An ACT-R/PM Model of the Articulatory Loop

David G. Huss (dhuss@rice.edu)
Rice University, 23 Sunset Blvd.
Houston, TX 77005 USA

Michael D. Byrne (byrne@rice.edu)
Department of Psychology, 6100 Main Street
Houston, TX 77005 USA

Abstract

This paper describes an initial attempt to model serial memory span tasks using articulatory rather than retrieval-based rehearsal. This was accomplished through utilizing ACT-R's audition and vocal modules to implement an articulatory loop. Other production systems, notably EPIC, have models for verbal working memory and the articulatory loop (Kieras et al. 1999). Until now, this was lacking in ACT-R. It is also believed that the implementation of an articulatory loop can be useful in models of more complex tasks outside of serial memory. The model owes its inspiration to the existent EPIC model, but because there are major architectural differences between the two systems, there are marked differences in how the two models approach the task.

Introduction

The articulatory loop is a memory mechanism proposed by Baddeley to explain a wide range of verbal serial memory effects. This paper intends to demonstrate how ACT-R can incorporate the concepts of an articulatory loop into a model of a serial memory task. Such a demonstration is worthwhile for its own sake, but also because subvocal articulation is believed to be a commonly used memory strategy. Thus it may be important in numerous tasks outside the domain of serial memory and should be considered in their respective models.

The articulatory loop functions through subvocal rehearsal. Verbal items are subvocalized and subsequently processed by the auditory system only to be subvocalized again. Thus, a loop is created between the vocal and auditory systems. However, such a loop is thought to have its limits, as items decay and are forgotten if rehearsal does not occur frequently enough. This results in a loop that appears able to store approximately 2 seconds of verbal information (Baddeley, Thomson, & Buchanan, 1975).

There are a variety of serial memory effects that all can be explained by the articulatory loop and thus provide support for it.

Time of Articulation

Items that take more time to articulate are less likely to be recalled correctly (Baddeley et al, 1975). For example a list of the words "hate", "sum", and "harm" would be more likely to be remembered than a list of the words "university" "association" and "representative". This is explained by the articulatory loop as longer words are articulated less often, raising the chance of their decay.

List Length Effect

Serial recall diminishes as list length increases (Baddeley et al., 1975). Like the time of articulation effect, if one thinks in the context of the articulatory loop, longer lists mean that each item is subvocalized less often.

Unattended Speech

Unattended speech has been shown to disrupt verbal working memory (Colle and Welsh, 1976). It is assumed to do this through interfering with the functioning of the phonological loop.

Phonological Similarity

Serial recall is impaired for phonologically similar words (Conrad, 1964). This points to a phonological system in working memory.

Articulatory Suppression

Serial recall is impaired when subvocal rehearsal is suppressed (Baddeley, Lewis, & Vallar, 1984). This suppression is accomplished through another verbal task (vocalizing repeated stimuli) and is strong evidence for the articulatory loop.

Memory Mechanisms in ACT-R

Currently, no complex ACT-R models take considerations of the articulatory loop into account. This is troubling due to the fact that the articulatory loop has many unique features, which lead to the previously mentioned properties of verbal working memory. These effects are currently not considered. As was previously mentioned, one of the purposes of this endeavor is to investigate the articulatory loop in ACT-R with the idea of possibly including it in more complex models.

Models of Serial Memory

There are numerous computational models of serial memory and the articulatory loop. Of these, two are particularly relevant to our ACT-R model. These are an

EPIC model of the articulatory loop (Kieras, Meyer, Mueller, & Seymour, 1999) and a previous ACT-R model of serial memory (Anderson, Bothell, Lebiere, & Matessa, 1998). Both of these models differ from ours, yet model many of the same features.

EPIC

EPIC is a cognitive architecture that shares many features with ACT-R/PM. They both use production systems to model "central" cognition and have separate, asynchronous processors or modules to handle peripheral systems such as vision. In this sense, they both resemble more fleshed-out versions of the Model Human Processor of Card, Moran, & Newell (1983). Yet, despite these similarities, there are some key differences.

EPIC has no "cognitive bottleneck." It is a production system in which all matching productions fire (Meyers & Kieras, 1997). This contrasts with ACT-R's conflict resolution mechanism, which results in only one production firing per cycle. EPIC's memory structure is also much less defined than ACT-R's. There are other differences, but these are two major ones that lead to two very different architectures.

The EPIC Model

Like our proposed ACT-R/PM model, the EPIC model is articulatory in its utilization of its architecture's vocal and auditory systems. Our modeling approach differs in its implementation of the articulatory loop. Thus, a comparison between the results of the models should prove interesting.

ACT-R Model of Serial Memory

The Anderson et al. (1998) model of serial memory exists but does not utilize the auditory and vocal modules to implement an articulatory loop. Instead, subvocal articulation was simulated by regulating retrieval and production firing times. Thus the model was not truly articulatory, differentiating it from our ACT-R/PM model of the articulatory loop.

In addition, the ACT-R model of serial memory was concerned with higher order effects. It is hard to compare it with our model as they are aimed at understanding serial recall at two different levels. However there are similarities between the structures and approaches of these ACT-R/PM models.

The Data Set

In order to validate the model, a data set had to be chosen. After taking numerous factors into account, we decided on the first experiment from Baddeley et. al's classic 1975 study. Aside from the seminal nature of this experiment, such a choice allows for comparison to the EPIC model.

In this experiment, two variables were manipulated: word length (in syllables) and list length. Pools of short words were used to create random lists of 4-8 items in length (figure 1).

Word List	
Short Words	**Long Words**
sum	association
harm	representative
Wit	opportunity
bond	organization
yield	considerable
worst	immediately
twice	university
hate	individual

Figure 1: Word Pool from experiment 1 (Baddeley et al, 1975)

Subjects were presented eight sequences of each list length and word duration. The word lists were read to the subject at a rate of 1.5 seconds a word and the subject was given 15 seconds to respond.

Performance was scored based on the correct recall (all the items being correct and in the right order) of the entire list.

Results

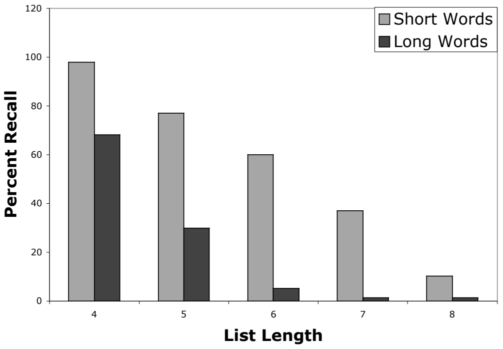

Figure 2: Data from Experiment 1 (Baddeley et. al, 1975)

Both word length effects and list length effects are present. Subjects showed decreased performance for longer lists as well as for longer word durations. The interaction of the two was shown to further decrease performance.

Modeling the Data

Based on the data from Baddeley's et al. experiment, a model was constructed with the goal of demonstrating effects of list length and word duration. The nature of the task places a large emphasis on ACT-R's activation based structure of memory and on the perceptual motor elements necessary to implement the loop.

Activation Based Memory

One of ACT-R's distinguishing features is its activation-based system of memory. Since the task being modeled is a serial memory task, this system has a great influence on the creation of the model and how it works.

Basic Activation

Each chunk (element) in ACT-R's memory has an associated activation level. This activation level is a dynamic, changing entity that determines the likelihood and the speed at which a chunk is retrieved. The relevant equation for determining a chunk's activation for this model is given in 1.

$$A_i = B_i + \sum_j W_j S_{ji} + \epsilon \quad (1)$$

In this equation B_i represents a chunks baselevel (this will be discussed later), $W_j S_{ji}$ represents a spreading of activation factor and ϵ is a random noise factor.

Base Level Activation and Learning

The equation for a chunk's baselevel is defined in 2. This shows that a chunk's baselevel is a function of the summation of several logarithmically decaying entities. These are the times since the chunk was last retrieved or encoded.

$$B = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) (2)$$

Thus, the more often an item is accessed, the more active it will be, and the more recently it has been accessed, the more active it will be.

Retrieval Threshold

The retrieval threshold is simply the value that A_i must equal in order for its respective chunk to be retrieved from memory. In order for successful recall to occur the articulatory loop must rehearse an item to maintain its base level, and hence its activation, above this threshold.

Perceptual Motor Modules in ACT-R

A key piece of the model is its use of ACT-R/PM's audition and vocal modules (Byrne & Anderson, 1998). In ACT-R/PM the various perceptual and motor modules operate in parallel to the core of the production system and each other, as in figure 3. These include the vision, motor, audition, and vocal modules. Thus while the vocal module is busy articulating an item, productions can continue to fire and the audition module can be attending to stimuli. It is clear that although only a single production fires at once, the various perceptual/motor modules allow for a great deal of parallelism within ACT-R/PM. Utilizing such a parallelism allows for the implementation of the articulatory loop.

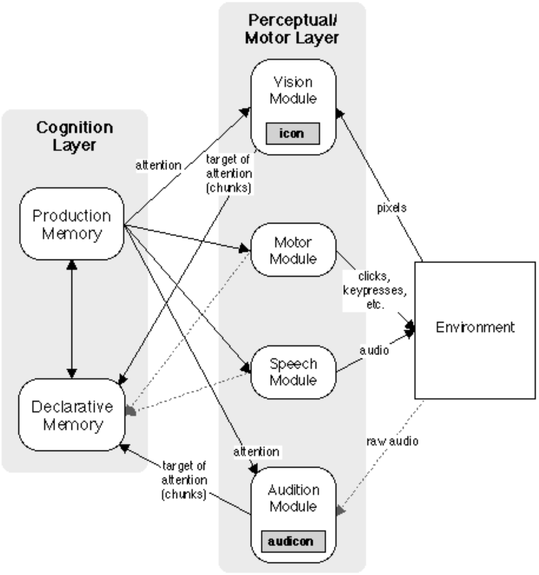

Figure 3: Overview of ACT-R/PM's modules and interaction with a simulated environment.

Audition Module in ACT-R/PM

The Audition Module is responsible for "hearing" in ACT-R/PM. When a sound occurs, a representation of that audio event is placed in the Audition Module's store, termed the "audicon." Events remain in the audicon for 3 seconds, after which they decay and can no longer be retrieved. The audio module is very similar to vision in that it contains two parts. These are essentially a "where" and a "what" system.

The "where" system is used to find a sound event and is referred to as aural-location. The production system requests a sound event and provides certain constraints, such as "earliest temporal onset that has not previously been attended." If a matching item is found, a representation of it is placed in ACT-R's "aural-location buffer." (Note that this is not the traditional use of the word "buffer." In ACT-R parlance, a buffer is simply the interface between the production system and peripheral system.)

The "what" system extracts information from sound events. Given a sound event from the "where" system, the "what" system shifts auditory attention to the event and produces a chunk which represents the content of the sound, such as the identity of a word or the frequency of a tone. After a content-dependent processing time this chunk is placed in the "aural buffer" and may be accessed by production rules.

The Speech Module in ACT-R/PM

The Speech Module in ACT-R/PM gives ACT-R the ability to simulate speech. Execution time for a "speak" command is dependent on the text to be spoken. The default duration is based on 0.15 seconds per syllable, and the number of syllables is computed based on the raw number of characters in the string divided by three. ACT-R/PM also “hears” what it speaks as the Auditory Module has access to the Speech Module's output.

Model Design

Positional vs. Serial Encoding

One of the major design decisions in producing the model was whether to encode chunks with a positional or a serial method.

The previously mentioned EPIC model utilizes serial encoding. Items are linked in a serial “chaining” method (Kieras et al., 1999). Each item points to the next item in the list. Recall fails when one item decays to the point that it is no longer retrievable, thus breaking the chain.

In contrast, our model encodes chunks in a positional manner. Chunks have a slot that takes values such as “first”, “second” and so on that determines its position in the list. There is strong evidence for doing so. Positional confusions and omissions are common in serial recall. It is hard to see how these can arise from a “chaining” method, yet are easily accounted for with our approach.

Chunk Structure

```
goal124
isa         Item
position    Second
Word        Sum
List        New-list
```

The chunk structure is outlined above. Items are encoded with a position, a word, and the list they are in.

Failure of Recall

Recall fails when a chunk decays below the retrieval threshold. The said chunk is assumed to be “forgotten” and so is omitted in subsequent rehearsals and recall.

Parameters

Most of ACT-R’s parameters were kept at their default values. Those that were manipulated were ACT-R’s retrieval threshold and the noise of the system. Both of these parameters do not have well defined default values and so were manipulated freely in order to provide a fit to the data. This resulted in values of .54 for the retrieval threshold and .3 for noise. These parameters factor in to ACT-R’s activation based system of memory as previously discussed.

Overview of the Model

The diagram below (figure 4) demonstrates how the model works once it has encoded the first item, until it is prompted to do recall.

In the first step in the looping cycle the model checks to see if there is a new item in its aural-location buffer. If there is nothing, it simply goes on and articulates the next item in the list. If there is something, it is attended to by the aural system. Since this process takes time, the model moves on and articulates the next item just as it would have if the aural-location buffer were empty.

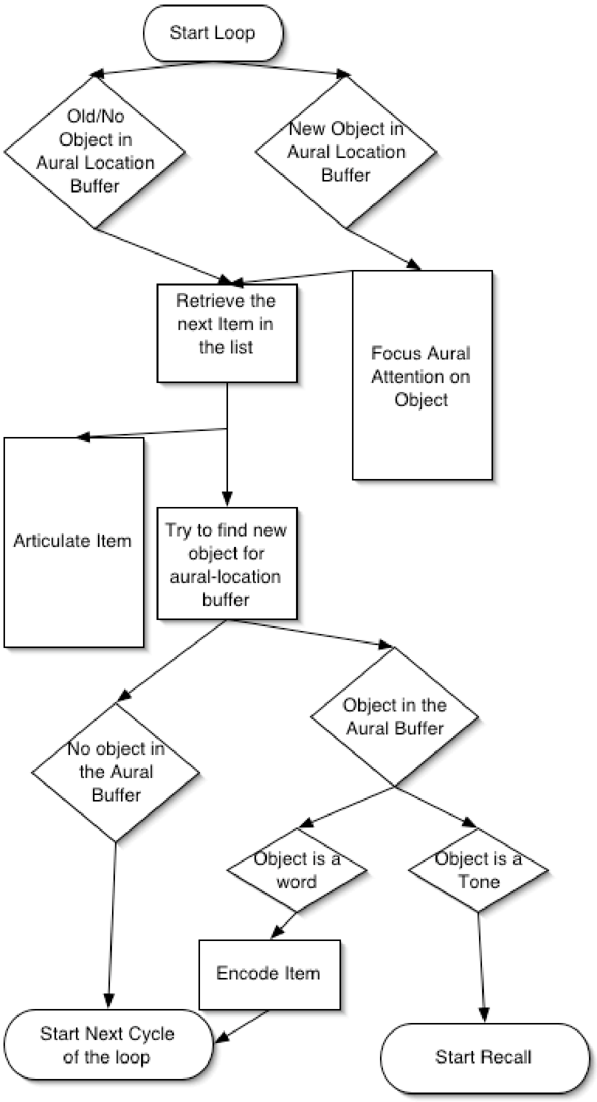
Start Loop
Old/No Object in Aural Location Buffer
New Object in Aural Location Buffer
Retrieve the next Item in the list
Focus Aural Attention on Object
Articulate Item
Try to find new object for aural-location buffer
No object in the Aural Buffer
Object in the Aural Buffer
Object is a word
Object is a Tone
Encode Item
Start Next Cycle of the loop
Start Recall

Figure 4: Overview of the functioning of the model.

While an item is articulated, the model looks in the audiocon for an unattended audio event. If this is found it is placed in the aural-location buffer.

Next the model checks the aural buffer. If nothing is present the looping process starts over. If something is present its kind is checked (i.e. sound or tone). If the object is a tone, recall is initiated. If it is a sound, it is encoded and the looping cycle repeats itself. This process occurs indefinitely until recall is prompted.

The process analysis above demonstrates several points that were made before. First is that the modules are working in parallel. Attending to aural items and vocalization are two processes that take considerable time in ACT-R/PM. Thus these two processes are overlapped to optimize the functioning of the loop.

Results

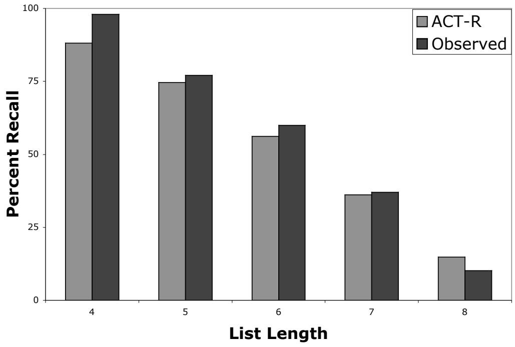

Figure 5: Comparison of the model's predictions to the observed data for short words

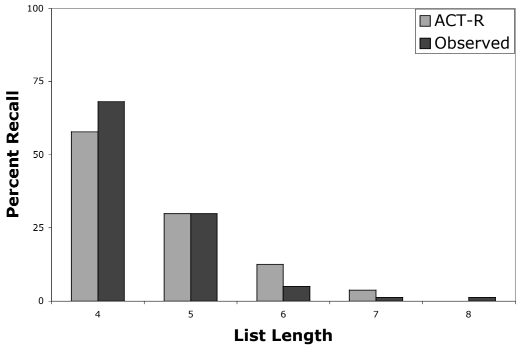

Figure 6: Comparison of the model's predictions to the observed data for long words

The model successfully produces quantitatively approximate effects of both list length and word length. Longer lists mean items are articulated less often, and longer words take longer to articulate, so both degrade performance.

As the graphs show, the model underestimates the percent recall of short lists and overestimates the percent recall of longer lists, regardless of the word pool. Thus it is clear that list length effects are not being as well construed as they might have been.

It is believed that the addition of partial matching, resulting in positional confusions, may result in a better fit to the data. This may be implemented in future revisions.

Serial Position Curve

Out of curiosity, a serial position curve was made from the data collected for the short duration words. No such data was collected in the original experiment, making the comparison between the two impossible. However the data qualitatively yields a serial position curve. The words in the initial and final positions have better recall than those in the middle. Such a curve is characteristic of serial recall data and points to the robustness of our model.

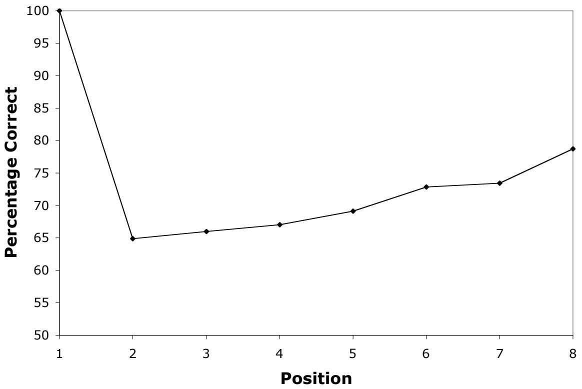

Figure 7: A serial position curve from the model

Comparison to the Epic Model

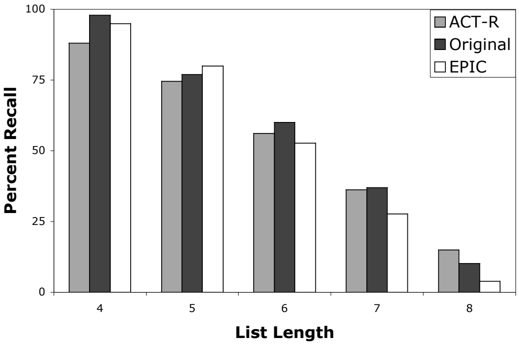

Figure 8: The performance of the EPIC and ACT-R models for short words

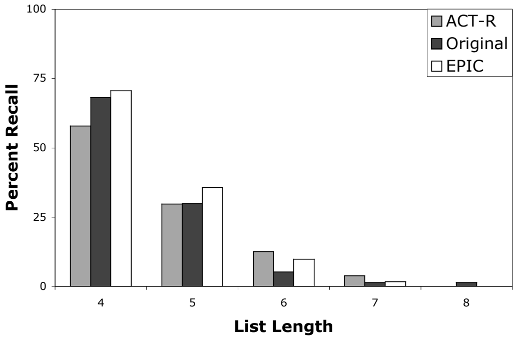

Figure 9: The performance of the EPIC and ACT-R models for long words

Despite the different architectures and modeling approaches, performance for our model was comparable to the EPIC model.

Table 1: Fit Metrics for the EPIC and ACT-R Models

		r-squared	RMSD	Percentage Average Absolute Error
ACT-R	short	0.99	5.23	13.51%
	long	0.97	5.81	90.65%
EPIC	short	0.98	6.30	21.35%
	long	0.99	3.63	49.75%

Overall, the two models produced quantitatively similar fits. This is potentially surprising for two reasons. First, no structural modifications were made to ACT-R to achieve this fit, while the EPIC model makes use of a similarity-based decay mechanism engineered specifically for this model. Second, the EPIC model makes use of more exact articulation times, while our ACT-R model uses the rather crude approximation outlined earlier.

Accounting for the Serial Memory Effects

Several serial memory effects were previously mentioned as evidence for the articulatory loop. A discussion of how these are or can be accounted for in our model is particularly relevant.

Time of Articulation

The Speech Module calculates time of articulation in a rather rough method as previously described. However this method is enough to demonstrate word length effects, as shown by our model's performance.

List Length Effect

Like one would assume, in longer lists words are articulated less often. This is what happens in the model. This leads to the failure of retrieval of chunks and thus decreased performance for longer lists.

Unattended Speech

Currently the model does not demonstrate the unattended speech effect. The aural module distinguishes between externally generated words and internal ones. Confusions are impossible and external stimuli would not degrade performance.

Phonological Similarity

Through the implementation of ACT-R/PM's partial matching, effects of phonological similarity could be modeled. For the sake of simplicity, our current model does not make use of partial matching and so does not demonstrate effects of phonological similarity.

Articulatory Suppression

The model can show effects of articulatory suppression by simply turning off its articulation. This will make it unable to implement an articulatory loop and thus degrade its performance on a serial memory task.

Conclusion

We believe the model suffices as a proof of concept for articulatory rehearsal in ACT-R.

The design of the model allows for a fair amount of expandability. It successfully produced a serial position curve, despite not being fitted to do so. The addition of partial matching should allow for acoustic and positional confusions. Such additions could make it useful for a larger variety of serial memory tasks.

The model also performed on par with the EPIC model. We believe this is quite an accomplishment.

Furthermore, we have striven to build the model in a flexible manner so that articulatory rehearsal can be built into other ACT-R models of more complex tasks. Such an inclusion will ensure that the unique effects of verbal working memory are present in more complex tasks, leading to both more accurate and plausible models.

References

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

Baddeley, A. D., & Hitch, G. J. (1974). Working memory. In G. Bower (Ed.), Recent advances in learning and motivation, Vol. VIII (pp. 47-90). New York: Academic Press.

Baddeley, A. D., Thomson, N., & Buchanan, M. (1975). Word length and the structure of short-term memory. *Journal of Verbal Learning and Verbal Behavior*, 14, 575-589.

Baddeley, A. D., Lewis, V., & Vallar G. (1984). Exploring the articulatory loop. *Quarterly Journal of Experimental Psychology 36A,* 233-252.

Byrne, M. D., & Anderson, J. R. (1998). Perception and action. In J. R. Anderson & C. Lebiere (Eds.), The atomic components of thought (pp. 167-200). Hillsdale, NJ: Erlbaum.

Card, S. K., Moran, T. P., & Newell, A. (1983). The psychology of human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum Associates.Colle, H. A., & Welsh, A. (1976). Acoustic masking in primary memory. *Journal of Verbal Learning and Verbal Behavior,* 15, 17-32.

Conrad, R. (1964). Acoustic confusion in immediate memory. *British Journal of Psychology,* 55, 75-84.

Kieras, D.E., & Meyer, D. E., Mueller, S. T., Seymour, T. L. (1999). Insights into working memory from the perspective of The EPIC Architecture for modeling skilled perceptual-motor and cognitive human performance, in A. Miyaki & P. Shah (Eds.), *Models of Working Memory*. New York: Cambridge University Press.

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

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits to our capacity for processing information. *Psychological Review*, 63, 81-97.