Endowing a Cognitive Architecture with World Knowledge

Dario D. Salvucci (salvucci@drexel.edu)
College of Computing & Informatics, Drexel University
3141 Chestnut St., Philadelphia, PA 19104 USA

Abstract

Although computational models developed in cognitive architectures are often rich in their knowledge of procedural skills, they are often poor in their knowledge of declarative facts about the world. This work endows the ACT-R cognitive architecture with world knowledge derived from Wikipedia, compiling a knowledge base of over 37 million declarative facts that can be accessed by a cognitive model via standard memory retrievals. Estimates of the accessibility of these facts are also derived from Wikipedia text, allowing ACT-R to utilize the likelihood of knowing a fact and associations between related facts. Integration with a simple procedural model demonstrates how the knowledge base may serve not only to answer simple factual questions, but also to disambiguate among multiple possible meanings based on context. The resulting knowledge base can be queried quickly (typically well under one second) and is easily generalizable to other cognitive architectures.

Keywords: Cognitive architectures; Wikipedia; ACT-R

Introduction

Cognitive architectures, particularly production-system architectures (e.g., Anderson, 2007; Laird, Newell, & Rosenbloom, 1987; Meyer & Kieras, 1997; Newell, 1990), have been used for a number of years as a computational framework for representing human cognition and behavior. Researchers have employed such architectures to model behavior in a large array of task domains. The vast majority of these models were developed with an emphasis on the procedural skills necessary to perform particular tasks; for instance, models have been developed to simulate behavior in the domains of piloting (Jones et al., 1999), game playing (Laird, 2002; Taatgen et al., 2003), and driving (Salvucci, 2006). At the same time, these models often have minimal declarative, factual knowledge; while they may include tens of facts to represent, say, the addition tables up to 9+9, they typically have little to no general knowledge about the world—for instance, what is the capital of Pennsylvania, or who invented the light bulb, or what sport is played by the Pittsburgh Steelers.

This project aims to develop a large-scale knowledge base that can easily be integrated into cognitive architectures to provide models with general world knowledge. Although past efforts have created large-scale knowledge databases (e.g., *Cyc*: Lenat, 1994; *Scone*: Fahlman, 2006; *WordNet*: Miller, 1995), these databases do not necessarily integrate easily with a cognitive architecture: they cannot be accessed in a straightforward way from a production system, nor do they include the cognitively plausible properties—such as the accessibility of knowledge elements—that some architectures rely on for modeling cognition (see Ball, Rodgers, & Gluck, 2004, for further discussion). More recent efforts to create knowledge bases for cognitive architectures (e.g., Douglass & Myers, 2010; Derbinsky, Laird, Smith, 2010; Emond, 2006) have explored the practical challenges inherent in such work, especially in understanding and reducing the computational demands of retrieving information from a large-scale database.

This project uses the Wikipedia knowledge base to derive a declarative database for the ACT-R cognitive architecture (Anderson, 2007), complete both with tens of millions of world-knowledge facts and with estimates of the accessibility (activation) of these facts. In doing so, the project addresses theoretical challenges (e.g., an appropriate representation of these facts) and practical challenges (e.g., computational efficiency) in a way that generalizes to other cognitive architectures beyond ACT-R.

Declarative Knowledge Base

Wikipedia [http://www.wikipedia.org] is the largest open body of general knowledge on the Internet today, with over 4 million articles in English alone, written by thousands of active contributors. Both its breadth of topics and its open licensing makes Wikipedia extremely amenable to use as a knowledge base for cognitive modeling. Unfortunately, the primary content of Wikipedia comes in the body of its full-text articles, and until cognitive architectures have large-scale robust natural-language capabilities, they cannot make direct use of such articles. Fortunately, other aspects of the Wikipedia knowledge base are available in representations that more easily interface with modern architectures.

Knowledge Content

The primary content for this work comes from the DBpedia [http://www.dbpedia.org] project, which extracts and disseminates structured representations of Wikipedia knowledge. Specifically, DBpedia makes available several large datasets that served useful in building a knowledge base for cognitive architectures. The datasets, and the resulting knowledge arising from them, are described here.

Relations. The first dataset includes information from Wikipedia "infoboxes" that appear alongside the full-text articles and provide knowledge in terms of relations. Table 1 shows the (partial) infobox for "Harrison Ford" as it

appears in Wikipedia, including basic information about his life and work. The DBpedia project extracts Wikipedia infobox content and cleans up these data based on the DBpedia ontology of objects, ensuring that key attributes are handled in a uniform way (e.g., all birthdates are translated to a common format associated with the attribute "birth date"). The cleaned version of this information is included in Table 1. This version comprises relations as object-attribute-value triplets: objects ("Harrison Ford") with attributes from the ontology ("spouse") and values for these attributes ("Calista Flockhart"). Note that, in some cases, two sets of triplets may encode redundant information (e.g., Harrison Ford's spousal relationship to Calista Flockhart). These data serve as the core knowledge for this effort, with a wide variety of object, attributes, and values. [Note that the triplets here are equivalent to a predicate-argument-value representation like *spouse(HarrisonFord) = CalistaFlockhart*.]

Table 1: Sample infobox and relation representation.

Infobox [Wikipedia]

Born:	July 13, 1942 (age 71) Chicago, Illinois, U.S.
Occupation:	Actor, producer
Years active:	1966–present
Spouse:	Calista Flockhart (2010–present)

Relation Representation [DBpedia]

Harrison Ford	isa	actor
Harrison Ford	isa	producer
Harrison Ford	isa	person
Harrison Ford	birth date	1942-07-13
Harrison Ford	birth place	Chicago
Harrison Ford	spouse	Calista Flockhart
Calista Flockhart	spouse	Harrison Ford
Star Wars Episode IV	starring	Harrison Ford
Raiders of the Lost Ark	starring	Harrison Ford
Witness	starring	Harrison Ford

Types. DBpedia also provides information about the ontology types of Wikipedia objects. In essence, these types can be thought of as the categories to which the objects belong—very much analogous to the "isa" relationship common to cognitive architectures and artificial intelligence frame representations. For example, "Harrison Ford" is listed as belonging to three categories ("actor", "producer", and "person") and thus these three "isa" relationships included in the object-attribute-value triplets in Table 1. This information is critical in providing the knowledge base with an understanding of object membership in categories.

Names. A third dataset available through DBpedia is the list of Wikipedia "redirects," whereby the entry of a particular name is redirected (forwarded) to a common page. Some of the redirects are intended for misspelled entries (e.g., "Harison Ford") or entries with variant spellings ("Muammar Qaddafi" or "Gadaffi" redirecting to "Muammar al-Gaddafi"). But the redirects also encode important differences in how people refer to common objects, such as nicknames ("Bill" for "William", "Big Apple" for "New York City"). The redirect database is incorporated into the knowledge base via a "name" attribute; for example, with "Jimmy Stewart" as the value of the *name* attribute for the object represented canonically as the symbol *James_Stewart_(actor)*.

The final knowledge base comprises 11,862,387 unique symbols and 37,100,782 facts (including all relations, types, and names as object-attribute-value triplets).

Representation and Implementation

The integration of the knowledge base into the ACT-R architecture brings up two important issues, one theoretical and one practical. The critical theoretical issue is one of representation. Retrievals of declarative facts, or so-called "chunks," in ACT-R arise from requests to the architecture's memory resource (see Anderson, 2007). ACT-R declarative chunks are typically modeled in representations such as:

Harrison Ford
isa actor
film Star Wars IV
spouse Calista Flockhart

When an ACT-R model retrieves such a chunk, it provides a partial pattern that specifies some or all of the attributes and values, and the memory resource chooses and returns the best-matching chunk (explained further in the next section). Although this representation works as needed for many domains (especially those that are not knowledge-intensive), it is less desirable than the previously described triplet representation in two ways. First, ACT-R chunks can only have one value for a particular attribute, and thus in the fact above, multiple films cannot be included in the same chunk representation. Second, retrieval of an entire chunk is overly powerful: once the model recalls one bit of information about the object (say, that Harrison Ford starred in *Star Wars*), it immediately has access to all bits of information (including the name of his spouse, his place of birth, etc.).

In contrast, the triplet representation can incorporate multiple values for an attribute, and can account for variations in the accessibility of knowledge for different units of information about a particular object. For these reasons, the current implementation of the proposed knowledge base stores and retrieves chunks in the triplet representation defined earlier—that is, each chunk is represented like the following:

Fact-1234
object Harrison Ford
attribute spouse
value Calista Flockhart

As a practical issue, the implementation of the knowledge database must allow for flexible queries that mimic the architecture's memory processing, must be fast enough to ensure reasonable simulation times, but must be sufficiently lightweight to facilitate portability and use across systems. For these reasons, SQLite [http://www.sqlite.org] was chosen as the back-end database for the project (most similar to Douglass, Ball, & Rodgers, 2009), and Java ACT-R [http://cog.cs.drexel.edu/act-r] was chosen as the front-end implementation of ACT-R. The integration with the ACT-R architecture strived to make the new knowledge base transparent to the cognitive model, in that retrievals were requested and processed in the normal way. The implementation in Java ACT-R uses a "extended memory" module to augment the standard memory module: when no chunk satisfying a retrieval request is found in ACT-R's standard memory, the system accesses the full SQLite database for retrieval. (In principle, all memory elements could be stored in the database; however, having a two-level approach with standard and extended memory greatly facilitates the code, and allows computation of chunk properties to be performed only on the recently created chunks in standard memory, described shortly.)

Estimation of Knowledge Accessibility

The core knowledge base described above contains a great many facts, but does not distinguish among them in terms of accessibility for an average person; the most commonly known people, places, and so on (e.g., Bob Dylan, Muhammad Ali, New York City) are not treated any differently than the (much more numerous) scarcely known people, places, and so on. In contrast, facts in the human memory system may be more or less accessible, and we would like the computational knowledge base to reflect this feature. Of course, individuals may differ themselves with respect to accessibility of particular knowledge: most people may be able to name, say, only a few Civil War battles, whereas a history buff might be able to name a great many more. To maintain simplicity for this first effort, the proposed knowledge base aims to represent the accessibility of knowledge for an "average" person.

Accessibility of knowledge, when instantiated in the ACT-R architecture, can be broken down into two primary components: *base-level activation* representing general accessibility, and *associative activation* representing accessibility based on the current task context. For the proposed knowledge base, both quantities are derived from Wikipedia's infobox link structure, using links as a surrogate for the strengths of, and relationships among, knowledge elements. Each component is detailed below.

Base-Level Activation. For each factual chunk, ACT-R maintains a base-level activation that represents the chunk's general accessibility: a chunk with a higher base-level activation is more accessible than another with a lower base-level activation. For example, the chunk representing a well-known musician (e.g., Bob Dylan) would, for most people, have a higher base-level activation than a chunk representing a less widely known musician. ACT-R posits that base-level activation changes as that chunk of information is used, or neglected, over time. Specifically, the base-level activation B of a concept can be approximated as follows (Anderson, 2007):

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

In this equation, n is the number of times the chunk has been used (i.e., created or retrieved by the memory system); L is the lifetime of the chunk (the time since chunk creation); and d is a decay parameter. We assume that L has a constant value for all chunks in the knowledge base (i.e., that they were all created at roughly the same long-ago time), and because all computations in the remainder of this paper will only need to compare chunks, we ignore the constant second term in the equation. In addition, we assume the ACT-R default value of 0.5 for d. Thus, the equation simplifies to:

$$B = \ln(2n)$$

The knowledge base assumes that Wikipedia links can serve as a conceptual surrogate to chunk usage in ACT-R. Specifically, the number of links to a particular Wikipedia concept can be treated as roughly proportional to the number of times a person would encounter and recall the chunk associated with that concept (e.g., the number of times a person would encounter a thought or perceptual input about "Bob Dylan"). Thus, for a given chunk relation, we set n to the number of times the relation's object appears in the triplet slots of any chunk, and compute B using this value. For example, the base-level activation of the sample chunk *Fact-1234* shown earlier (representing that Harrison Ford's spouse is Calista Flockhart) would be set according to the number of times the symbol *Harrison_Ford* appears in all chunks. This process makes an assumption that each chunk with a given object (such as *Harrison_Ford*) is equally accessible. Of course, there are several ways in which this assumption might not be accurate—for instance, Harrison Ford's birth date or birthplace may not be as widely known as his spouse.[1] Nevertheless, the assumption provides a good baseline for accessibility, as demonstrated in the upcoming examples.

Associative Activation. Whereas base-level activation represents a chunk's overall accessibility, ACT-R also posits that a chunk can receive additional activation from associated chunks in the current task context. In ACT-R, "context" is defined as the other chunks in the processing buffers, especially those in the "imaginal" buffer that serves as a working scratchpad of information for the current task. First, we define a strength of association S_{ji} between symbols i and j as

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

[1] Note also that the accessibility of the object and value cannot be combined; many people familiar with both Harrison Ford and Chicago may not know that the actor was born in Chicago.

where fan_j is the number of other chunks that contain symbol j in one of its slots. For example, the symbol *Chicago* would have a relatively high *fan* value, since, as a populous and popular city, it is referenced in a relatively large number of other chunks. S_{max} represents a value larger than all values $\ln(fan_j)$ (currently set at 20).

When attempting to retrieve a relation chunk, the system first identifies all potential matches for the given pattern and sets their initial activations as their base-level activations described earlier. Next, it spreads associative activation from the current context: for all symbols j in the imaginal buffer, if any potential matches contain a symbol i for which S_{ji} is non-zero, the value S_{ji} is added to its activation. For example, *Harrison_Ford* and *Chicago* appear in the same chunk; therefore, if *Chicago* appears in the current context, it will spread activation to any potentially matching chunk that includes *Harrison_Ford*.

Procedural Knowledge

Although the declarative knowledge base is the focus of this work, we require procedural knowledge to demonstrate how the declarative knowledge can be retrieved in realistic and useful ways. To this end, this work includes a cognitive model with a simple production system that understands and responds to basic questions about common facts. The model takes a similar approach to earlier work on sentence processing in ACT-R (e.g., Anderson, Budiu, & Reder, 2001; Lewis & Vasishth, 2005).

The model parses and responds to a given question as follows. First, the model listens to a question word-by-word, and when encountering a lexical item (word or logical phrase) through vision or audition, the system associates the item to a semantic symbol by retrieving a *name* relation chunk; for instance, the phrase "Harrison Ford" initiates a retrieval for the symbol with that name, that is, the symbol *Harrison_Ford*. Note that often, a single phrase can map to different symbols, such as "New York" to the city or the state; both base-level activation and associative activation play a critical role here in resolution of ambiguity, as seen in the examples shortly.

Second, the model places the found symbol into the imaginal buffer in a slot associated with its role in the sentence (e.g., subject, verb, object). This basic parser does not attempt to form a parse tree, but rather fills out a simple flat structure with the noted grammatical elements. When the entire question has been encoded, the model performs a retrieval to answer the question based on the structure of the question; for example, "What is the capital of Pennsylvania?" would eventually lead to a retrieval request for a chunk with object *Pennsylvania* and attribute *capital*. Again, as for retrieval of a lexical item's semantic symbol, retrieval of the question's answer is guided by both base-level and associative activations. The model uses the retrieved chunk to respond verbally to the question. Because some questions have multiple answers, the model will attempt a few additional retrievals (suppressing recently retrieved items) and generate those responses as well.

One useful way to understand the interactions of declarative and procedural knowledge in the model is to examine the behavior of the whole system for illustrative examples. We present a number of examples below.

"What is the capital of Pennsylvania?"
For this straightforward question, the model processes each word in order, mapping each to an appropriate semantic symbol and finally attempting to retrieve a relation chunk with object *Pennsylvania* and attribute *capital*. The correct answer is successfully retrieved and used to generate a spoken response to the question ("Harrisburg").

"What is Philadelphia?"
This deceptively simple question illustrates the workings of the base-level activations in the knowledge base. Although most people would associate "Philadelphia" with the city in Pennsylvania, this term can refer to other things as well; in fact, the knowledge base contains 8 possible mappings of this term (to Philadelphia, NY or IN; to the film or magazine with this name; and so on). The base-level activation of Philadelphia, PA, however, is more than twice that of any of the other interpretations, and thus the model retrieves its semantic symbol as the assumed interpretation and responds with its *isa* properties ("city", "place", etc.).

"Name a musician."
This open-ended request also demonstrates the importance of base-level activation in the model's responses. Although there are 37,872 musicians identified in the knowledge base, most are not familiar to most people. Guided by base-level activation, however, the model's first responses are well-known musicians (though certainly their exact ordering is debatable and would realistically be variable among individuals): "David Bowie", "Prince", "Bob Dylan", "Kanye West", and "James Brown".

"What is the population of Philadelphia?"
"Who is the director of Philadelphia?"
Because the model processes lexical items in order, items encountered earlier in the sentence can guide interpretation of later items because of associative activation from the current context. In the first question above, the term "population" spreads associative activation to the city "Philadelphia", although as noted in the previous example, this interpretation is already the dominant one because of base-level activation. In the second question, the term "director" spreads activation to a different interpretation, that of the film; this associative activation, when added to base-level activation, makes the film interpretation more active than the city, and *Philadelphia_(film)* is retrieved as the semantic symbol for this term. As a result, the model answers each question correctly ("1,526,006" and "Jonathan Demme" respectively).

"Who is the author of No_Country_for_Old_Men?"
"Who is a star of No_Country_for_Old_Men?"
There are many examples for which associative activation helps in understanding and responding to a question. The examples above demonstrate the resolution of an ambiguity

with respect to the book versus film version of "No Country for Old Men." The term "author" activates the correct response for the book ("Cormac McCarthy"). The term "star" activates the film interpretation, and the model in fact generates three responses in the order of their base-level activations ("Tommy Lee Jones", "Javier Bardem", "Josh Brolin")—a measure of their overall familiarity (quantified by references within the knowledge base) as opposed to the importance of their roles within the film (which is not encoded in any way in the knowledge base).

"What actor is a star of Airplane?"
"What athlete is a star of Airplane?"
In these two examples, associative activation from the context guides both the retrieval of semantic information and retrieval of the response itself. The term "star" helps the model disambiguate the meaning of the term "Airplane", mapping this term to the film *Airplane!*. When the model attempts to retrieve a chunk for a star of *Airplane!*, the terms "actor" and "athlete" appear in the current context (in ACT-R terms, in the imaginal buffer) and spread associative activation to particular responses. Thus, the term "actor" guides the response to those identified as actors in the knowledge base ("Robert Hays", "Leslie Nielsen", etc.), while the term "athlete" guides the response to the prominent athlete in the film ("Kareem Abdul-Jabbar").

"What is Jackson the capital of?"
"What film is Robert_De_Niro the star of?"
Because of the flexible nature of the triplet representation, the model can retrieve responses from the attribute and value just as well as from the object and attribute. The two questions above are reversals of earlier examples, providing the city (instead of the state) and the actor (instead of the film). In both cases, the model is able to retrieve the same relation chunks and respond to the questions ("Mississippi" and "The Godfather Part II", "Taxi Driver", etc.).

"Where is the Baseball_Hall_of_Fame?"
"Who is Theodore_Geisel?"
The inclusion of names and aliases in the knowledge base is critical in that it allows the model to understand commonly used aliases for semantic items. The term "Baseball Hall of Fame" maps to its canonical representation *National_Baseball_Hall_of_Fame_and_Museum*, yielding the correct response ("Cooperstown, New York"). Similarly, "Theodore Geisel" maps to his more commonly recognized alias, "Dr. Seuss", and the model responds accordingly ("cartoonist", "writer", etc.).

"What actor was born in Philadelphia?"
"What musician was born in New_Jersey?"
Again we see the role of the various activations at play in these examples: context helps to retrieve the appropriate semantic item for "Philadelphia"; associative activation guides the response to an actor or musician; and base-level activations guide the response to the most familiar names ("Bill Cosby" as the top response for the first question, "Bruce Springsteen" for the second).

General Discussion

Whereas most efforts related to cognitive architectures like ACT-R have focused primarily on procedural knowledge, the work here aims to develop a usable large-scale declarative knowledge base for easy integration with existing models. From a theoretical standpoint, this work is somewhat atypical in that it does not compare data directly to human behavior and performance. Nevertheless, its important theoretical contribution is the demonstration that ACT-R's memory constructs—specifically base-level and associative activation—scale well to very large knowledge bases. Although ACT-R's base-level and associative activation calculations have always been assumed to operate over the entire span of memory, the vast majority of models include only tens or hundreds of chunks, a number too small to thoroughly test this assumption. In contrast, the work here calculates base-level and associative activations from a more realistic set of tens of millions chunks, and demonstrates that such activations can guide cognitive processing to produce reasonable interpretations of questions and generation of familiar responses. This is a critical step for cognitive architectures: as architectures gain in their ability to learn and expand their procedural knowledge base (e.g., Salvucci, 2013; Taatgen, 2013), they will require an equally powerful declarative knowledge base with which to reason about the world.

From an engineering standpoint, this work aims to show that modern architectures can successfully simulate cognition and behavior with large-scale knowledge bases. There have been several efforts to incorporate large-scale knowledge into production-system cognitive architectures (e.g., Ball, Rodgers, & Gluck, 2004; Douglass, Ball, & Rodgers, 2009; Douglass & Myers, 2010; Emond, 2006); the work here represents by far the largest effort to date (over 37 million chunks). In addition, some efforts have focused specifically on real-time performance of memory retrieval mechanisms (e.g., Derbinsky, Laird, Smith, 2010; Douglass, Ball, & Rodgers, 2009; Douglass & Myers, 2010; Laird, Derbinsky, Voigt, 2011). Although real-time speed is not the primary goal of the work here, it should be noted that the model performs almost all retrievals in well less than one second. (Real-time latency is primarily a function of the number of potentially matching chunks, so only open-ended retrievals like "isa musician" typically take more than one second of real time.) Another benefit of the current knowledge base is its portability to other architectures: the flexible representation of knowledge and implementation in a commonly used database format greatly facilitate use and extension by other frameworks.

Moving forward, one of the major challenges with this effort is in more flexible processing of concepts and more general natural language understanding. For example, one can imagine cases in which the first retrieved interpretation for a lexical item is not the correct one, and the model might re-retrieve alternative interpretations; such an extension could be incorporated into the procedural knowledge for helping to disambiguate sentences. More generally, a more

complex inference engine as embodied in other systems (e.g., Bello & Cassimatis, 2006; Cassimatis, 2006; Lenat, 1995) has not yet been attempted here, and a translation of these ideas into ACT-R would be a difficult but worthwhile effort to make further use of the large-scale knowledge base.

Acknowledgments

This work is supported by Office of Naval Research grant #N000140910096.

References

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

Anderson, J. R., Budiu, R., & Reder, L. M. (2001). A theory of sentence memory as part of a general theory of memory. *Journal of Memory and Language*, *45*, 337–367.

Ball, J., Rodgers, S., & Gluck, K. (2004). Integrating ACT-R and Cyc in a large-scale model of language comprehension for use in intelligent agents. In *Papers from the AAAI Workshop*, Technical Report WS-04-07, (pp. 19-25). Menlo Park, CA: AAAI Press.

Bello, P., & Cassimatis, N. (2006). Developmental accounts of theory-of-mind acquisition: Achieving clarity via computational cognitive modeling. In *Proceedings of the Twenty-Eighth Annual Conference of the Cognitive Science Society* (pp. 1014-1019).

Cassimatis, N. L. (2006). A cognitive substrate for achieving human-level intelligence. *AI Magazine*, *27*, 45-55.

Derbinsky, N., Laird, J. E., Smith, B. (2010). Towards efficiently Supporting large symbolic declarative memories. *Proceedings of the Tenth International Conference on Cognitive Modeling*.

Douglass, S., Ball, J., & Rodgers, S. (2009). Large declarative memories in ACT-R. In *Proceedings of the 9th International Conference of Cognitive Modeling*.

Douglass, S. A., & Myers, C. W. (2010). Concurrent knowledge activation calculation in large declarative memories. In *Proceedings of the 10th International Conference on Cognitive Modeling* (pp. 55-60).

Emond, B. (2006). WN-LEXICAL: An ACT-R module built from the WordNet lexical database. In *Proceedings of the Seventh International Conference on Cognitive Modeling*.

Fahlman, S. E. (2006). Marker-passing inference in the scone knowledge-base system. In *Knowledge Science, Engineering and Management* (pp. 114-126). Springer Berlin Heidelberg.

Jones, R. M., Laird, J. E., Nielsen P. E., Coulter, K., Kenny, P., & Koss, F. (1999). Automated intelligent pilots for combat flight simulation. *AI Magazine*, *20*, 27-42.

Laird, J. E. (2002). Research in human-level AI using computer games. *Communications of the ACM*, *45*, 32-35.

Laird, J. E., Derbinsky, N., Voigt, J. (2011). Performance evaluation of declarative memory systems in Soar. *Proceedings of BRIMS 2011*.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, *33*, 1–64.

Lenat, D. B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, *38*, 32-38.

Lewis, R. L., & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, *29*, 375– 419.

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

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, 38, 39-41.

Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In *Visual Information Processing* (pp. 283-308). New York: Academic Press.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors*, *48*, 362-380.

Salvucci, D. D. (2013). Integration and reuse in cognitive skill acquisition. *Cognitive Science*, *37*, 829-860.

Taatgen, N.A. (2013). The nature and transfer of cognitive skills. *Psychological Review*, *120*, 439-471.

Taatgen, N.A., van Oploo, M., Braaksma, J. & Niemantsverdriet, J. (2003). How to construct a believable opponent using cognitive modeling in the game of set. In *Proceedings of the Fifth International Conference on Cognitive Modeling* (pp. 201-206). Bamberg: Universitätsverlag Bamberg.