Received: 9 September 2017 | Revised: 5 October 2018 | Accepted: 20 October 2018
DOI: 10.1002/wcs.1488

WILEY WIREs COGNITIVE SCIENCE

ADVANCED REVIEW

ACT-R: A cognitive architecture for modeling cognition

Frank E. Ritter[1] | Farnaz Tehranchi[2] | Jacob D. Oury[1]

[1]College of Information Sciences and Technology, Pennsylvania State University, University Park, Pennsylvania

[2]Department of Computer Science and Engineering, Pennsylvania State University, University Park, Pennsylvania

Correspondence
Frank E. Ritter, College of Information Sciences and Technology, Pennsylvania State University, University Park, PA 16802.
Email: frank.ritter@psu.edu

Funding information
Office of Naval Research, Grant/Award Number: N00014-15-1-2275

ACT-R is a hybrid cognitive architecture. It is comprised of a set of programmable information processing mechanisms that can be used to predict and explain human behavior including cognition and interaction with the environment. We start by reviewing its history, which shapes its current form, contrasts and relates it to other architectures, and helps readers to anticipate where it is going. Based on this history, we then describe it as a theory of cognition that is realized as a computer program. After this, we briefly discuss tools for working with ACT-R, and also note several major accomplishments that have been gained by working with ACT-R in both basic and applied science, including summarizing some of the insights about human behavior. We conclude by discussing its future, which we believe will include adding emotions and physiology, increasing usability, and the use of non-generative models.

This article is categorized under:
Computer Science > Artificial Intelligence
Psychology > Reasoning and Decision Making
Psychology > Theory and Methods

KEYWORDS
ACT-R, cognitive architecture, human memory, modeling, simulation, unified theories of cognition

1 | INTRODUCTION

ACT-R is a theory of the mechanisms that make up cognition, a cognitive architecture. The theory posits a fixed set of mechanisms that use task knowledge to perform a task thereby predicting and explaining the steps of cognition that form human behavior. Thus, it is one example of a unified theory of cognition (Byrne, 2012; Newell, 1990). Currently, it also predicts the activation of brain regions used to generate behavior by using mechanisms that make use of procedural (how to do a task) and declarative (facts about the world) knowledge, and working memory as activation, to perform tasks.

For example, Salvucci (2006) has created a set of declarative and procedural knowledge about how to drive a car (e.g., what other cars and lanes are, and when and how to turn). He then added this knowledge to ACT-R that had its vision and motor systems connected to a car simulator. ACT-R when it applies this knowledge of how to drive then provides predictions about the knowledge set, showing that it is sufficient to drive a car. The architecture also provides predictions about how fast the knowledge is applied, when turns and stops are performed, and what parts of the brain are active at each point in time. Additional knowledge can be added about how to dial telephones (what and where are buttons, and how to retrieve a number from memory and then use the memory to push buttons). The combined model can be used to predict the effect of dialing on driving (Salvucci, 2009).

The ACT-R theory has evolved over the course of more than four decades, and so have the acronyms used to describe the theory. ACT-R stands for *Adaptive Control of Thought-Rational*, although ACT-R has also been referred to as *Atomic Components of Thought* (Anderson & Lebiere, 1998).

There are numerous reviews of ACT-R, many noted here. Therefore, in this study in addition to describing it as a theory, we start by briefly reviewing its history, which explains its current form and helps readers to anticipate where ACT-R is going. Similar to most reviews of cognitive architectures, we describe the theory and its structure (see Box 1 for key concepts), and how these are realized as a running computer program; we also briefly discuss tools for getting started and working with ACT-R and major accomplishments in both the scientific and applied science areas. This includes summarizing some of the insights about human behavior that have been gained by working with ACT-R. We conclude by discussing its future, which we believe will include emotions and physiology, usability, and the use of nongenerative models. We include explicit lessons for other architectures, but there are many implicit lessons as well.

2 | HISTORY

The history of ACT-R is worth reviewing briefly for several reasons. First, the predecessors and previous iterations of the theory continue to shape ACT-R's current form. Second, it shows ACT-R's evolution from early theories of cognition and the influence of contemporaneous cognitive architectures (see Box 2 for types). Third, we hope the history will help readers and future researchers anticipate where ACT-R is going. Finally, the progression shows how cognitive architectures can evolve. The progression from HAM to ACT-R 7, as well as research that influenced ACT-R, is summarized in Figure 1.

Every broad simulation tool has innate strengths attributable to its base simulation's original purpose (i.e., the phenomena that were modeled). For example, the HumMod physiology simulation was initially developed from a heart simulation before being developed into a unified model of human physiology “from birth to death” (Hester et al., 2011). HumMod's heart simulation is its most developed module. ACT-R fits this paradigm as well: ACT-R began as a model of human memory before being developed into a unified theory of cognition. As such, ACT-R is strongest when modeling memory.

Work based on the ACT-R architecture will, by convention, refer to certain canonical publications to reference a particular implementation of ACT-R. For example, the most recent versions of the theory, ACT-R 7 and ACT-R 6, will typically reference one (or both) of Anderson (2007) and Anderson and colleagues (Anderson et al., 2004). Accordingly, work with older versions like ACT-R 5 will use older reference points such as Anderson (1993). Our history is drawn from a combination of these primary sources, milestone publications related to the theory (e.g., GRAPES, an early implementation of ACT-R's memory system Sauers & Farrell, 1982), and information disseminated through the annual ACT-R Workshop. A brief review of the history can be found in Anderson's (2007) book, *How can the human mind occur in the physical universe?,* which we also use here.

2.1 | Predecessors and early variations

As a field, research into cognitive architectures is relatively young. Anderson (2007, p. 4) claims that the term itself was introduced by Newell (Bell & Newell, 1971). It is worth noting that upon definition of the term, some previous theories could be arguably described as cognitive architectures after the fact (e.g., Hull, 1952). However, the ACT-R theory can specifically trace its lineage to the Human Associative Memory (HAM) model of memory (Anderson & Bower, 1973). This memory simulation model is built around the distributed memory model presented by Hunt (1971). The distributed memory model proposed that information processing operated through buffers that coded and recoded information from outside stimuli into

BOX 1

KEY CONCEPTS

Cognitive mechanisms: Information processing structures that exist across time and tasks. A set of mechanisms define an architecture. In ACT-R these include a way to see the world, visual memory, working memory, procedural memory, and a way to act upon the world. These modules (which are called buffers in ACT-R) have characteristics, such as how quickly memory decays, what is passed from vision to cognition, and what actions can be taken between buffers.

Productions: If-then rules made up of a pattern to match (if) and actions to take (then). The structures that the patterns can take are defined by the architecture. In ACT-R these patterns match objects (or their absence) in the buffers that make up the mechanisms of ACT-R. These are used to represent procedural task knowledge.

Declarative memory elements: These are symbols that represent declarative memory objects that can have attributes and values. They can be linked together. They are used to represent facts about the world and can be used to represent internal state information, such as goals. They have symbolic components that are accessible to the model and subsymbolic information that moderates behavior and accessibility of the declarative memory. Accessing them increases their activation, which decreases the time to access them. They are also called chunks in ACT-R and cognitive psychology.

BOX 2

TYPES OF ARCHITECTURES IN WIDE USE

There is a wide range of cognitive architectures that have been created. There are several general reviews of architectures available. The first one was perhaps a *SIGArt* review edited by Laird (1991) that noted 38 architectures. Later reviews include the use of architectures in military simulations by Pew and Mavor (1998), which noted 4 major and over 7 minor. In their review, they examined mostly architectures developed in the United States and noted general areas for projects. Ritter et al. (2003) provide a further review of 7 architectures created outside the United States and specific projects. In an unpublished but available paper, Morrison (2003) generated a similar broad review of 19 architectures. Langley, Laird, and Rogers (2009) reviewed a selected set of 7 architectures and noted several more in an appendix. They included some summary lessons from these architectures and areas that they and most architectures are deficient (e.g., emotions, interaction). Samsonovich (2010) and his colleagues in the Biologically Inspired Cognitive Architectures (BICA) Society created an updated and structured review of that has had a website (http://bicasociety.org/cogarch) with details on a wide range of 26 architectures. More recently, Kotseruba, Gonzalez, and Tsotsos (2016) provide a summary of the history of architectures, noting 55 active architectures. To see which architectures are being actively used, examples can be found in the *Proceedings of the Cognitive Science Society's Annual Conference*, the *Proceedings of the International Conference on Cognitive Modeling*, and the *Proceedings of the Artificial General Intelligence (AGI) Conference*.

Cognitive architectures can be organized in several ways. One way is to group them into five categories based on the goals of the architectures and then the levels of their mechanisms: (a) Artificial intelligence (AI) and agent architectures, such as JACK (Busetta, Rönnquist, Hodgson, & Lucas, 1999) and JESS (Friedman-Hill, 2003), that use high-level knowledge structures such as plans and focus on performance alone; (b) symbolic architectures, which include previous versions of Soar (Newell, 1990) and Icarus (Konik et al., 2009), that use production rules (Newell, 1973) and care somewhat about the match to human cognition; (c) subsymbolic architectures, such as connectionist networks (e.g., Grossberg, 1999; O'Reilly & Munakata, 2000), where concepts are represented across multiple components or nodes that are organized in networks, and often but not exclusively care about matching human cognition; (d) hybrid architectures, which include both symbolic and subsymbolic components (Sun & Bookman, 1994), and aim to match human cognition; and (e) nongenerative architectures that do not generate behavior, such as IMPRINT (Allender, Archer, Kelley, & Lockett, 2005; Booher & Minninger, 2003) and PROCRU (Pew, 2007), that are intended to predict the time and capabilities required to perform a task. These are used as engineering design tools.

ACT-R best fits in the hybrid architecture group because it has rules and declarative memory that are symbolic, but it also has activations that modify how these components are related and used in a subsymbolic way. The current version of Soar (Laird, 2012) also has subsymbolic components.

useable chunks, and that these chunks were connected. Although somewhat shallow in its scope, HAM provided the foundational model on which to build a comprehensive theory of cognition.

2.2 | ACT theory (1973–1993)

Anderson first proposed the ACT theory in 1976 in the book *Language, Memory, and Thought.* ACT was a culmination of several avenues of concurrent research. The ACT theory implemented procedural memory and thus expanded the capabilities of the model to account for many more aspects of cognition. ACT expanded HAM to include procedural memory (from Newell's PSG), alongside HAM's declarative memory. Procedural memory is how to do a task. ACT also included the addition of subsymbolic quantities, that is, numeric activation values for each production rule (sometimes, simply "rule") and declarative memory element. These activation values enabled variation in how and when and how quickly memory elements were used (Collins & Quillian, 1969). Subsymbolic changes in knowledge are represented in smaller increments than whole symbols (e.g., changing the weight of association between two symbols). These mechanisms incorporated the spreading activation model (Collins & Quillian, 1972) for declarative memory and a similar strength measure for the production system. These were the precursors to work on mapping ACT-R processes to their neurological correlates (Anderson, 1976).

In the 1980's, the theory was moved into a more complete simulation that included general cognitive processing (ACT-Embodied), and then into ACT* (pronounced Act Star), which was proposed as a general cognitive architecture. Following the initial creation of the ACT theory, Anderson, Farrell, and Sauers (1984) formulated the Goal-Restricted Production System (GRAPES) to implement these latest developments to the theory. GRAPES is marked as the first true implementation of the ACT theory. GRAPES also marked the final direct contribution of the HAM model's underlying code in the form of the

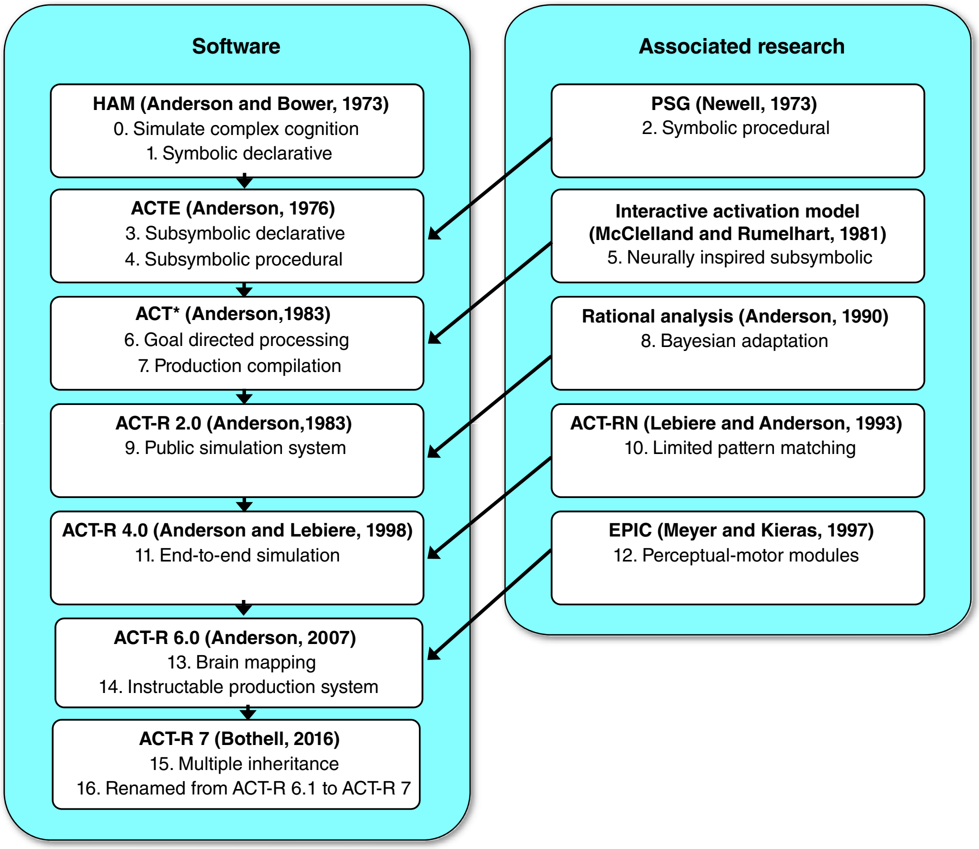


FIGURE 1 An illustration of the source of ideas and structures in current ACT-R. Figure based on fig. 1.11 in Anderson (2007) and extended to include ACT-R 7

declarative memory system derived from HAM. By acting as a knowledge interpreter built upon the Lisp environment, GRAPES incorporated a dynamic working memory and procedural knowledge bank with weighted values for the importance of different goals which allowed the system to model goal-oriented cognition with a tree structure (Sauers & Farrell, 1982). GRAPES was a more complete simulation of declarative and procedural memory (Anderson et al., 1984). The GRAPES model provided a goal-oriented model of human cognition that was used as a basis for computer-based tutoring systems.

GRAPES is also important for its implementation of the theory of knowledge compilation. Knowledge compilation is the process of integrating extended computation (multiple rule applications and/or memory retrievals) into a single production rule (Neves & Anderson, 1981). Anderson proposed that task memory was initially encoded as declarative knowledge through learning. However, upon repeated practice, the components of a task (e.g., the memories used to complete the task and the knowledge of what memories to retrieve, and what tasks to perform) are gradually linked together into a single procedural memory, or "compiled." This mechanism is proposed to be how people improve at a procedural task through repeated practice.

The embodiment of this learning mechanism in GRAPES in 1982 provided the basis for Anderson's next revision of the ACT theory: ACT*. ACT* integrated Anderson's (1983) work on knowledge compilation and McClelland and Rumelhart's (1981) Interactive Activation model. Furthermore, this new model also incorporated goal-directed processing within the goal module and a new model for production learning, both of which are still present within ACT-R today (Anderson, 2007).

2.3 | ACT-Rational: 1993–1998

Following the release of the ACT* theory, Anderson's work branched out into other applications. As noted in the preface for *Rules of the Mind* (Anderson, 1993), there was a period of uncertainty during the decade following the release of ACT*. During this period, Anderson revised his methodology for understanding cognition through the development of *rational analysis of cognition* (Anderson, 1990). This new methodology led to a significant expansion of the ACT theory. This work resulted in the updated theory, called ACT-R (ACT-Rational), which modified the memory and learning equations to more accurately incorporate use and learning from the environment.

Understanding ACT-Rational requires some explanation of rational analysis. The complexity of the human mind presents a significant "black box" problem, wherein researchers are forced to infer information processing mechanisms from behaviors. Rational analysis provides a solution through the assumption of optimality (i.e., rationality) of mechanisms; if multiple mechanisms are possible, the choice of optimal performance suggests which mechanism is most likely. This assumption of optimal mechanisms for human behavior offers a method for reducing the problem space by providing constraints on the gamut of plausible mental mechanisms (Anderson, 1990). The following sources provide further explanation: Chater and Oaksford (1999) provide a retrospective on the first decade of rational analysis. Anderson and Schooler (1991) provides an in-depth description of its foundation and methodology. Finally, Anderson (1993) provides an explanation of ACT-R, including the integration of the rational analysis methodology.

This addition, as well as including a usable copy of the program on a disc included with *Rules of the Mind*, stimulated the cognitive architecture research community and greatly increased the number of researchers using the theory. This led to the first ACT-R Workshop in 1994. Later, annual summer schools were organized to provide a teaching and learning conference for like-minded researchers to discuss their work with ACT-R (Anderson, 2007).

The next iteration, ACT-R 4,[1] released in 1998, made several key improvements to the theory and within the program itself as a tool for other cognitive researchers. First, it was presented as a more "polished" form compared to the 1993 release. By incorporating researchers' work outside of Anderson's lab, it promoted a more diverse range of views allowing for even greater collaboration. This version was heavily influenced by connectionism research at that time, and an updated pattern-matching algorithm allowed for increased power of the rule selection system. ACT-R 4 was also improved through the implementation of providing an "end-to-end" simulation tool with the ACT-R distribution—where the model sees the stimuli that participants see and can pass motor outputs into the same simulation participants use—this allowed human participants and cognitive models to interact with the same tasks, thus improving the ability of the program to be tested alongside human participants doing the same tasks (Anderson, 2007; Anderson & Lebiere, 1998).

During this time, other cognitive architectures continued to be developed. One such architecture, EPIC (Kieras & Meyer, 1997), ended up playing a crucial role in ACT-R's development. While sharing some underlying theoretical basis with ACT-R, EPIC used a highly modular structure that incorporated a perceptual-motor system meant to further "embody" the simulation. EPIC's perceptual-motor module would be the core addition of Byrne's ACT-R/PM (Byrne & Anderson, 1998), an offshoot of ACT-R concurrent to the release of ACT-R 4 (Meyer & Kieras, 1997). Figure 2 shows how a perceptual-motor layer was added to the cognitive layer to provide support interaction with the world (similar work was going on with Soar at the same time to provide simulated eyes and hands that were usable across models, Ritter, Baxter, Jones, & Young, 2000).

2.4 | ACT-R 5, 6, and 7: 1999–2018

The release of ACT-R 5 in 2001 once again offered significant advancements in the theoretical framework. As noted earlier, perhaps the most noticeable addition was a perceptual-motor module. This version also reduced the size of the goal module by splitting it into an imaginal module and a smaller goal module. The goal module maintains a chunk for state information and the imaginal module maintains a chunk for context information. In doing so, it brought the theory closer inline to the minimal inclusion of nonessential information in goals (Anderson, 2005; Anderson & Douglass, 2001).

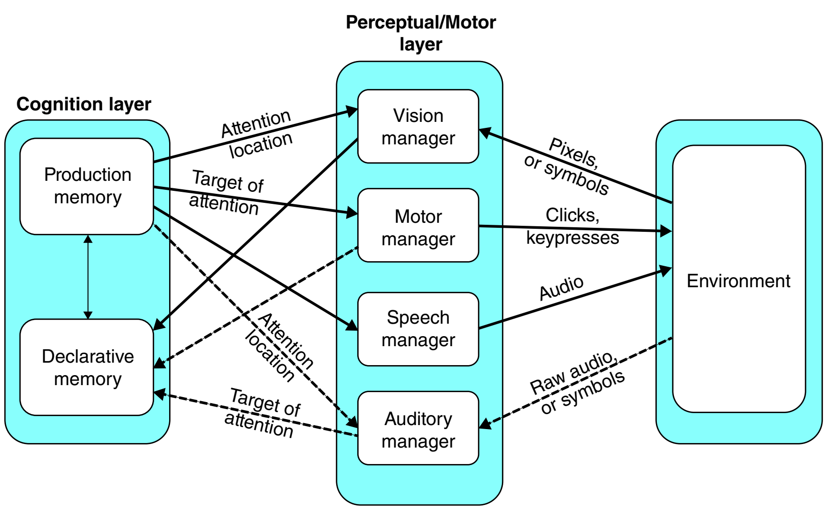


FIGURE 2 Schematic diagram of the perceptual-motor components for ACT-R, based on fig. 1 in Byrne and Anderson (1998)

ACT-R/PM was integrated into ACT-R 5 around 2002 (Fleetwood & Byrne, 2002). It provided a more accurate simulation of human perceptual-motor interaction, providing simulated eyes and hands (Ritter et al., 2000). The basic interaction theory was taken from the EPIC architecture. The software provided a way to interact with an instrumented Lisp window that was included with ACT-R. This work shows how architectures can influence each other.

The perceptual-motor module enhanced the implementation of subgoals within ACT-R by forcing researchers to include additional subgoals to use the perceptual-motor module. Behavioral research on the Tower of Hanoi, a common cognitive task, indicated that the number of subgoals has a significant role on performance of this task (Anderson & Douglass, 2001). Follow-up research on this effect using fMRI indicated a neurological basis for subgoal implementation and planning, specifically within the dorsolateral prefrontal cortex (DLPFC), bilateral parietal regions, and the premotor cortex (Fincham, Carter, Van Veen, Stenger, & Anderson, 2002). *An Integrated Theory of Mind* (Anderson et al., 2004) offers a detailed overview of the relationship between the modules of ACT-R 5 and its ability to predict and model complex behavior with neurologically compatible mechanisms that also predict brain activity.

ACT-R 6 was released in 2005, although it was first presented during the 2002 ACT-R summer school (Lebiere, 2002). Although very similar to ACT-R 5, it reinforced the modular design by delineating the procedural and declarative mechanisms as modules. It also more closely unified Byrne's Perceptual-Motor module with the architecture and implemented further perceptual/motor components. ACT-R 6 made it much easier for researchers to extend and modify different modules while also speeding up the program relative to ACT-R 5 (Bothell, 2015), and the community was encouraged to not refer to ACT-R/PM but just to ACT-R.

Over the next 10 years, ACT-R 6 was used as a stable base for yearly updates and improvements. In 2014, ACT-R 6.1 was released (Bothell, 2016a). This included some changes to make declarative memory chunk-types more useful by allowing for multiple inheritances, as well as some usability fixes. Specifically, memory chunks were given a “type” to identify their contents that improved matching and tying the objects to a theory. Following the 2015 Workshop, ACT-R 6.1 was renamed to 7.0 to alleviate some confusion regarding the numbering of the program. This led to ACT-R 6.1 being considered a “transient transitional version” to help with this (Bothell, 2016b). ACT-R 7 is the current version (http://act-r.psy.cmu.edu/software/).

2.5 | Summary

The history of ACT-R began in 1973 with a memory simulator called HAM and continues with ACT-R 7. It has undergone several revisions, and it is extensively documented and applied in several books and over 1,100 publications (most summarized at http://act-r.psy.cmu.edu). The workshops continue, allowing geographically dispersed researchers to gather and collaborate on how ACT-R can evolve. The community surrounding ACT-R provides a cadre of scientists who may not always agree, but all have the same goal: creating a model of the mind. As Newell (1990) noted at the beginning of his seminal work:

> Psychology has arrived at the possibility of unified theories of cognition—theories that gain the power of positing a single system of mechanisms that operate together to produce the full range of human cognition.
> I do not say that they are here. But they are within reach and we should strive to attain them. (p. 1)

Although nearly 30 years have passed since those words were written, it may seem that we are no closer to a unified theory of cognition. However, as we have seen over the 50 years of work on cognitive architectures, each new model and theory strives to integrate additional regularities and assimilate prior research. Whether ACT-R is the ideal theory for the creation of a unified theory is yet to be seen. While it may cover many aspects of cognition, there are many other theories that each have strengths and weaknesses. Progress may require the transition into different architectures and models to accommodate existing and future results, but the spirit remains the same. The next section explains the mechanisms in more detail (Boxes 3 and 4).

3 | THEORY COMPONENTS

The mechanisms that make up ACT-R are shown in Figure 3. These components consist of modules and buffers. Each module is responsible for processing a different kind of information. Modules are the mechanisms for modifying and implementing a buffer; buffers store contents that are visible to other modules.

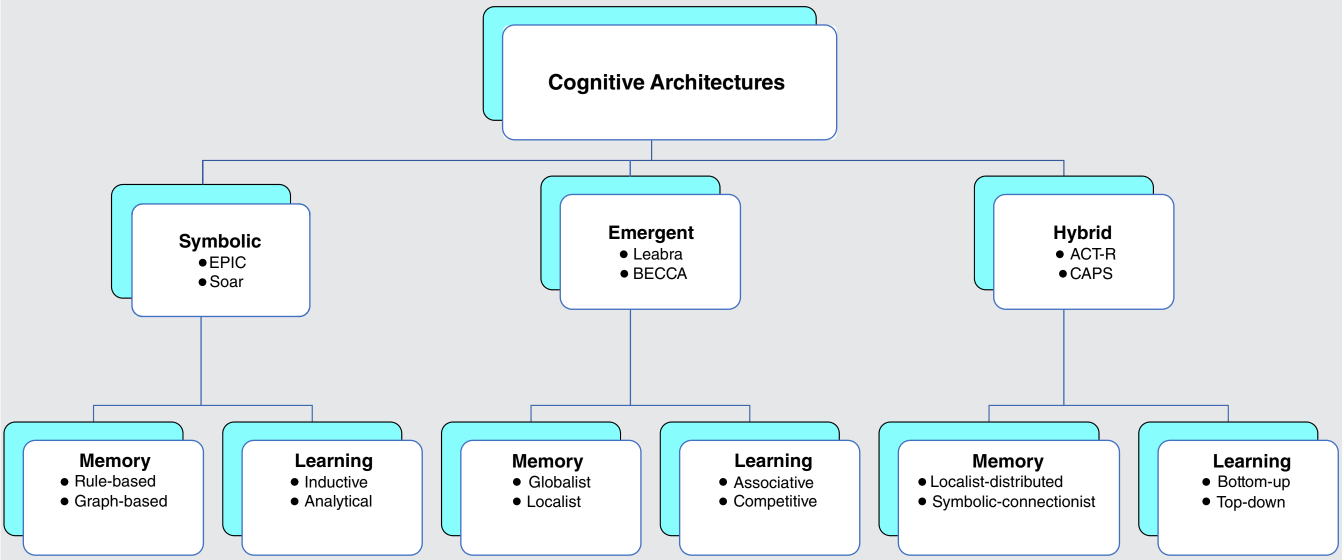
ACT-R has four main modules and a production system: (a) a visual module for identifying objects in the visual field, (b) a goal module for keeping track of current goals and intentions, (c) a declarative module for retrieving information from memory, and (d) a manual module for controlling the hands. A central production system coordinates the communication and performance of these modules through the application of production rules. This central production system is not sensitive to most of the activity of these modules unless the rule being applied is matching against a buffer. However, the central

BOX 3

ACT-R AND TYPES OF INFORMATION PROCESSING COGNITIVE ARCHITECTURES

Anderson's work on ACT-R aims to create a unified theory of cognition and implement this theory in a computational program. Broadly speaking, cognitive architectures and their underlying theoretical frameworks have been around since 1960. Newell and Simon (General Problem Solver, Newell, Shaw, & Simon, 1960) and Selfridge (Pandemonium, Selfridge, 1959) are some of the first researchers to have explored the idea of creating a computational architecture able to explain and predict the human mind. Simon worked with Feigenbaum, his student, to develop what may be the first cognitive architecture, the Elementary Perceiver And Memorizer (EPAM, Feigenbaum, 1961).

In the following years, many different cognitive architectures have been theorized and developed. A 2016 review of cognitive architectures found 195 different architectures with about 55 active architectures (Kotseruba et al., 2016). A wide range of architectures are available because each architecture addresses different issues, including topics such as memory, learning, problem solving, emotions, perceptual-motor control, development, language, engineering psychology, errors, and biological, neural, ease of use, and application issues.



Taxonomy of cognitive architectures based on how information processing occurs in them with examples. These are broadly distinguished by their use of top-down (Symbolic), or bottom-up (Emergent), or a hybrid of these two paradigms. Diagram is based on fig. 1 in Duch, Oentaryo, and Pasquier (2008).

The different architectures vary widely, but one commonly used heuristic for organizing them is the information processing and knowledge representation schemes that they use. The above figure presents one way to organize architectures, with architectures categorized as symbolic, emergent, or hybrid. Control in symbolic architectures is typically high-level, top-down, and analytic, while for emergent architectures it is low-level, bottom-up, and self-organizing. Hybrid architectures combine features from these two types (Duch et al., 2008; Sun & Bookman, 1994). In Box 4, based on a reviewer's request, we provide a comparison between ACT-R and Soar, the two most prominent and widely used cognitive architectures (Laird, Lebiere, & Rosenbloom, 2017).

production system has access to the buffers' data and will update the buffers by production rule applications. Thus, the shape, context, and complexity of the rules partially define and are constrained by the architecture.

The manual/motor buffer handles controlling and monitoring hand movement. Visual objects and their identities are located in the visual buffer. The visual and manual systems are the main components in defining interactive ACT-R tasks such as typing on a keyboard or scanning a computer screen.

Figure 3 also shows the current best mapping from the components to brain regions. Since ACT-R 6, researchers have attempted to map the components to brain regions (Anderson, 2007; Anderson et al., 2008; Anderson & Lebiere, 2003; Stocco, Lebiere, & Anderson, 2010). This has primarily been done as predictions of the regions' BOLD (blood flow) responses while doing a task.

Actions in each of these modules take time, although they work concurrently. The time used within models is based on human performance. This time can be provided as a real-time trace where the model runs at the rate predicted in real time; however, most models use a simulated time frame faster than real time. In either case, the predicted times appear in the trace.

BOX 4

ACT-R AND SOAR

ACT-R and Soar are two prominent cognitive architectures that are often compared and contrasted. However, although they are rooted in the overall idea of creating a unified theory, differences still exist in their approaches. Soar's current development is a result of AI-based research and thus is more theoretically based on assumptions of intelligent agents rather than behavioral phenomena (Laird, 2012; Ritter et al., 2003). Soar combines the heuristic search approach, procedural view of routine problem solving, and a symbolic theory of bottom-up learning designed to produce the power law of learning (Laird, Rosenbloom, & Newell, 1986). ACT-R, on the other hand, is strongly based on detailed experimental data of memory, learning, and problem solving.

There are also numerous similarities, however. Both have two memory types, declarative and procedural, with both architectures maintaining a limitless declarative and procedural memory (although, ACT-R's declarative memory can become difficult to access if the memories are not practiced). Finally, both systems use goal-directed problem solving, although Soar maintains multiple goal states in a hierarchy and ACT-R uses basically a single goal; both Soar and ACT-R are limited to a single, top-level goal.

Soar's decision procedure provides a fundamental difference from ACT-R. Soar uses a decision cycle, where the changing of states only occurs through a decision procedure dictated by the indirect application of rules. Once no more productions apply, the operator is selected or a state is modified. When unable to select an operator, subgoal creation of choosing the next operator occurs. ACT-R has a more direct state change mechanism through the firing of productions to alter the state or goal stack. Rule firing also differs between the two, while both examine which rules to fire in parallel, Soar fires rules in parallel and ACT-R in serial.

Soar's learning mechanisms also contrast with those in ACT-R. Soar learns in several ways (Laird, 2012). Perhaps the primary way is within the production memory through the creation of new rules upon resolving subgoals (called chunking in Soar), allowing for the learned rule to become the new standard rule for some situations. Soar also learns to compare operators using reinforcement learning and also learns semantic and episodic knowledge, but these are symbolic objects, not objects that have variable activations. Finally, Soar's new rules are immediately available for use (providing in some cases one-shot learning), while ACT-R's new rules start with a specified strength (utility) that must be strengthened (learned multiple times) to compete successfully with the rules that gave rise to it.

ACT-R learns in both declarative and procedural memory through the strengthening of chunks when accessed and rules upon firing. Productions are rated on their probability of success, cost, and current goal's value, allowing for conflict resolution when choosing between possible productions. These differences lead to ACT-R being considered less reactive than Soar, but more cognitively plausible. Thus, Soar tends to be used for larger systems (soar-ware engineering) for AI applications, and ACT-R tends to be used for simulating detailed human behavior.

The central cycle in ACT-R is one that starts with the buffers holding representations determined by the external world and the internal representations. Patterns in these buffers are recognized using production rules. A production that matches working memory is selected and applied, and the buffers are then updated for another cycle. The assumption in ACT-R is that this cycle takes about 50 ms to complete—this estimate of 50 ms as the minimum cycle time for cognition has emerged in a number of cognitive architectures including Soar, where a value between 30 and 300 ms is used (Newell, 1990, p. 121, 224). Actions requested from other buffers typically take longer. The manual notes the equations that are used to generate the predicted times to perform actions and how learning occurs through rule learning and changes to chunk activation. The parameters to these equations can be used to model individuals' differences and to explore how the architecture works.

Figure 3 notes several systems. In addition to the systems explained below, the schematic diagram also includes an Aural system that, like the visual system, tracks location and object. The Temporal buffer and module provide time estimations (Taatgen, Rijn, & Anderson, 2004) that can also be used for scheduling when multitasking. The Intentional system includes an Imaginal buffer that is used as a scratch pad for holding images and a goal buffer.

3.1 | Visual

Earlier ACT-R models did not always model the perception and motor aspects of behavior (and some still do not). These models do not include the perceptual-motor components either because they have been difficult to implement or because these

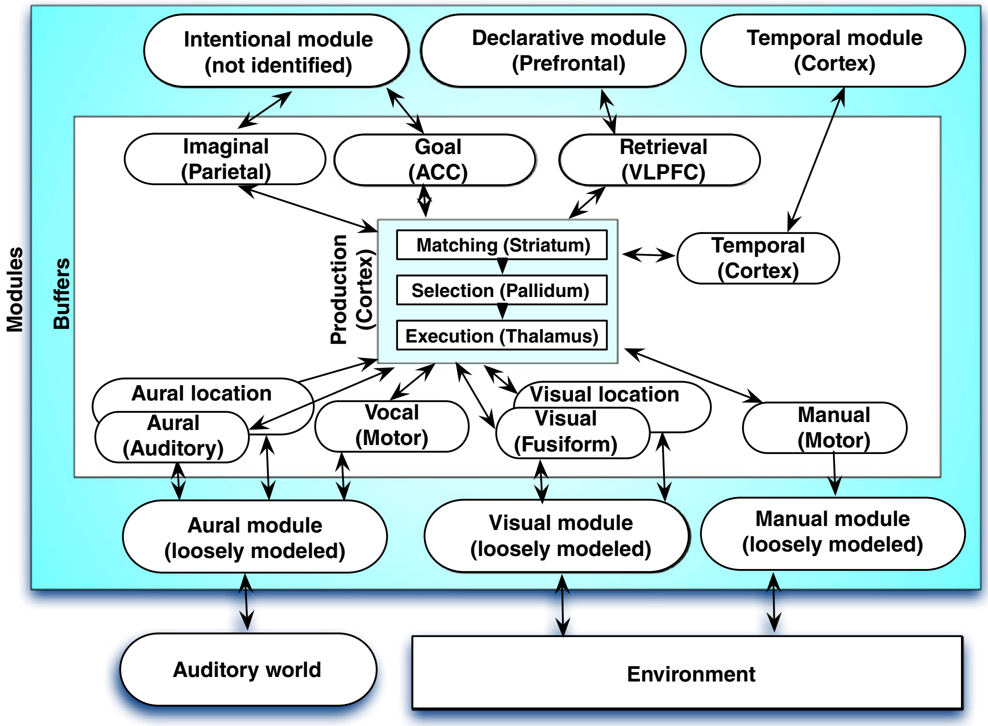


FIGURE 3 Schematic diagram of the ACT-R cognitive architecture and how its components work together to generate behavior. (Figure by authors based on and extending Anderson, 2007 and Anderson et al., 2004, the current ACT-R 7 manual, and comments from Bothell)

aspects are seen as not necessary for insights (e.g., interaction in the task is either trifling, does not lead to learning, or both). Some of the difficulty has been removed by this system and the motor system.

Much of the interaction with the world is done with the visual and motor system.[2] ACT-R's vision module, shown in Figure 3, includes two buffers, one for "Where" information (location) and one for "What" information (visual). The "What" buffer holds a chunk that represents an object in the visual scene, while the "Where" buffer holds the object's location. These two systems are analogous with the dorsal ("Where") and ventral ("What") visual streams in the human visual system (Anderson et al., 2004). The "What" buffer information is only available if the system focuses on (attends) to a location.

The visual system can attend to a screen location and move its attention around the scene, which is a simulation of fixations and eye movements, and in some systems, lock the eye onto an object so that smooth pursuit eye movements can be performed. This results in chunks representing the location of objects and related chunks holding the semantic context of the object. These visual chunks are then available to the central production system to reason about and lead to behavior, for example, shifting visual attention.

These perception modules are routinely expanded by researchers to accommodate other capabilities like hearing. These results can be included with the distributed architecture after they have been used and tested several times.

The perceptual and motor components work with an instrumented Lisp window included with ACT-R. Modelers interacting with other systems have created extensions to work with other tasks. For example, SegMan (for segmentation/manipulation) has been developed to translate pixel-level input from a computer screen into the objects and symbols that a cognitive model can manipulate. SegMan further translates the manual output of a cognitive model, such as mouse movements and key presses, into a form that can be executed appropriately by a computer (St. Amant, Riedel, Ritter, & Reifers, 2005; Tehranchi & Ritter, 2018).

3.2 | Central production system

During the decision cycle in ACT-R, production rules are used to analyze patterns in all of the buffers that may represent internal and external information. This pattern recognition system determines the next action. While the rules do not have direct access to the modules themselves, they can check some limited state information that has been made available by each module. Production rules consist of a condition and an action, which are also sometimes referred to as the left- and right-hand sides of the rule. The pattern structures that the rules can match represent architectural commitments about representations in the buffers and in the simulated mind.

There is a conflict-resolution phase in the rule-based system; a trace of this appears in the ACT-R output trace noting how rules are chosen. In ACT-R, the rules that match the current goal are matched against the central set of buffers connected to the central processing. A rule is selected based on this match (highest utility with noise added), and the action of the rule is applied (the rule "fires," "matches," or is "applied").

The rule can have multiple actions, and these can pass commands to the perception or motor module, or they can modify any buffer's values, updating a declarative or goal memory or putting a command into the visual or motor modules.

3.3 | Goal module

The goal module is the simplest module. It provides the system with a goal buffer that is typically used to maintain a model's current task state. The goal buffer also serves as a source of activation for declarative memory retrievals.

It has one buffer named Goal that holds a chunk containing the current control state. The only actions that this module can do are to create a new goal chunk when requested, to update the contents of any slot, and it can clear and save the goal's chunk into declarative memory (Bothell, 2017).

3.4 | Declarative memory

The declarative memory architecture in ACT-R is designed to mimic human memory. There are equations (available in the ACT-R manual) that note how quickly the memories are retrieved, which is based on how often they have been used as well as a threshold (that might vary by individuals or over time) for retrieval, and noise. The fan effect—how well connected the memory is to other memories, particularly currently accessed memories—also influences the time to retrieve a given memory (Anderson & Reder, 1999).

Typically, models will retrieve a memory using search terms, and then use the resulting memory in producing behavior. If a memory cannot be retrieved, the model will need to have strategies to work around this problem, or it will stop running. This approach also helps to explain why negative responses are slower than positive responses. There is also partial matching of rules to the current knowledge state (Lebiere, 1999), blending of memory retrievals (Gonzalez, Lerch, & Lebiere, 2003), forgetting (Anderson, Fincham, & Douglass, 1999), and instance-based learning (Gonzalez et al., 2003) mechanisms, where related memories can be matched or merged, respectively. They are not used in all models.

3.5 | Manual (motor)

The motor module receives command requests through the motor buffer. These commands direct the motor module how to output actions to the world. There is a small set of commands for typing on a keyboard and moving the mouse in the window provided with ACT-R, and some researchers have extended this approach with additional code to drive cars and robots. Again, the extensions are typically not included in the ACT-R distribution until they have been used and tested a few times.

The motor buffer can be overwritten by the actions of repeated rule firings, losing action commands. If too many commands are sent too quickly to the fingers, the commands can jam the system causing no behavior to occur. This loss of behavior during a jam is an implementation detail of the software rather than a specific claim regarding motor behavior (D. Bothell, personal communication, September 2018).

3.6 | Learning

There are several types of learning in ACT-R. Not all models use all of these learning mechanisms, and these mechanisms have varied somewhat across ACT-R versions.

Perhaps the mechanism with the largest learning effect is that of declarative memory strengthening. As memories are retrieved and used, they are strengthened; their activation is increased. Retrieval time for a memory is dependent on this activation; higher activation causes lower retrieval times. This approach has been used, for example, to choose strategies in games (West, Lebiere, & Bothell, 2005).

ACT-R also can learn procedural memories. In ACT-R 6 and 7, when two rules fire close enough together, the rules can be merged into a single rule. The times of the rule firings are often influenced by the retrieval of declarative memory. So, these two mechanisms of declarative and procedural knowledge interact in this way.

Notably, there are other areas where learning occurs in humans but does not yet appear in ACT-R. For the most part, ACT-R does not learn to recognize new visual icons, to improve its perceptual-motor capabilities, and to have memories that include synthesis (e.g., smell and sound) or emotions and semantics. These are areas for future work.

3.7 | Summary

Figure 3 provides a summary of the major components in ACT-R and how they interact. These major components are often found in other cognitive architectures (Laird et al., 2017). This figure can also be used to see components that are currently missing. For example, it does not have a limbic system and emotions. Further details on the implementation are available in the ACT-R tutorial and ACT-R manual. Further high-level descriptions are available in the canonical reports noted earlier.

4 | TOOLS FOR WORKING WITH ACT-R

As cognitive architectures have become more complex, tools for working with them have become more important. Tools also help to attract and support a wider range of users for models (Pew & Mavor, 1998; Ritter et al., 2003). There are several resource types that could be usefully provided for each architecture. Most of these tools for ACT-R are provided through the general repository (http://act-r.psy.cmu.edu). Box 5 notes how to get started with ACT-R.

4.1 | Programming interface and other implementations

The ACT-R distribution includes a graphical user interface (GUI) to help develop models. The primary implementation and realization of the ACT-R theory is in the Lisp version available from the ACT-R website (http://act-r.psy.cmu.edu).

The theory has been implemented in other base languages to help with integration with other systems. Their proponents argue that these help to understand what are the fundamental aspects of theory and what are implementation choices caused by the programming language. These implementations currently include two in Java (jACT-R, Harrison, 2008 and Java ACT-R, Salvucci, 2015).

4.2 | Instructional materials

As new people come into a research community, it is useful to have common instructional materials for them. The ACT-R community has several resources, including a manual, which continues to be updated as needed. Importantly, there is also a tutorial (act-r.psy.cmu.edu/software/) with eight units. Each unit notes an example model and has the learner to modify the model slightly to learn a modeling paradigm within the architecture (Taatgen, Lebiere, & Anderson, 2006). These tutorials are used in ACT-R summer schools, and in university classes and labs to learn ACT-R.

A list of frequently asked questions (FAQs, acs.ist.psu.edu/projects/act-r-faq/) about ACT-R with answers is also available. The FAQ is posted as a guide for finding out more about ACT-R. It is not comprehensive but can be helpful.

4.3 | Higher-level modeling languages based on ACT-R

There have been several high-level behavior representation languages that create ACT-R or ACT-R-like models (Ritter et al., 2006). These high-level languages let users create models in a higher, more declarative representation, such as the Problem Space Computational Model (PSCM) level (Newell, Yost, Laird, Rosenbloom, & Altmann, 1991) or GOMS (Card, Moran, & Newell, 1983). Herbal creates ACT-R models from PSCM constructs (Cohen, Ritter, & Haynes, 2010) and hierarchical task analyses (Paik, Kim, Ritter, & Reitter, 2015). G2A creates ACT-R models from GOMS descriptions (St. Amant, Freed, & Ritter, 2005). ACT-UP provides a more imperative programming approach for creating ACT-R models (Reitter & Lebiere, 2010).

With these tools, users can represent hierarchical or sequential tasks in an ACT-R model more easily. Users can lay out their whole task hierarchically or sequentially, and the relations among tasks are shown in a tree form. Based on these relationships, the ACT-R productions are created by a compiler.

These languages, including Herbal, have all advantages that other high-level languages used in computer science have. These advantages include: (a) smoother integration into systems created in those widely used languages, such as Java, supported by extensive libraries and tools; (b) a perceived and sometimes greater degree of implementation modularity, and thus the ability to more easily investigate changes and extensions to existing cognitive architectures; (c) the opportunity to make comparative analyses; and (d) ease of use through less data manipulation.

5 | ACT-R SUCCESS STORIES AND APPLICATIONS

It is useful to reflect upon ACT-R, and other architectures, by suggesting a Hall of Fame, that is, models within ACT-R that have provided notable insights, styles of models, or results. It is useful to examine places where ACT-R has been used

BOX 5

HOW TO GET STARTED WITH ACT-R*

How does one get started with ACT-R, or any cognitive architecture? There are several ways. One way that is most common is to use the architecture that is being used locally. This provides support and running example models within your environment. Another approach is to first read about several architectures and then choose one that seems to best support the type of work you are interested in. You can read books, journal articles, conference papers, and manuals to learn about the architectures and how they are used. If you are interested in memory and learning, ACT-R, for example, is a good choice. If you are interested in large system design and problem solving, Soar might be a better choice. If you are interested in neural implementations, then other architectures including Nengo (Eliasmith, 2013) can be good choices.

You might also find that none of the existing architectures suit your purposes, and then you might either create a new architecture or modify an existing one. We would suggest that modifying one is a better option because then you help results cumulate, you can build on existing theory and software, you can save time and effort, and you have more researchers immediately interested in your work who can comment.

When you have chosen your architecture, you can download it and start to use it. You should read the manual (and then can later come back to it once you know what it contains). You should also do some and perhaps all the tutorials that come with the architecture. You can be somewhat selective and choose the tutorials that teach the materials related to your modeling. You should get on a mailing list for the architecture, if there is one, and attend tutorials or meetings with others working with the architecture if you can. Finally, stay in touch with the community working with the architecture to share specific advice and also to learn about changes and uses of the architecture. There are further notes available on how to start using an architecture (Ritter, 2004).

There are several tools to help create ACT-R models. Some have been widely used, such as the GUI provided with ACT-R 6. Others have shown potential. The usability of these theories realized as cognitive architectures, despite their promise and the role of their scientific value, will also be judged by how many people can use them, and how easily the theories are to understand, manipulate, and apply.

successfully for at least two reasons. The first is that knowing where an architecture has been successful helps to define the architecture. The second is that it provides evidence that the architecture is worth taking seriously. Here, we briefly review some of the success stories. Some preference was given to models that are available on the ACT-R website. There are more models and other reviews could choose other success stories.

These models are noted because they provide insights about human behavior. They receive support for these insights typically because they match reaction times, but some predict error rates and strategy choices (Daily, Lovett, & Reder, 2001; Marewski & Mehlhorn, 2011). Some show that certain knowledge sets and mechanisms are sufficient to perform a task. Some provide applications that are useful.

5.1 | Models of memory

The most classic ACT-R models are of human memory (Anderson, Bothell, Lebiere, & Matessa, 1998; Anderson & Matessa, 1997; Anderson & Reder, 1999; Lebiere, 1999; Pavlik & Anderson, 2005; Stocco & Anderson, 2008). In particular, the Fan effect (e.g., Anderson & Reder, 1999) and the Stroop effect (Juvina & Taatgen, 2009) have been modeled. These models help to explain why memory can get stronger with practice, the effect on memory elements of being connected to other memory elements (fan effect), and how automatic processes can dominate less practiced processes (Stroop effect).

5.2 | Tutors

A cognitive tutor is a computer-based instructional technology that uses a cognitive architecture such as ACT-R. It can use the cognitive architecture for (a) understanding student's actions and guiding them; (b) solving the students' problems in a student-like behavior as an illustration (Ritter & Feurzeig, 1988); or (c) used as a basis for the design of the tutor and its strategies. ACT-R has been used to create tutors that use several of these approaches.

A modified version of ACT-R has been used to generate tutors for high school math curriculums (and other topics, typically formal domains with static problems, like programming), including proof generation in geometry and fundamental programming skills in Lisp (e.g., Anderson, Corbett, Koedinger, & Pelletier, 1995; Koedinger, Anderson, Hadley, & Mark, 1997). These tutors have a model in a modified version of ACT-R that can recognize student actions in the tutor as steps towards the solution (or not). Rather than generating behavior, they recognize behavior, particularly correct behavior. In studies, students achieve the same level of proficiency as conventional instructions in one-third of the time (Anderson et al., 1995) and perform better on standardized tests (Koedinger et al., 1997). ACT-R has also been used to design the ASSISTments tutoring system (Mendicino, Razzaq, & Heffernan, 2009) and been inspirational for the Declarative to Procedural tutoring system (Ritter et al., 2013).

5.3 | Models of driving

ACT-R has been used to create models of vehicle operators. Salvucci and colleagues have been creating a series of models of driving behavior and how driving interacts with other tasks (Salvucci, 2001, 2006, 2009; Salvucci & Taatgen, 2008, 2011; Salvucci, Zuber, Beregovaia, & Markley, 2005). Their models have been compared to and predict driver's input to the car and other interfaces, car performance, and eye movements. Similar but much simpler models have been created for cars (Ritter, Van Rooy, St. Amant, & Simpson, 2006) and robots (Ritter, Kukreja, & St. Amant, 2007; Trafton et al., 2013). ACT-R has also been used to create a model of airplane operation (Schoppek & Boehm-Davis, 2004; Schoppek, Holt, Diez, & Boehm-Davis, 2001) and air traffic control (Schoelles & Gray, 2000; Taatgen, 2002). These models use all the mechanisms in ACT-R to generate behavior, from recognizing objects and the road, to reasoning, to generating commands to drive.

We also include Byrne and Kirlik's (2005) model of airplane taxiing here. In their work, they used ACT-R models of how pilots choose strategies when taxiing planes around a simulation of the runway system of the O'Hare airport. They found suggestions for how to decrease errors when taxiing, a nice application of using a model to suggest system improvements.

5.4 | Models for testing interfaces and agents

There has been a long interest in creating models of users in ACT-R for use in interface design and as opponents and colleagues in shared interfaces (Anderson, Matessa, & Lebiere, 1997; Byrne & Gray, 2003; Gray & Altmann, 2001; Gray, Young, & Kirschenbaum, 1997). We note a few particular models here, but many of the models of specific regularities are also in service of modeling interface users, or could be, including models of perception and motor output (Ritter, Baxter, Jones, & Young, 2001).

There are several interesting examples. Paik et al. (2015) created a model of how novice interface users turn into relatively expert users of an interface over four trials based on their task time. This model used the learning mechanisms in declarative memory and rule compilation. Ball et al. (2010) created a synthetic teammate that interacts with natural language and a piloting task, and St. Amant, Horton, and Ritter (2007) have created models to evaluate cell phones. These models used all the mechanisms. Also, the previously noted driver and pilot models can be seen as these types of models.

These models have also led to insights about interaction. One is that if the information is not available, neither the model nor the user can do the task. Another is that milliseconds matter, that is, the difficulty of the moves and clicks can change how users process information (Gray & Boehm-Davis, 2000). Models have also explained why users continue to use suboptimal strategies (Fu & Gray, 2004).

5.5 | Models of how cognition changes

The basis of cognitive architectures is to study knowledge and the mechanisms to apply that knowledge, where the mechanisms do not change across tasks, people, or time. There are, however, aspects of how human behavior changes that appear not to be changes to knowledge but appear to be changes to the architecture or architecture parameters, influencing how information is acquired, stored, and processed. ACT-R has been used to represent the effect of sleep fatigue (Gunzelmann, Gross, Gluck, & Dinges, 2009), task appraisal stress (Dancy, Ritter, Berry, & Klein, 2015), task appraisal stress and caffeine (Kase, Ritter, Bennett, Klein, & Schoelles, 2017), and development on cognition (Jones, Ritter, & Wood, 2000).

5.6 | Models of language

ACT-R has been used to create models of language production and use. They contribute to the psycholinguistic question of how the mind selects words and processes sentences, but they also spell out how general cognition and language can be integrated. Examples include models of how language use co-evolves between users (Reitter, Keller, & Moore, 2011; Reitter & Lebiere, 2011; Trafton et al., 2013), particularly using declarative representations that change with learning, and how architectural constraints (e.g., time to process an utterance) shape language comprehension (Lewis, Vasishth, & Van Dyke, 2006).

5.7 | Other notable models

We include here other notable models that do not fit another category. The ACT-R distribution comes with a small set of models of psychology experiments. These include models of mental arithmetic, subitizing and the Sperling Iconic Task (a visual memory task), several memory tasks, and problem solving. The models are not designed to be a necessary or sufficient set of tasks. They are, however, a useful set and have been used to teach many modelers. Other architectures can either use this set of tasks as a set to model, or they can use the set as an exemplar and provide their own set of example tasks, models, and data. This set can be used to test other architectures, and we know of at least one ongoing effort to model these tasks and behavior in another architecture.

There are also significant models of multitasking (Moon & Anderson, 2013; Salvucci & Taatgen, 2008, 2011), decision making (Marewski & Mehlhorn, 2011), and spatial reasoning (Gunzelmann, Anderson, & Douglass, 2004), including how robots reason (Kennedy, Bugajska, Harrison, & Trafton, 2009). There are more models on the ACT-R website sortable by task, and there are more significant models than there is space to list.

5.8 | Summary

There are numerous ACT-R models available. The models that are available are a subset of all that are possible. That there are not models for all tasks simply suggests how broad is the range of behavior to model. The coverage also indicates the interests of the modelers using ACT-R. As researchers enter the community they will bring other interests and knowledge, and they will use these to create models in new areas. The sparse coverage could indicate problems with the architecture, but more likely indicates a lack of modelers and time to model.

Table 1 provides a summary of some of the significant models and extensions to ACT-R, a type of Hall of Fame. The ACT-R website notes many further models, searchable by author and category. These arose initially from ACT-R's core and can help to summarize further areas where ACT-R models have particular impact.

The work in Table 1 starts with models of memory and moves through applications to model core cognitive psychology tasks. The additional models show that ACT-R can be used in applied ways in psychology and in engineering. Some of the engineering applications then come back and show how core concepts in psychology can be explored in applied settings such as how novice to expert transitions can be modeled and how social networks can be explored with intelligent agents.

6 | CONCLUSIONS

This review summarizes and explains ACT-R, a fairly well developed and widely used cognitive architecture that has been used to model behavior in numerous areas. ACT-R's components provide a useful set of broad mechanisms to implement behavior. We conclude by noting insights ACT-R has provided for psychology and cognitive architectures as unified theories of cognition.

TABLE 1 Some significant models in ACT-R and their implications

Model	Insights/implications	Authors
Models of memory effects, (e.g., fan effect, spreading activation, priming)	Shows how memory effects affect numerous aspects of behavior and provides a foundation for modeling	Numerous
ACT-R/PM (perceptual-motor)	Shows how to incorporate a wide range of micro-theories about perception into a cognitive architecture as architectural components	Byrne and Anderson (1998)
Mapping of ACT-R to brain structures	Shows that the structures in ACT-R can be mapped to blood flow (activation) using fMRI and cognitive models that predict activation (blood flow) in the brain	Anderson (2007), Anderson et al. (2004)
Driving models	Shows how models can summarize behavior regularities in a complex real-time task	Salvucci (2006), Schoppek et al. (2001; 2004)
Taxiing at O'Hare Airport	Shows how to optimize large processes in large systems using ACT-R	Byrne and Kirlik (2005)
Herbal/Dismal	Models novice to expert transition on a 20-min spreadsheet task using a model compiler	Paik et al. (2015)
Synthetic teammate	Provides a comprehensive agent to live in a synthetic environment	Ball et al. (2010)

6.1 | Insights for psychology

ACT-R has provided several insights for psychology. One insight is how numerous aspects of cognition are intertwined. There are several interactions worth mentioning, such as how memory and information processing are entwined, and how problem solving and learning are entwined. In many models, the architecture also shows how learning supports memory and vice versa—with learning, memory becomes stronger, with stronger memory, more problem solving can be performed.

Joint work across architectures is starting to show how theories of cognition's mechanisms can be merged, like the perceptual-motor theories in EPIC were reused. The theories appear to be easier to merge on the level of the architecture rather than with rules and knowledge, but this could just be what has been able to be merged so far.

Numerous scientists in many fields have encouraged models that make strong predictions that integrate across a wide area (Grant, 1962; Marewski & Olsson, 2009; Newell & Simon, 1972), with a preference for those theories that can do the task (Feynman & Leighton, 1985; Newell, 1990). ACT-R shows that to predict behavior by generating it, mechanisms are required to produce the behavior. Thus, ACT-R models are larger than regression models, and they also produce many predictions including response items, what knowledge is used, error rates (sometimes), what part of the brain is used to generate the behavior, learning, and forgetting. This approach also provides a principled way to merge theories and an immediate way to apply the theories in simulations and games to test and apply the theories. These advantages are not obtained by small theories that simply describe the results as a single equation.

6.2 | Insights for cognitive architectures

ACT-R has provided insights about cognitive architectures. ACT-R suggests that cognitive architectures primarily need to be a theory of how behavior is generated through information processing including perception and motor output. ACT-R has shown that it can generate novel predictions and help explain human behavior. Generally, the architecture has been reused rather than the knowledge sets (models).

It also appears that having an interface and tools to help create models, a manual, example models, a summer school, a website, workshops, and some central technical support are also at least helpful (if not necessary) for architectures to have this level of impact. These resources are useful for building models and supporting a community of researchers who can be more powerful than a single team.

6.3 | What's next? The future of ACT-R

Table 2 gives several future directions we think that ACT-R should and can develop in. Most of these are self-explanatory or have been mentioned earlier in this study.

One direction that could use more explanation is nongenerative models. ACT-R has generally been used to create generative, information-processing models. That is, they perform the information processing required to do the task, and they interact with some piece of software to generate behavior.

A future for ACT-R is to make predictions more quickly and easily without doing the task as a nongenerative model, that does not do the task, but only makes predictions. ACT-R is intended to provide a theory of cognition, in the same way that Newtonian physics is intended to explain simple motion and provide direct predictions. In most instances, knowledge of Newton's equations allows physicists and even engineers to use these theories informally and without the use of equations.

While we are strong believers in formal theories and building models, in time, it may be possible for a wider range of folks to use ACT-R by only using the constructs and results of existing models. We are not sure how far this can or should go, but it appears to be a promising approach to extend our understanding for cognition and to extend the application of the theories in ACT-R.

TABLE 2 Future directions for ACT-R

Extending ACT-R to model further aspects of motor and sensory behavior
Integration with other task simulations
Learning a task from natural language
Integration of multiple models
Integration of physiology with cognition to support modeling the effects of physiology on cognition (e.g., heat stress) and as a principled way of combining moderators
Usability, so bigger models can be created and can be used by a broader range of users
The use of nongenerative models

ACKNOWLEDGMENTS

This report was supported by Office of Naval Research (N00014-15-1-2275) and by a project with Harris Corporation. Benjamin Bauchwitz, Marc Burns, Chung Il "Chad" Chae, Gary Cottrell, Ysabelle Coutu, William Kennedy, Julian Marewski, Ashley McDermott, and David Reitter gave comments that improved the paper. Alex Kirlik gave us the idea of a Hall of Fame for models; Dan Bothell gave numerous useful technical corrections; and Kenan Zecik gave suggestions on the figures. But any mistakes remain ours.

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

ENDNOTES

[1]ACT-R versions from v4 are available as software with manuals at http://act-r.psy.cmu.edu/old-act-r-software/

[2]http://chil.rice.edu/projects/RPM/index.html

RELATED WIREs ARTICLE

New and emerging models of human intelligence

REFERENCES

Allender, L., Archer, S., Kelley, T. D., & Lockett, J. (2005). Human performance modeling in the army: A long and winding road. In *Proceedings of Human Factors and Ergonomics Society* (Vol. 49, pp. 1191–1195). HFES: Santa Monica, CA.

Anderson, J. R. (1976). *Language, memory, and thought*. Hove, England: Psychology Press.

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Erlbaum.

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Erlbaum.

Anderson, J. R. (2005). Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science*, *29*(3), 313–341.

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

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036–1060.

Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, *38*(4), 341–380.

Anderson, J. R., & Bower, G. H. (1973). *Human associative memory*. Hillsdale, NJ: Erlbaum.

Anderson, J. R., Carter, C. S., Fincham, J. M., Qin, Y., Ravizza, S. M., & Rosenberg-Lee, M. (2008). Using fMRI to test models of complex cognition. *Cognitive Science*, *32*(8), 1323–1348.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, *4*(2), 167–207.

Anderson, J. R., & Douglass, S. (2001). Tower of Hanoi: Evidence for the cost of goal retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *27*(6), 1331–1346.

Anderson, J. R., Farrell, R., & Sauers, R. (1984). Learning to program in LISP. *Cognitive Science*, *8*, 87–129.

Anderson, J. R., Fincham, J. M., & Douglass, S. (1999). Practice and retention: A unifying analysis. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(5), 1120–1136.

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.

Anderson, J. R., & Lebiere, C. (2003). The Newell test for a theory of cognition. *Behavioral and Brain Sciences*, *26*(5), 587–601.

Anderson, J. R., & Matessa, M. (1997). A production system theory of serial memory. *Psychological Review*, *104*(4), 728–748.

Anderson, J. R., Matessa, M., & Lebiere, C. (1997). ACT-R: A theory of higher level cognition and its relation to visual attention. *Human Computer Interaction*, *12*(4), 439–462.

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., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, *2*, 396–408.

Ball, J., Myers, C., Heiberg, A., Cooke, N. J., Matessa, M., Freiman, M., & Rodgers, S. (2010). The synthetic teammate project. *Computational and Mathematical Organization Science*, *16*, 271–299.

Bell, C. G., & Newell, A. (1971). *Computer structures: Readings and examples*. New York, NY: McGraw-Hill.

Booher, H. R., & Minninger, J. (2003). Human systems integration in Army systems acquisition. In H. R. Booher (Ed.), *Handbook of human systems integration* (pp. 663–698). Hoboken, NJ: John Wiley.

Bothell, D. (2015). Recent updates to ACT-R. Presented at the ACT-R Summer School, CMU. Retrieved from http://act-r.psy.cmu.edu/wordpress/wp-content/themes/ACT-R/workshops/2015/bothell-actr2015-updates.pdf

Bothell, D. (2016a). ACT-R 6.1 is now ACT-R 7. Presented at the ACT-R Summer School, CMU. Retrieved from http://act-r.psy.cmu.edu/actr7/renaming-6.1-to-7.pdf

Bothell, D. (2016b). ACT-R 7 and recent updates. Presented at the ACT-R Summer School, CMU. Retrieved from http://act-r.psy.cmu.edu/wordpress/wp-content/uploads/2016/08/bothell-ACT-R-7-pgss.pdf

Bothell, D. (2017). ACT-R 7 reference manual. Retrieved from act-r.psy.cmu.edu/wordpress/wp-content/themes/ACT-R/actr7/reference-manual.pdf

Busetta, P., Rönnquist, R., Hodgson, A., & Lucas, A. (1999). JACK intelligent agents—Components for intelligent agents in JAVA. AgentLink News Letter, 2 (January). www.agent-software.com/white-paper.pdf

Byrne, M. D. (2012). Unified theories of cognition. *WIREs Cognitive Science*, *3*(4), 431–438.

Byrne, M. D., & Anderson, J. R. (1998). Perception and action. In J. R. Anderson & C. Lebiere (Eds.), *The atomic components of thought* (p. 82). Mahwah, NJ: Erlbaum.

Byrne, M. D., & Gray, W. D. (2003). Returning human factors to an engineering discipline: Expanding the science base through a new generation of quantitative methods. Preface to the special section. *Human Factors*, *45*(1), 1–4.

Byrne, M. D., & Kirlik, A. (2005). Using computational cognitive modeling to diagnose possible sources of aviation error. *International Journal of Aviation Psychology*, *15*(2), 135–155.

Card, S. K., Moran, T., & Newell, A. (1983). *The psychology of human–computer interaction*. Hillsdale, NJ: Erlbaum.

Chater, N., & Oaksford, M. (1999). Ten years of the rational analysis of cognition. *Trends in Cognitive Sciences*, *3*(2), 57–65.

Cohen, M. A., Ritter, F. E., & Haynes, S. R. (2010). Applying software engineering to agent development. *AI Magazine*, *31*(2), 25–44.

Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, *8*, 240–247.

Collins, A. M., & Quillian, M. R. (1972). Experiments on semantic memory and language comprehension. In *Cognition in learning and memory*. Oxford, England: John Wiley.

Daily, L. Z., Lovett, M. C., & Reder, L. M. (2001). Modeling individual differences in working memory performance: A source activation account. *Cognitive Science*, *25*(3), 315–355.

Dancy, C. L., Ritter, F. E., Berry, K., & Klein, L. C. (2015). Using a cognitive architecture with a physiological substrate to represent effects of psychological stress on cognition. *Computational & Mathematical Organization Theory*, *21*(1), 90–114.

Duch, W., Oentaryo, R. J., & Pasquier, M. (2008). Cognitive architectures: Where do we go from here? In *Proceedings of the first conference on Artificial General Intelligence* (pp. 122–136). AGI.

Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition*. New York, NY: Oxford.

Feigenbaum, E. A. (1961). *The simulation of verbal learning behavior.* Paper presented at the Western Joint IRE-AIEE-ACM Computer Conference, May 9–11 (pp. 121–132). ACM.

Feynman, R. P., & Leighton, R. (1985). *"Surely you're joking, Mr. Feynman!": Adventures of a curious character*. New York, NY: Norton.

Fincham, J. M., Carter, C. S., Van Veen, V., Stenger, V. A., & Anderson, J. R. (2002). Neural mechanisms of planning: A computational analysis using event-related fMRI. *Proceedings of the National Academy of Sciences*, *99*(5), 3346–3351.

Fleetwood, M. D., & Byrne, M. D. (2002). Modeling icon search in ACT-R/PM. *Cognitive Systems Research*, *3*(1), 25–33.

Friedman-Hill, E. (2003). *JESS in action*. Greenwich, CT: Manning Publications.

Fu, W.-T., & Gray, W. D. (2004). Resolving the paradox of the active user: Stable suboptimal performance in interactive tasks. *Cognitive Science*, *28*(6), 901–935.

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, *27*(4), 591–635.

Grant, D. A. (1962). Testing the null hypothesis and the strategy and tactics of investigating theoretical models. *Psychological Review*, *69*(1), 54–61.

Gray, W. D., & Altmann, E. M. (2001). Cognitive modeling and human–computer interaction. In W. Karwowski (Ed.), *International encyclopedia of ergonomics and human factors* (Vol. 1, pp. 387–391). New York, NY: Taylor & Francis, Ltd.

Gray, W. D., & Boehm-Davis, D. A. (2000). Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology*, *6*(4), 322–335.

Gray, W. D., Young, R. M., & Kirschenbaum, S. S. (1997). Introduction to this special issue on cognitive architectures and human–computer interaction. *Human Computer Interaction*, *12*, 301–309.

Grossberg, S. (1999). How does the cerebral cortex work? Learning, attention and grouping by the laminar circuits of visual cortex. *Spatial Vision*, *12*, 163–186.

Gunzelmann, G., Anderson, J. R., & Douglass, S. (2004). Orientation tasks with multiple views of space: Strategies and performance. *Spatial Cognition and Computation*, *4*(3), 207–253.

Gunzelmann, G., Gross, J. B., Gluck, K. A., & Dinges, D. F. (2009). Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling. *Cognitive Science*, *33*(5), 880–910.

Harrison, A. M. (2008). jACT-R: Making cognitive modeling portable. Retrieved from http://jact-r.org/

Hester, R. L., Brown, A. J., Husband, L., Iliescu, R., Pruett, D., Summers, R., & Coleman, T. G. (2011). HumMod: A modeling environment for the simulation of integrative human physiology. *Frontiers in Physiology*, *2*(12), 12.

Hull, C. L. (1952). *A behavior system: an introduction to behavior theory concerning the individual organism*. New Haven, CT: Yale University Press.

Hunt, E. (1971). What kind of computer is man? *Cognitive Psychology*, *2*(1), 57–98.

Jones, G., Ritter, F. E., & Wood, D. J. (2000). Using a cognitive architecture to examine what develops. *Psychological Science*, *11*(2), 93–100.

Juvina, I., & Taatgen, N. A. (2009). A repetition-suppression account of between-trial effects in a modified Stroop paradigm. *Acta Psychologica*, *131*(1), 72–84.

Kase, S. E., Ritter, F. E., Bennett, J. M., Klein, L. C., & Schoelles, M. (2017). Fitting a model to behavior reveals what changes cognitively when under stress and with caffeine. *Biologically Inspired Cognitive Architectures*, *22*, 1–9.

Kennedy, W. G., Bugajska, M. D., Harrison, A. M., & Trafton, J. G. (2009). "Like-me" simulation as an effective and cognitively plausible basis for social robotics. *International Journal of Social Robotics*, *1*(2), 181–194.

Kieras, D. E., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human–computer interaction. *Human Computer Interaction*, *12*(4), 391–438.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, *8*, 30–43.

Konik, T., O'Rorke, P., Shapiro, D., Choi, D., Nejati, N., & Langley, P. (2009). Skill transfer through goal-driven representation mapping. *Cognitive Systems Research*, *10*, 270–285.

Kotseruba, I., Gonzalez, O. J. A., & Tsotsos, J. K. (2016). A review of 40 years of cognitive architecture research: Focus on perception, attention, learning and applications. arXiv preprint arXiv: 1610.08602.

Laird, J. E. (1991). Preface for special section on integrated cognitive architectures. *SigART Bulletin*, *2*(4), 12–13.

Laird, J. E. (2012). *The Soar cognitive architecture*. Cambridge, MA: MIT Press.

Laird, J. E., Lebiere, C., & Rosenbloom, P. S. (2017). A Standard Model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *AI Magazine*, *38*(4), 13–26.

Laird, J. E., Rosenbloom, P. S., & Newell, A. (1986). Chunking in Soar: The anatomy of a general learning mechanism. *Machine Learning*, *1*(1), 11–46.

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, *10*, 141–160.

Lebiere, C., & Anderson, J. R. (1993). A connectionist implementation of the ACT-R production system. In *Proceedings of the fifteenth annual conference of the Cognitive Science Society* (pp. 635–640). Hillsdale, NJ: Erlbaum. Retrieved from http://act.psy.cmu.edu

Lebiere, C. (1999). The dynamics of cognition: An ACT-R model of cognitive arithmetic. *Kognitionswissenschaft*, *8*(1), 5–19.

Lebiere, C. (2002). The future of ACT-R. Presented at the ACT-R Summer School, CMU. Retrieved from http://act-r.psy.cmu.edu/wordpress/wp-content/themes/ACT-R/workshops/2002/talks/Lebiere-Future.pdf

Lewis, R. L., Vasishth, S., & Van Dyke, J. A. (2006). Computational principles of working memory in sentence comprehension. *Trends in Cognitive Sciences*, *10*(10), 447–454.

Marewski, J. N., & Mehlhorn, K. (2011). Using the ACT-R architecture to specify 39 quantitative process models of decision making. *Judgment and Decision making*, *6*(6), 439–519.

Marewski, J. N., & Olsson, H. (2009). Beyond the null ritual: Formal modeling of psychological processes. *Journal of Psychology*, *217*(1), 49–60.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: I. An account of basic findings. *Psychological Review*, *88*(5), 375–407.

Mendicino, M., Razzaq, L., & Heffernan, N. T. (2009). A comparison of traditional homework to computer-supported homework. *Journal of Research on Technology in Education*, *41*(3), 331–358.

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*(1), 3–65.

Moon, J., & Anderson, J. R. (2013). Timing in multitasking: Memory contamination and time pressure bias. *Cognitive Psychology*, *67*(1), 26–54.

Morrison, J. E. (2003). *A review of computer-based human behavior representations and their relation to military simulations (IDA Paper P-3845)*. Alexandria, VA: Institute for Defense Analyses.

Neves, D. M., & Anderson, J. R. (1981). Knowledge compilation: Mechanisms for the automatization of cognitive skills. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 57–84). Hillsdale, NJ: Erlbaum.

Newell, A. (1973). Production systems: Models of control structures. In W. G. Chase (Ed.), *Visual information processing* (pp. 463–526). Oxford, England: Elsevier.

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

Newell, A., Shaw, J. C., & Simon, H. A. (1960). Report on a general problem-solving program for a computer. In *International conference on information processing* (pp. 256–264). Paris: UNESCO.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Newell, A., Yost, G. R., Laird, J. E., Rosenbloom, P. S., & Altmann, E. (1991). Formulating the problem space computational model. In R. F. Rashid (Ed.), *Carnegie Mellon computer science: A 25-Year commemorative* (pp. 255–293). Reading, MA: ACM-Press (Addison-Wesley).

O'Reilly, R. C., & Munakata, Y. (2000). *Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain*. Cambridge, MA: MIT Press.

Paik, J., Kim, J. W., Ritter, F. E., & Reitter, D. (2015). Predicting user performance and learning in human–computer interaction with the Herbal compiler. *ACM Transactions on Computer-Human Interaction*, *22*(5), 25.

Pavlik, P. I., & Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive Science*, *29*(4), 559–586.

Pew, R. W. (2007). Some history of human performance modeling. In W. Gray (Ed.), *Integrated models of cognitive systems* (pp. 29–44). New York, NY: Oxford University Press.

Pew, R. W., & Mavor, A. S. (Eds.). (1998). *Modeling human and organizational behavior: Application to military simulations*. Washington, DC: National Academy Press.

Reitter, D., Keller, F., & Moore, J. D. (2011). A computational cognitive model of syntactic priming. *Cognitive Science*, *35*(4), 587–637.

Reitter, D., & Lebiere, C. (2010). Accountable modeling in ACT-UP, a scalable, rapid-prototyping ACT-R implementation. In D. D. Salvucci & G. Gunzelmann (Eds.), *Proceedings of the 10th international conference on Cognitive Modeling* (pp. 199–204). Philadelphia, PA: Drexel University.

Reitter, D., & Lebiere, C. (2011). How groups develop a specialized domain vocabulary: A cognitive multi-agent model. *Cognitive Systems Research*, *12*(2), 175–185.

Ritter, F. E. (2004). Choosing and getting started with a cognitive architecture to test and use human-machine interfaces. *MMI Interaktiv*, *7*, 17–37.

Ritter, F. E., Baxter, G. D., Jones, G., & Young, R. M. (2000). Supporting cognitive models as users. *ACM Transactions on Computer-Human Interaction*, *7*(2), 141–173.

Ritter, F. E., Baxter, G. D., Jones, G., & Young, R. M. (2001). User interface evaluation: How cognitive models can help. In J. Carroll (Ed.), *Human–computer interaction in the new millenium* (pp. 125–147). Reading, MA: Addison-Wesley.

Ritter, F. E., & Feurzeig, W. (1988). Teaching real-time tactical thinking. In J. Psotka, L. D. Massey, & S. A. Mutter (Eds.), *Intelligent tutoring systems: Lessons learned* (pp. 285–301). Hillsdale, NJ: Erlbaum.

Ritter, F. E., Haynes, S. R., Cohen, M. A., Howes, A., John, B., Best, B., … Vera, A. (2006). High-level behavior representation languages revisited. In *Proceedings of the seventh international conference on Cognitive Modeling* (pp. 404–407). Trieste, Italy: Edizioni Goliardiche.

Ritter, F. E., Kukreja, U., & St. Amant, R. (2007). Including a model of visual processing with a cognitive architecture to model a simple teleoperation task. *Journal of Cognitive Engineering and Decision Making*, *1*(2), 121–147.

Ritter, F. E., Shadbolt, N. R., Elliman, D., Young, R. M., Gobet, F., & Baxter, G. D. (2003). *Techniques for modeling human performance in synthetic environments: A supplementary review*. Wright-Patterson Air Force Base, OH: Human Systems Information Analysis Center.

Ritter, F. E., Van Rooy, D., St. Amant, R., & Simpson, K. (2006). Providing user models direct access to interfaces: An exploratory study of a simple interface with implications for HRI and HCI. *IEEE Transactions on System, Man, and Cybernetics, Part A*, *36*(3), 592–601.

Ritter, F. E., Yeh, K.-C., Cohen, M. A., Weyhrauch, P., Kim, J. W., & Hobbs, J. N. (2013). Declarative to procedural tutors: A family of cognitive architecture-based tutors. In *Proceedings of the 22nd conference on behavior representation in modeling and simulation* (pp. 108–113). Centerville, OH: BRIMS Society.

Salvucci, D. D. (2001). Predicting the effects of in-car interface use on driver performance: An integrated model approach. *International Journal of Human-Computer Studies*, *55*(1), 85–107.

Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors*, *48*(3), 362–380.

Salvucci, D. D. (2009). Rapid prototyping and evaluation of in-vehicle interfaces. *ACM Transactions on Computer-Human Interaction*, *16*(2), Article 9, 33 pp.

Salvucci, D. D. (2015). Java ACT-R. A Java simulation and development environment for the ACT-R cognitive architecture. Retrieved from http://cog.cs.drexel.edu/act-r/

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review*, *115*, 101–130.

Salvucci, D. D., & Taatgen, N. A. (2011). *The multitasking mind*. New York, NY: Oxford.

Salvucci, D. D., Zuber, M., Beregovaia, E., & Markley, D. (2005). Distract-R: Rapid prototyping and evaluation of in-vehicle interfaces. In *CHI 2005 conference proceedings on human factors in computing systems* (pp. 581–589). New York, NY: ACM Press.

Samsonovich, A. V. (2010). Toward a unified catalog of implemented cognitive architectures. In *Biologically inspired cognitive architectures* (pp. 195–244). Amsterdam: IOS Press.

Sauers, R., & Farrell, R. (1982). *GRAPES user's manual*. Pittsburgh, PA: Carnegie-Mellon University.

Schoelles, M. J., & Gray, W. D. (2000). Argus Prime: Modeling emergent microstrategies in a complex simulated task environment. In *Proceedings of the 3rd international conference on cognitive modelling* (pp. 260–270). Veenendaal, the Netherlands: Universal Press.

Schoppek, W., & Boehm-Davis, D. A. (2004). Opportunities and challenges of modeling user behavior in complex real world tasks. *MMI Interaktiv*, *7*, 47–60.

Schoppek, W., Holt, R. W., Diez, M. S., & Boehm-Davis, D. A. (2001). Modeling behavior in complex and dynamic situations—The example of flying an automated aircraft. In *Proceedings of the fourth international conference on cognitive modeling* (pp. 265–266). Mahwah, NJ: Erlbaum.

Selfridge, O. G. (1959). Pandemonium: A paradigm for learning. In *Proceedings of the symposium on mechanisation of thought processes*, 511–529. London: Her Majesty's Stationery Office.
St. Amant, R., Freed, A. R., & Ritter, F. E. (2005). Specifying ACT-R models of user interaction with a GOMS language. *Cognitive Systems Research*, *6*(1), 71–88.
St. Amant, R., Horton, T. E., & Ritter, F. E. (2007). Model-based evaluation of expert cell phone menu interaction. *ACM Transactions on Computer-Human Interaction*, *14*(1), 1–24.
St. Amant, R., Riedel, M. O., Ritter, F. E., & Reifers, A. (2005). Image processing in cognitive models with SegMan. In *Proceedings of HCI International '05*. Mahwah, NJ: Erlbaum.
Stocco, A., & Anderson, J. R. (2008). Endogenous control and task representation: An fMRI study in algebraic problem-solving. *Journal of Cognitive Neuroscience*, *20*(7), 1300–1314.
Stocco, A., Lebiere, C., & Anderson, J. R. (2010). Conditional routing of information to the cortex: A model of the basal ganglia's role in cognitive coordination. *Psychological Review*, *117*(2), 541–574.
Sun, R., & Bookman, L. A. (Eds.). (1994). *Computational architectures integrating neural and symbolic processes: A perspective on the state of the art*. Norwell, MA: Kluwer.
Taatgen, N. A. (2002). A model of individual differences in skill acquisition in the Kanfer–Ackerman air traffic control task. *Cognitive Systems Research*, *3*(1), 103–112.
Taatgen, N. A., Lebiere, C., & Anderson, J. R. (2006). Modeling paradigms in ACT-R. In R. Sun (Ed.), *Cognition and multi-agent interaction: From cognitive modeling to social simulation* (pp. 29–52). New York, NY: Cambridge.
Taatgen, N. A., Rijn, H. V., & Anderson, J. R. (2004). Time perception: Beyond simple interval estimation. In *Proceedings of the sixth international conference on cognitive modeling* (pp. 296–301). Mahwah, NJ: Erlbaum.
Tehranchi, F., & Ritter, F. E. (2018, 2018). Modeling visual search in interactive graphic interfaces: Adding visual pattern matching algorithms to ACT-R. In *Proceedings of 16th international conference on cognitive modeling* (pp. 162–167). Madison, WI: University of Wisconsin.
Trafton, G., Hiatt, L., Harrison, A. M., Tamborello, F., Khemlani, S., & Schultz, A. (2013). ACT-R/E: An embodied cognitive architecture for human–robot interaction. *Journal of Human-Robot Interaction*, *2*(1), 30–55.
West, R. L., Lebiere, C., & Bothell, D. J. (2005). Cognitive architectures, game playing, and human evolution. In R. Sun (Ed.), *Cognition and multi-agent interaction: From cognitive modeling to social simulation* (pp. 103–123). New York, NY: Cambridge.

How to cite this article: Ritter FE, Tehranchi F, Oury JD. ACT-R: A cognitive architecture for modeling cognition. *WIREs Cogn Sci*. 2018;e1488. https://doi.org/10.1002/wcs.1488