Towards Cognitive Architectures with Multiple Levels of Abstraction

Niels Taatgen (n.a.taatgen@rug.nl)
University of Groningen, Institute of Artificial Intelligence, Nijenborgh 9
9747 AG Groningen, Netherlands

Abstract

Cognitive architectures aim to bridge the gap between the brain and behavior, providing a formal level of description that provides a basis for rigorous theories of behavior. Architectures typically operate on a single level of abstraction. In this paper I argue that we need multiple levels of abstraction, each with their own formalisms and learning mechanisms. Each of these levels should be to explain the abstraction level above, creating a reductionist hierarchy of theories that can bridge the gap, not with a single formalism, but with several.

Keywords: cognitive architectures, reductionism, neural networks, PRIMs, ACT-R, compositionality

Introduction

The goal of cognitive architectures is to provide a platform to model all aspects of human intelligence. Although they have been very successful in modeling wide ranges of phenomena, no current architecture is close to delivering an artificial general intelligence. A possible reason is that more work needs to be done on expanding the capabilities of existing architectures. The downside to that approach is that by expanding the capabilities of architectures, the ability to constrain possible models diminishes, in other words, they become weaker theories (Newell, 1990). This article explores a different approach: maybe an architecture at a particular level of abstraction is not enough to cover the gap between 100 billions of cells in the brain and human-level intelligence, and therefore we need a multiple levels of abstraction.

Take a typical cognitive architecture, for example ACT-R (Anderson, 2007). The bulk of the research in ACT-R involves modeling human data from behavioral experiments. Models in ACT-R therefore model phenomena that concern a single constrained task covering a modest time-span, say one or two hours, requiring as little background knowledge as possible. ACT-R assumes that our brains are capable of representing symbolic knowledge in the form of declarative chunks and production rules, without being overly concerned about underlying neural structures (but also not unconcerned, see Stocco, Lebiere, & Anderson, 2010, and many fMRI studies conducted with ACT-R). On the more abstract side, ACT-R also does not cover the more long-term aspects of knowledge and learning, and how knowledge is interconnected. More critically, ACT-R does not provide representations that are able to support compositionality: the human ability to quickly understand and carry out novel tasks as long as the components of that task are already mastered. Take for example the two example mini-tasks in Figure 1. Although both will be completely novel to most readers, we can all perform them out effortlessly, because we can easily combine concepts such as counting, selecting items with a particular attribute, and determining whether something is more than something else.

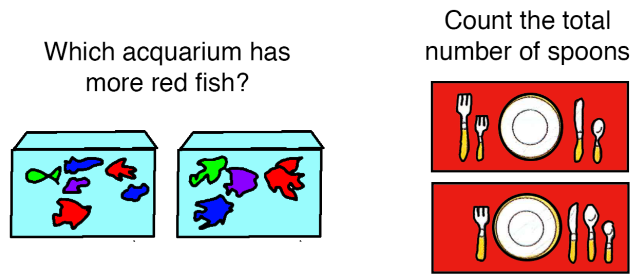

Figure 1. Two novel mini-tasks that require no training

This is not a critique of ACT-R, because it performs excellently at its chosen level of abstraction. However, if we want to move forward we have to consider moving to architectures with multiple levels of abstraction that each have their own representation and learning mechanisms. Critical to the idea is that the different levels are strongly linked together, by showing that a higher level of abstraction is produced by the lower level by composing lower-level knowledge elements of that lower level.

There are, of course, many ways in which such a multilevel architecture can be specified. In this article I want to kick off the discussion with a first proposal that builds on several existing architectures and modeling approaches. Given that I use ideas from many different sources, I do not want to "brand" the idea, but to save words I will refer to it with "the MLA" (multi-level architecture).

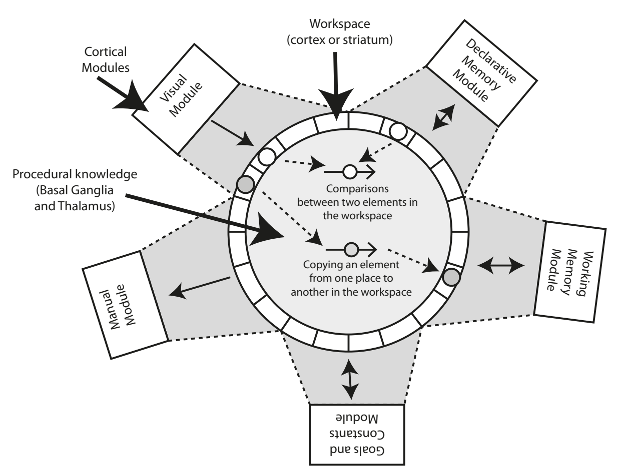

Figure 2. The global workspace model (PRIMs variation)

The Multi-Level Architecture

Global Workspace Assumption

The starting assumption of my proposal for an MLA is that the human brain is to some extent modular, and therefore can be subdivided into several modular areas, such as perception and motor areas, but also several types of memory, areas related to control, and possibly (but not necessarily) specialized cognitive functions such as language and numerical cognition. A consequence of this modular organization is that it is necessary to specify how modules make information available to other modules, and to have mechanisms that control the flow of that information. A starting point is the global neuronal workspace (Baars, 1997; Dehaene, Kerszberg, & Changeux, 1998). The workspace is a common area in which the different modules can exchange information. But how do modules "know" when to use information from one another? For this we need (procedural) knowledge that decides, based on the current contents of the workspace, what information has to be moved where within the workspace. Figure 2 shows the general concept of the workspace, where different modules have their own subspace of the workspace, and where procedural knowledge (in the middle) moves around information. The subspace for each module consists of a number of "slots" that can be used to represent separate information items. The idea of a global workspace has appeared in different other incarnations in cognitive science and artificial intelligence, for example in blackboard architectures (Hayes-Roth, 1985), and in buffers in ACT-R (Anderson, 2007).

Examples

Given the general assumptions of the global workspace, and taking this as the central organizing element in the MLA, we can now define the different layers of abstraction. I will use two examples to illustrate how the different levels represent, use and learn knowledge.

The first example is an aural-vocal task. The model hears a tone of low, medium or high pitch, and has to respond by saying "one", "two", or "three", respectively. I use this example because it requires few procedural steps, and has been successfully modeled in a neural architecture (Stocco et al., 2010). The second example consists of the two mini-tasks in Figure 1.

The Layered Architecture

Figure 3 gives an overview of the proposed MLA. At the bottom it starts with clusters of neurons, and ends with representations of tasks. In between are three layers.

The general idea is that the units at a particular level serve as the building blocks for a level higher. Therefore learning mechanisms mainly operate between levels as far as they build new knowledge representations. The Figure also

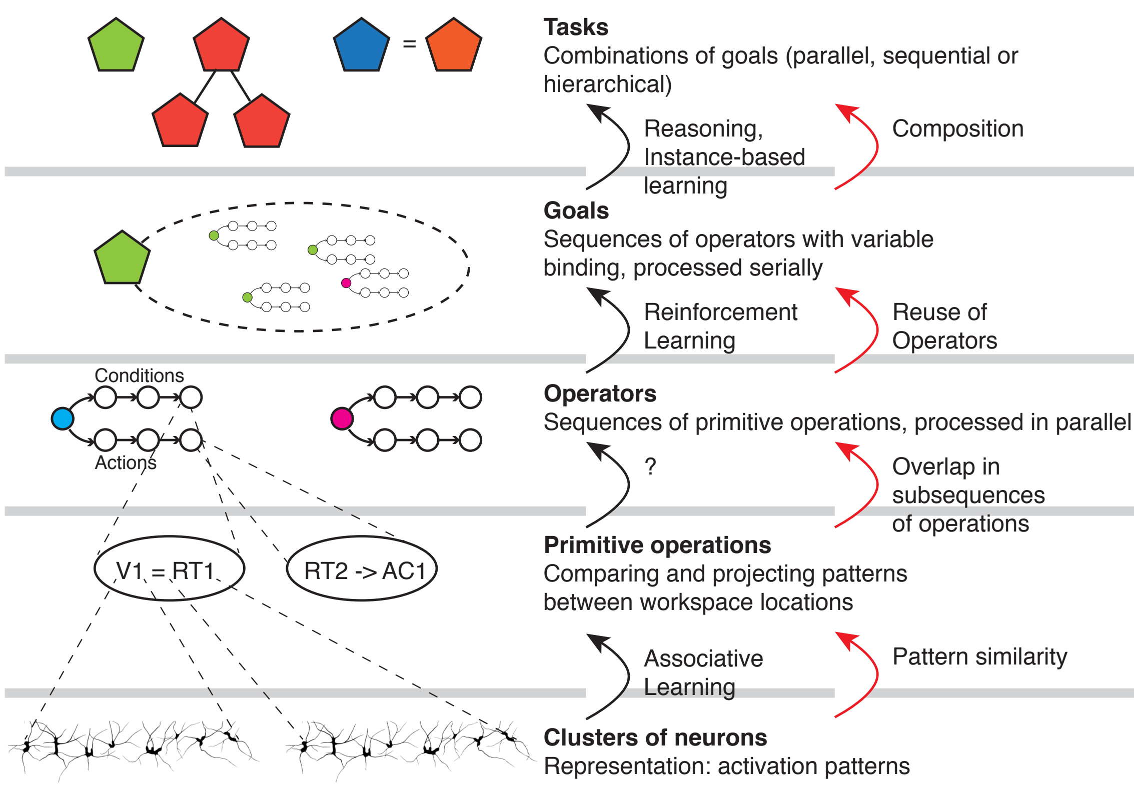

Figure 3. Proposal for an MLA with five levels of abstraction. Each level provides the building blocks for one level higher, and therefore learning mechanisms operate between levels. Also indicated (red arrows) is how knowledge can be reused to produce transfer of learning.

specifies how knowledge can be used for different purposes, providing knowledge transfer between tasks.

The Level of Clusters of Neurons

A generally held assumption in neural networks is that meaning is represented by activation patterns in clusters of neurons (e.g., Eliasmith et al., 2012; Stocco et al., 2010). At this level, each slot in the Global Workspace consists of a cluster of neurons, and to different firing patterns different meanings can be attached. For example, in the aural-vocal task aural processing of the tone can produce a particular firing pattern in a slot of the aural module. But how does this pattern become meaningful? For that it has to be able to activate certain memories that are linked to that tone. For this, the tone has to be used as a cue for memory recall. But the neural representation of, say, a low-pitched in memory may differ from that produced by the aural module. Therefore the aural pattern of a low-pitched tone has to be transformed into a memory pattern of a low-pitched tone, which is something the network has to learn through some form of associative learning (or by using an algorithm, Eliasmith et al., 2012).

The model by Stocco et al. (2010) assumes three modules: a vocal module, a memory module (Prefrontal), and an aural module. Coordination between these modules is performed through the Basal Ganglia. Figure 4 illustrates how the model performs the aural vocal task. The gray boxes represent three subareas in the Global workspace, each with 100 simulated neurons. In the top box (Stage A), the aural module has perceived the tone, and produces an activation pattern that represents that tone. This representation has to be transformed into a memory query to determine what word corresponds to the pitch of the tone. The basal ganglia therefore maps the input from the aural input onto the prefrontal area that is linked to memory, producing the activation pattern in Stage B. The prefrontal activation pattern corresponds to a memory query that asks: "What vocal output does this tone pitch correspond to". In Stage C, memory has produced the relevant memory trace, and produces a new pattern that represents the word that has to be spoken. It is now the Basal Ganglia's turn again to transform that memory pattern into an actual speech representation that can be put into the Vocal part of the Global Workspace (Stage D).

The Level of Primitive Operations

On the next level of the MLA we are going to abstract away neural firings. They are replaced by symbols that represent particular neural firing patterns. Instead of the particular pattern of neurons that represents a low pitch, we will just specify the LOW-PITCH is in a slot of the aural part of the workspace.

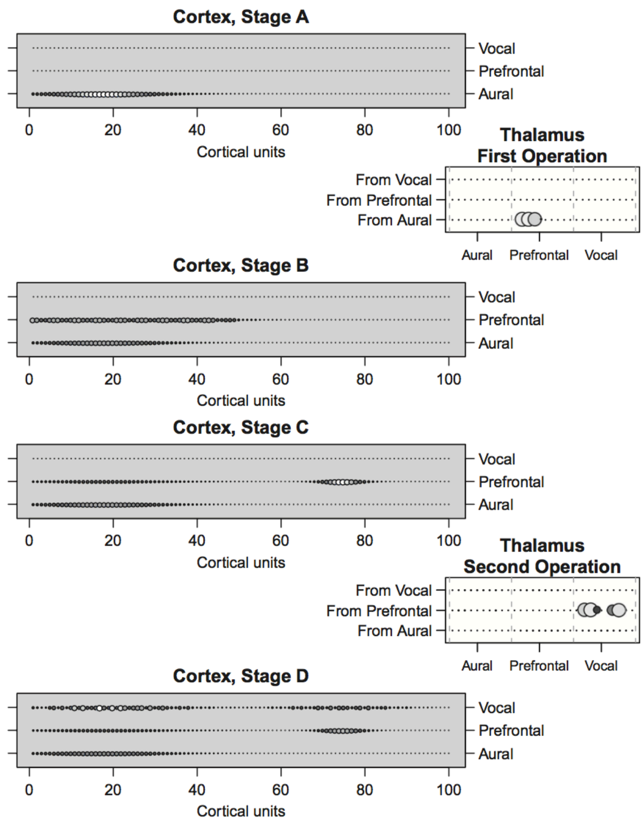

Figure 4. Activation in Stocco et al.'s model of the Aural-Vocal task. Reprinted with permission from Stocco et al. (2010). Copyright 2010 American Psychological Association.

By doing so, we are discarding a lot of information, for example the ability to determine that a low pitch is more similar to a medium pitch than a high pitch. We can salvage that at the higher levels, but we need to explicitly define similarities that we get for free at the lower levels.

In addition, we use the same representation for the low pitch in the aural part of the workspace as in the memory part, even though they are different patterns in the neural model that have to be learned through associative learning.

The activity in the Basal Ganglia can now be simplified into primitive operations (Taatgen, 2013). First, given an input the first slot of the aural workspace, copy that representation to the first slot of the memory system. Second, given a representation in the second slot of the memory system, copy that representation to first slot in the vocal system:

```
AU1 -> MEM1
MEM2 -> VOC1
```

All we need to additionally specify is that memory contains three items: (LOW-PITCH, ONE), (MEDIUM-PITCH, TWO), and (HIGH-PITCH, THREE).

For a very primitive single-task model this may be enough. But actions are always tied to a particular context, possibly augmented by particular conditions. We carry out an aural-vocal task based on skills we have (to respond to an aural stimulus with a pre-memorized response), supplied with the particulars (tones with certain pitches, certain words), and by setting the goal to actually perform this task. To do all this we need more levels of organization, but also an additional primitive operation, namely the ability to compare elements in the global workspace. For example, in this task we only want to respond to tones, and not to other sounds. Suppose that the aural system places the type of sounds in AU1, and, if it is a tone, the pitch of tone in AU2. In that case we want to make a comparison first, using a comparison primitive operator:

```
AU1 = tone
```

The problem with this comparison is that it is not a primitive operation, because it contains a particular value ("tone"). We don't want particular values at this level of abstraction, because we do not want to define primitive operators for every possible value. The solution is to create an additional subspace in the workspace ("C") to represent currently relevant particular values. If we do that, the comparison becomes:

```
AU1 = C1
```

The nice property of the primitive operations is that, given a particular size of the workspace, their number is fixed. This fixed-sized set are the building blocks for the next level, operators.

The Level of Operators

Operators are perhaps the most common form of representation in cognitive architectures. Soar also has operators (although they work differently), but in ACT-R and most other architectures they correspond to production rules. Operators combine several primitive operations, all of which are typically carried out in parallel (or in quick automated succession). Operators first carry out a number of primitive operations that test conditions, and if these are satisfied carry out operations that perform actions. For the Aural-Vocal task we need two operators that assume that C1 contains "Tone", C2 contains "Associate", and C3 contains "Say":

```
Operator 1:          Operator 2:
AU1 = C1             MEM1 = C2
C2 -> MEM1           C3 -> VOC1
AU2 -> MEM2          MEM2 -> VOC2
```

Although operators are like production rules, they have been kept neutral about what goal or task they are for. We therefore represent all the particulars in the "C" part of the workspace. Moreover, the operators do not explicitly reference a goal, which makes it possible to reuse operators for different goals. Another property of operators, contrary to productions, is that they do not bind variables, but assume relevant variables are supplied in the "C" subspace. The next level of abstraction, the goal, is responsibility for supplying values to that space.

The Level of Goals

To achieve all but the most elementary goals several thinking steps have to be taken, in other words, we need several operators that are carried out in sequence. Given that operators only move around information in the Global Workspace, operator activity is interleaved with module activity (we already saw this in the neural example in Figure 4).

Goals organize which operators belong to that goal, but are also responsible for instantiating particular values (i.e., binding is carried out at the level of goals). In the Aural-Vocal example, we specify that the goal is associated with two operators, and that it can be instantiated with an aural type ("Tone" in the example), an index to the type of associate fact that needs to be retrieved ("Associate"), and the action that needs to be performed ("Say"). However, we can also change these bindings to modify the goal's behavior, providing a means to use the same goal for different purposes.

For the elementary Aural-Vocal task, one goal is enough. However, for even the simple tasks from Figure 1, multiple goals have to be composed in order to explain why we can carry out these tasks without any prior learning.

The Level of Tasks

At the level of tasks we instantiate multiple goals and link them together in an appropriate structure. The idea leans on compositionality in language, where words and grammar allow us to produce and understand sentences we have never heard before. The task level offers the same versatility: to use goals and instantiations of goals to produce tasks we have never done before. Goals within a task can be organized in several ways: they can follow each other sequentially, they can be active in parallel, or may be organized hierarchically. For the Figure 1 examples, we need a hierarchical organization. For the "Which aquarium has more red fish?" task, we need a *compare* goal, a *count* goal, and a *has-property* goal. These goals are organized hierarchically: compare is the top goal, and has count as a subgoal, which in turn has had-property as subgoal, each instantiated with the appropriate bindings.

As an illustration for the compositionality property, the other example in Figure 1, "Count the number of spoons", can be carried out with some of the same goals: the top goal is now an *add-all* goal, but the subgoals are the same as in the aquarium task, but with different bindings. Figure 5 shows the two tasks across several levels of abstractions. Not only do the two tasks share two out of three goals, the two goals that they do not share still have overlap in operators, and also overlap in the sequences of primitive operations.

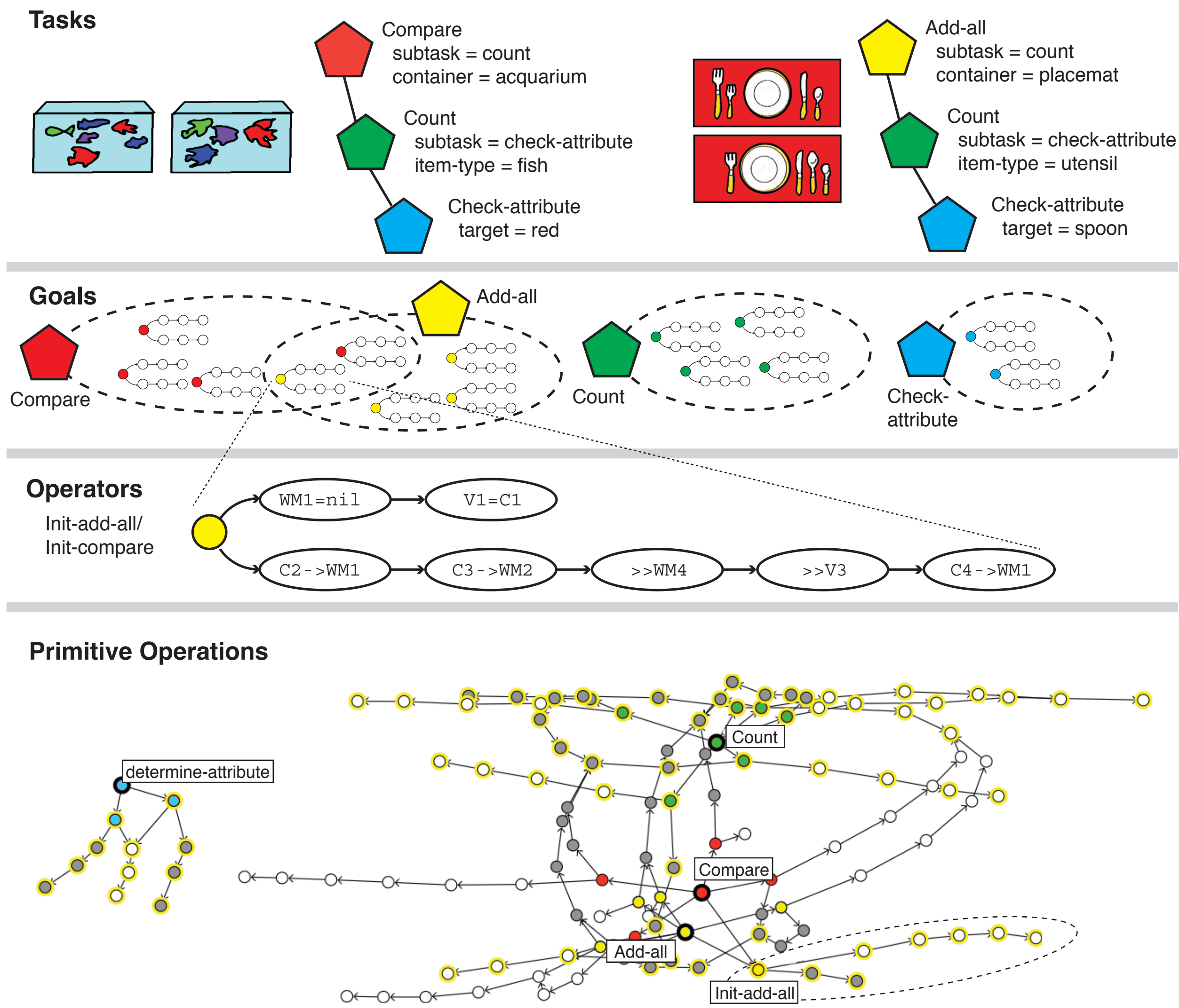

Figure 5. Illustration of the representation of the two tasks from Figure 1 across levels of abstraction. At the task level each task has a hierarchy of three goals with particular bindings. At the goal level, each goal is associated with a set of operators, which can potentially have overlap. At the operator level, several primitive operations are bound together, but particular substrings of primitive operations can be reused (note that, for brevity, we have not explained the >> primitive operation). The bottom section, "Primitive operations", is output from the PRIMs application in which both tasks are modeled. It shows all the operators with their strings of primitive operations, and where these overlap. For example, the determine-attribute goal (fat blue circle) has two operators (two thin blue circles), each of which has three condition primitive operations (gray circles), and three action primitive operators (white circles). The action primitive operations are the same for both operators, and are therefore reused.

Learning and Transfer

An interesting property of the MLA is that at the bottom levels the system is quite mechanical and syntactic in nature, lacking "meaning" of what it is doing, while the highest levels it is close to semantics in natural language. The learning mechanisms linked to the different levels should reflect this.

So why don't we just model the highest levels of abstraction, and consider the lower levels as implementation details? It is because learning and transfer happens at all levels of abstraction, and therefore leaving out levels results in an incomplete system.

Learning

In machine learning, typically three types of learning are distinguished: unsupervised learning, reinforcement learning (i.e., supervision by reward only), and supervised learning (correct answer is given as feedback). I would like to add a fourth level to this: learning by instruction and explicit reasoning. Learning has to be concerned with two aspects: how to assemble elements from one level to create a unit one level higher in the MLA, and how to evaluate whether existing elements of knowledge have to retained or discarded.

At the lowest level of the MLA unsupervised learning is the most likely candidate for learning, because processing at that level is too fine-grained to know correct from wrong. Instead, co-occurring patterns are associated with each other, creating the links that the next higher level (primitive operations) need to function. If we move up in levels, reinforcement learning combined with genetic algorithms is a possible candidate to construct operators out of primitive operations. At the highest level of abstraction, tasks can be assembled from instructions (by parsing natural language into task representations), from examples (by deriving what operators are needed to achieve a certain thinking step), past experiences, or by reflection (by mentally simulating what the outcome of a certain goal will be). For high-level learning there is much to learn from Soar (Laird, Rosenbloom, & Newell, 1986), a cognitive architecture that has been focused on higher-level reasoning processes.

Transfer

One of the conclusions we can draw from the concept of an MLA is that tasks are not learned in isolation, but that knowledge is interconnected at all levels of abstraction. At every level this offers opportunities for knowledge transfer.

At the neuronal level, similarity between activation patterns can signal that these patterns can be processed in a similar or even identical way. This similarity originates in perceptual and motor systems that are connected to the real world, but can result of associative learning processes that preserve some of the pattern similarities in higher-level representations.

Primitive operations often end up in the same clusters. Through compilation processes, these clusters can be executed more efficiently, and can therefore be preferred in the case of choice. Taatgen (2013) has shown that this type of low-level transfer can be quite powerful, and capable of modeling transfer among different text editors, as well as different tasks that require cognitive control. Similarly, goals can reuse operators that have been learned for a different purpose, sidestepping a long bottom-up learning process.

The largest potential for transfer is in the reuse of goals through a composition process that instantiates and connects goals for new tasks that the system has never done before, but that is nevertheless composable from the skills the system already has.

Conclusion

Multi-level architectures have the potential to unify many approaches to cognitive modeling, whether neural, symbolic, hybrid or otherwise. By creating a strong reductionist framework the pitfall of exploding complexity can be avoided.

This paper only outlines the structure of a possible MLA: a lot more work needs to be done to make it a reality.

References

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

Baars, B. J. (1997). In the theatre of consciousness. Global Workspace Theory, a rigorous scientific theory of consciousness. *Journal of Consciousness Studies*, *4*(4), 292–309.

Dehaene, S., Kerszberg, M., & Changeux, J. P. (1998). A neuronal model of a global workspace in effortful cognitive tasks. *Proceedings of the National Academy of Sciences*, *95*, 14529–14534.

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., … Rasmussen, D. (2012). A large-scale model of the functioning brain. *Science*, *338*(6111), 1202–5. http://doi.org/10.1126/science.1225266

Hayes-Roth, B. (1985). A blackboard architecture for control. *Artificial Intelligence*, *26*(3), 251–321. http://doi.org/https://doi.org/10.1016/0004-3702(85)90063-3

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

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

Stocco, A., Lebiere, C., & Anderson, J. R. (2010). Conditional routing of information to the cortex: A model of the basal ganglia{\textquoteright}s role in cognitive coordination. *Psychological Review*, *117*(2), 541–574.

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