghaus, H. (1964). *Memory: a contribution to experimental psychology*. New York: Dover.

Eich, J. M. (1980). The cue-dependent nature of state-dependent retrieval. *Memory & Cognition*, 8 (2), 157-173.

Eich, E., Macaulay, D., & Ryan, L. (1994). Mood dependent memory for events of the personal past. *Journal of Experimental Psychology: General*, 123 (2), 201-215.

Estes, W. K. (1972). An associative basis for coding and organisation in memory. In A. W. Melton & E. Martin (Eds.), *Coding Processes in Human Memory* (pp. 161-190).

Washington DC: V. H. Winston & Sons.

Godden, D., & Baddeley, A. (1980). When does context influence recognition memory?. *British Journal of Psychology*, 71, 99-104.

Godden, D., & Baddeley, A. (1975). Context dependent memory in two natural environments. *British Journal of Psychology*, 66 (3), 325-331.

Henson, R. N. (1996). Unchained memory: Error patterns rule out chaining models of immediate serial recall. *The Quarterly Journal of Experimental Psychology: Section A*, 49(1), 80-115.

Lebiere, C. & Best, B. J. (2009). Balancing Long-Term Reinforcement and Short-Term Inhibition. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.

Lebiere, C., Gray, R., Salvucci, D., & West, R. (2003). Choice and learning under uncertainty: A case study in baseball batting. In *Proceedings of the 25th Annual Meeting of the Cognitive Science Society* (pp. 704-709).

Lebiere, C., & Wallach, D. (2001). Sequence learning in the ACT-R cognitive architecture: Empirical analysis of a hybrid model. In *Sequence learning* (pp. 188-212). Springer Berlin Heidelberg.

Marian, V., & Neisser, U. (2000). Language-dependent recall of autobiographical memories. *Journal of Experimental Psychology: General*, 129 (3), 361-368.

Mehlhorn, K., Taatgen, N.A., Lebiere, C., Krems, J.F. (2011). Memory Activation and the Availability of Explanations in Sequential Diagnostic Reasoning. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 37, 1391-1411.

Nyamsuren, E., & Taatgen, N. A. (2013). Pre-attentive and attentive vision module. *Cognitive Systems Research*, 24, 62-71.

Peterson, L., & Peterson, M. J. (1959). Short-term retention of individual verbal items. *Journal of experimental psychology*, 58(3), 193-198.

Rescorla, R. A. & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A.H. Black & W.F. Prokasy (Eds.), *Classical Conditioning II*, Appleton-Century-Crofts.

Schooler, L. J. & Anderson, J. R. (1993). Recency and Context: An Environmental Analysis of Memory. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, pp. 889-894.

Smith, S., & Vela, E. (2001). Environmental context-dependent memory: A review and meta-analysis. *Psychonomics Bulletin Review*, 8 (2), 203-220.

Steinberg, S. (1966). High-speed scanning in human memory. *Science*, 153, 652-654.

Stewart, T. C., West, R., & Lebiere, C. (2009). Applying Cognitive Architectures to Decision-Making: How Cognitive Theory and the Equivalence Measure Triumphed in the Technion Prediction Tournament. In N. A. Taatgen & H. van Rijn (Eds.), *Proceedings of the 31st Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.

Waugh, N. C., & Norman, D. A. (1965). Primary memory. *Psychological Review*, 72 (2), 89-104.