6

The Representation of Task Knowledge at Multiple Levels of Abstraction

Niels A. Taatgen

Abstract

Interactive task learning requires knowledge and skills that are highly flexible and composable, and a cognitive architecture to support this. Cognitive architectures aim to bridge the gap between the brain and intelligence, providing a formal level of description for rigorous theories of behavior. Architectures typically operate on a single level of abstraction, but this may be too limited for interactive task learning. Instead, architectures with multiple levels of abstraction should be considered, each with their own formalisms and learning mechanisms. Each level should be able to explain the abstraction level above it, thus creating a reductionist hierarchy of theories to model human intelligence, not with a single formalism, but with several.

Introduction

Suppose a teacher were to give the following instruction: "Using this box of paperclips, create an outline of a rabbit." This instruction would not, in general, pose an immense challenge to the learner, even though the task itself may be novel, because the concepts involved are understood by the learner.

Now consider a second example of instruction: "Please use these paperclips to woffle a dingdong." Here, the directive would be far more problematic, because the concepts "to woffle" and "dingdong" are unknown. The instruction would need, therefore, to include a verbal definition of the concepts (e.g., "to woffle" means to create an outline). Words alone, however, may not suffice: the learner may require a demonstration. In addition, if an unknown concept (e.g., woffling) entails complex actions, several demonstrations and learning attempts may be required before the learner is able to replicate it successfully.

From "Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions," edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26, J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.

These two examples demonstrate different types of learning: In the first, the learner is able to draw on existing skills. In the second, a particular sequence and combination of mental operations is required, but if these mental operations cannot be expressed in words, other instructional methods may also be needed.

Interactive task learning (ITL) can also require a lower level of abstraction, as in the following: "Please use the paperclips to 信息 the 面写." Assuming that the learner has no experience with Chinese characters, s/he would first need to learn the meaning of the words involved as well as be able to recognize the particular characters. In ITL, the goal of the teacher is to supply the learner with the knowledge needed to perform the task. As illustrated in this example, the knowledge that the learner lacks may stem from different levels and require a different type of teaching: a verbal explanation consisting of a few sentences, an extensive demonstration, or a large set of training materials.

In this chapter, I investigate the hierarchy of abstraction of task knowledge through the lens of cognitive architectures. The goal of cognitive architectures is to provide a platform to model all aspects of human intelligence, including human ITL. Although architectures have been very successful in modeling wide ranges of phenomena, no current architecture is close to having the capability of learning arbitrary new tasks from instructions. A possible reason is that most architectures limit themselves to one or a few levels of abstraction. As a consequence, many architectures are very good in explaining certain phenomena, but may struggle to explain others (Newell 1990).

Take a prototypical cognitive architecture: ACT-R (Anderson 2007). The majority of research into ACT-R involves modeling human data from behavioral experiments. Models created with ACT-R, therefore, usually address phenomena that concern a single constrained task that can be completed within a minute, is repeated (possibly with variations) over a modest span of time (e.g., 1–2 hours), and requires 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 (but not indifferent) to how our brain implements these representations (see, e.g., Stocco et al. 2010). More abstractly, ACT-R is less suitable in covering the long-term aspects of knowledge and learning and how knowledge of different tasks is interconnected. ACT-R does not, however, provide representations that directly support compositionality; that is, the human ability to understand and carry out novel tasks quickly as long as the components of that task have already been mastered.

Consider the two tasks in Figure 6.1: although both may be completely novel to most readers, each of us should be able to perform them effortlessly, because we can easily combine skills such as counting, selecting items with a particular attribute, and determining whether something is more than something else. Moreover, after completing these tasks, we have not acquired a new "count the spoons on placemats" or "more red fish" skill. In terms of ITL, as adults, all we needed was the instruction written above the picture. Young

From "Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions," edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26, J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.

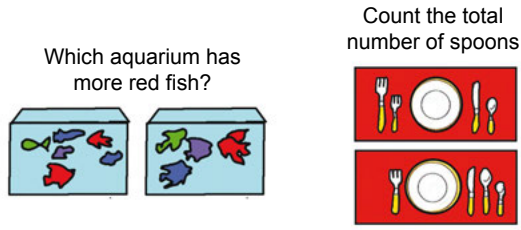

Figure 6.1 Two novel tasks that require no training for a typical adult.

children, by contrast, could have difficulty solving these tasks, if they have not mastered certain skills (e.g., counting, reading). Thus, for children to solve these tasks, additional instruction of a different type (e.g., demonstration) may be required.

This is not a critique of ACT-R: it performs excellently at its chosen level of abstraction. However, if we want to make progress on architectural approaches to ITL, we have to consider using architectures with multiple levels of abstraction, each with their own representation and learning mechanisms. Take, for example, symbolic-level architectures, which give a modeler complete freedom in how knowledge can be specified. A model of subitizing (i.e., counting dots on a screen), therefore, would not necessarily strengthen a counting skill that can be reused for other tasks. This is not a problem if our focus is on modeling subitizing phenomena, but it becomes one when counting is viewed as a skill that is itself a unit of representation. Critically, different levels may be strongly linked together; that is, a higher level of abstraction can be produced at a lower level. Let us, then, explore the interconnection of levels through composition, where a single unit of abstraction at a certain level can be decomposed into several units at a lower level.

The idea of multiple levels of abstraction is, of course, not new. In 1990, Newell described 12 levels of abstraction, subdivided into four bands (neural, cognitive, rational, and social) that operate at different timescales: from microseconds (neural) to extended periods of time (social). Newell, however, encouraged researchers to "carve nature at its joints"; that is, to settle on a level of abstraction sufficient to model intelligent behavior, below which processes are integral only to implementation. Given our current understanding of neural-level representations, we should not ignore any level of abstraction but rather look for representation, learning, and transfer at different levels of abstraction.

There are, of course, many ways to define a multilevel architecture. Here I wish to initiate the discussion with a proposal that builds on several existing architectures and modeling approaches. Since I will be using ideas from many different sources, I do not wish to "brand" the idea and will refer to it as a "multilevel architecture" (MLA), considered here in the context of ITL.

From "Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions," edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26, J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.

A Multilevel Architecture for ITL

Global Workspace Assumption

The starting assumption for my proposal is that the human brain is to some extent modular and can therefore be subdivided into distinct areas (e.g., perception, motor areas), different types of memory, areas related to control, and possibly (but not necessarily) specialized cognitive functions (e.g., language and numerical cognition). This assumption is visible across several (although not all) levels of abstraction, thus making it necessary to specify how information is made available to different modules, and to have a way of controlling the flow of that information.

A useful starting point is the global neuronal workspace (Baars 1997; Dehaene et al. 1998), a common area in which information is exchanged between modules. But how do modules "know" when to use information from one another? For this, (procedural) knowledge decides, based on the current contents of the workspace, which information has to be moved to a particular area within the workspace. Figure 6.2 illustrates the general concept of the workspace: different modules occupy their own subspace and procedural

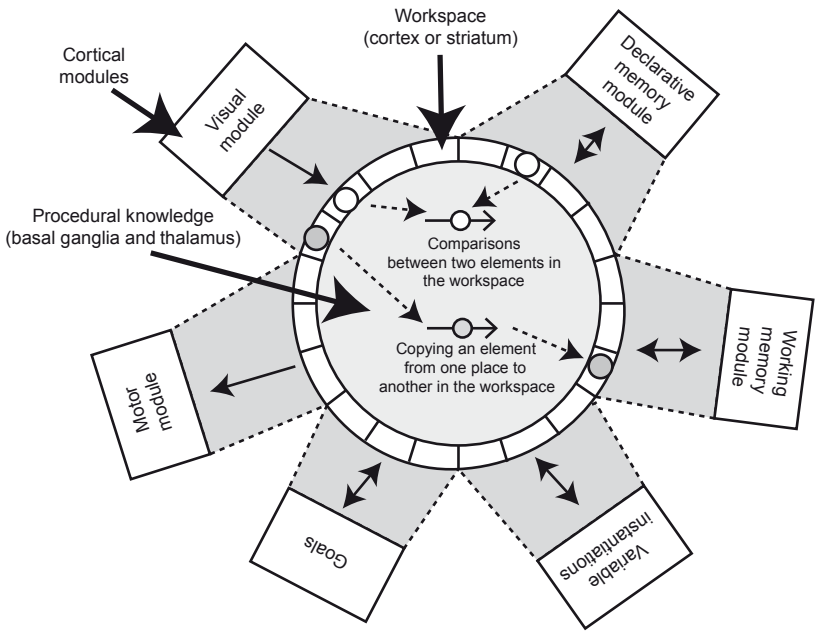

Figure 6.2 The global workspace model (PRIMs version), reprinted with permission from Taatgen (2013).

From "Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions," edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26, J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.

knowledge (in the middle) transfers information between modules. The subspace for each module consists of a number of "slots" which can be used to represent separate information items.

The concept of a global workspace has been applied differently in cognitive science and artificial intelligence, for example, in blackboard architectures (Hayes-Roth 1985) and in buffers in ACT-R (Anderson 2007). I use it here as the central organizing element in the MLA.

Five Levels of Abstraction

As illustrated in Figure 6.3, the MLA consists of five levels of abstraction. Building up from the bottom, these levels are neuronal clusters, primitive operations, operators, goals, and tasks leading to higher-level representations (e.g., language). The general idea is that the units at a particular level serve as the building blocks for a level higher. Between levels, learning mechanisms

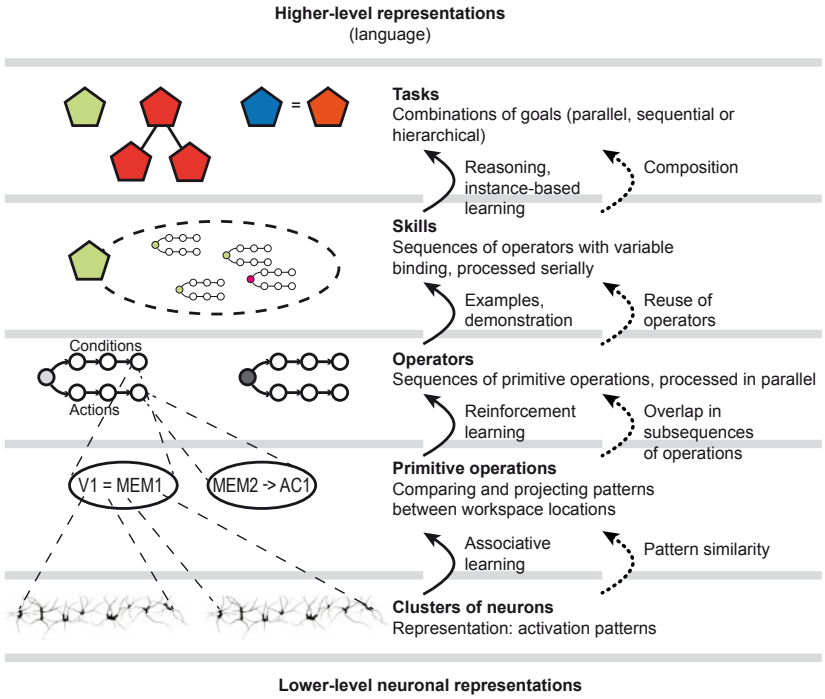

Figure 6.3 Proposal for an MLA with five levels of abstraction. Each level provides the building blocks for one level higher and learning mechanisms operate between levels. The figure indicates some tentative learning mechanisms on the arrows. Also indicated (dotted arrows) is how knowledge can be reused to produce transfer of learning. The activation patterns of the neuron clusters at the bottom of the figure represent the contents of global workspace buffers: V (visual), MEM (memory), and AC (action).

From "Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions," edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26, J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.

operate to build new knowledge representations. Figure 6.3 also illustrates how knowledge can be used for different purposes, providing knowledge transfer between tasks.

Below, I describe how the different levels represent, use, and learn knowledge using two examples: an aural-vocal task and the simple tasks depicted in Figure 6.1. In the aural-vocal task, the model is presented a tone of low, medium, or high pitch and must 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).

Clusters of Neurons

A generally held assumption in neural networks is that meaning is represented by activation patterns in clusters of neurons (e.g., Stocco et al. 2010; Eliasmith et al. 2012). At this level, each slot in the global workspace consists of a cluster of neurons, and different meanings can be attached to different firing patterns. In the aural-vocal task, aural processing of the tone produces a particular firing pattern in the aural module. For this pattern to take on meaning, it must be able to activate certain memories linked to that tone. For this to happen, the tone has to be used as a cue for memory recall. For this, the system has to be able to consider the firing pattern as a single unit that can be used at a higher level of abstraction. Yet the neural representation of, say, a low-pitched tone in memory may differ from that produced by the aural module. Thus, 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). A single symbol at the higher level may therefore correspond to many patterns at the neural level. Learning at this level is typically very slow, requiring many practice trials. If knowledge at this level is missing in an ITL setting, it has to be painstakingly trained before higher levels can integrate and use it (e.g., processing Chinese characters).

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 by a loop through the basal ganglia and the thalamus. Figure 6.4 illustrates how this model performs on the aural-vocal task. The gray boxes represent three sub-areas (vocal, prefrontal, aural) 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 which word corresponds to the pitch of the tone. The basal ganglia 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: "To which vocal output does this tone pitch correspond?" In Stage C, memory has produced

From "Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions," edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26, J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.

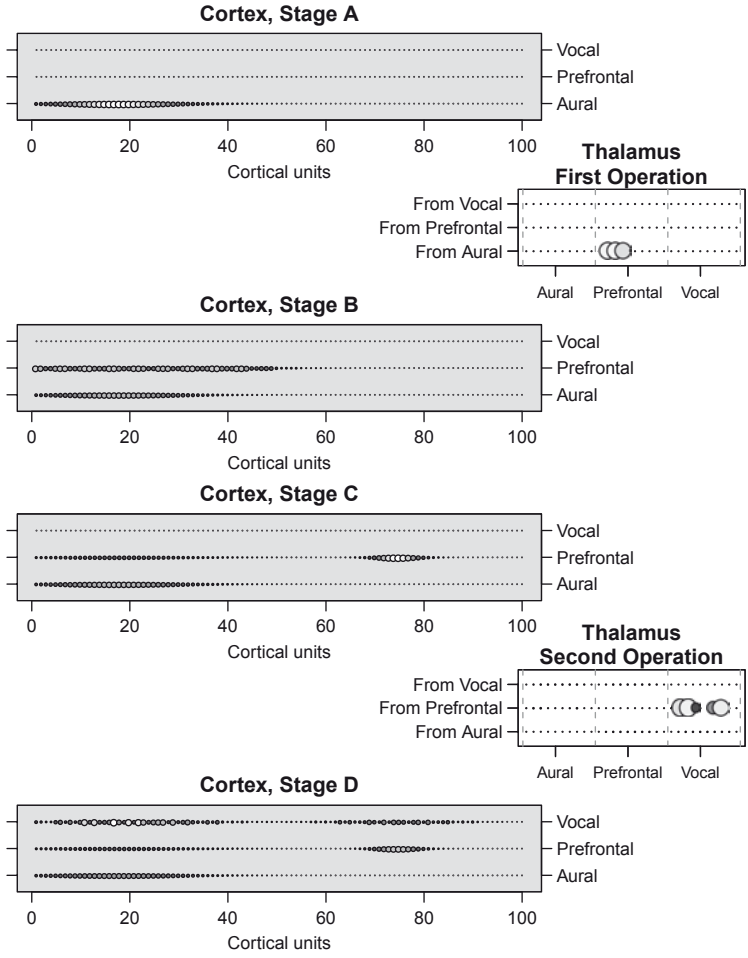

Figure 6.4 Activation in Stocco et al.'s model of the aural-vocal task (Stocco et al. 2010), reprinted with permission.

the relevant memory trace, and a new pattern is created that represents the word that must be spoken. It is now the basal ganglia's turn to transform that memory pattern into an actual speech representation that can be put into the vocal part of the global workspace (Stage D).

Primitive Operations

At this level, neural firings are replaced by symbols that represent specific firing patterns. Instead of specifying the particular pattern of neurons that represents a low pitch, we will just specify "low pitch" in a slot of the aural part of the workspace. By doing so, we discard information (e.g., the ability to

From "Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions," edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26, J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.

determine whether a low pitch is more similar to a medium pitch than it is to a high pitch). This information is still available, assuming that we simulate the lower levels.

In addition, the same representation for the low pitch in the aural part of the workspace is used in the memory part, even though both parts were represented by different patterns in the neural model, and these patterns had to be learned through associative learning.

Activity in the basal ganglia can now be simplified into primitive operations (Taatgen 2013): First, given an input in the first slot of the aural (AU) workspace, copy that representation to the first slot of the memory (MEM) system. Second, given a representation in the second slot of the memory system, copy that representation to the first slot in the vocal (VOC) system:

$$\text{AU1} \rightarrow \text{MEM1}$$

$$\text{MEM2} \rightarrow \text{VOC1}$$

In addition, we need to specify 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. However, actions are always tied to a particular context and may also be augmented by particular conditions. Thus, an aural-vocal task is performed based on skills (to respond to an aural stimulus with a prememorized response), supplied with the particulars (tones with certain pitches, certain words) and guided by the goal to actually perform this task. To do all this, more levels of organization are needed as well as an additional primitive operation; namely, the ability to compare elements in the global workspace. For example, in the aural-vocal task, we only want to respond to specific tones—not to just any sound. Suppose that the aural system places a type of sound 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:

$$\text{AU1} = \text{tone}$$

The problem with this comparison is that it is not a primitive operation, because it contains a particular value ("tone"). At this level of abstraction, we do not want particular values 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:

$$\text{AU1} = \text{C1}$$

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

From "Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions," edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26, J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.

Learning at the level of primitive operations involves finding the right combinations to perform something meaningful, probably through trial-and-error combined with reinforcement learning. Such a combination provides the unit for the next level of abstraction: operators.

Operators

Operators are perhaps the most common form of representation found in cognitive architectures. Soar (Laird 2012) uses 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). Initially, operators carry out a number of primitive operations that test conditions; if these are satisfied, they 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." The tone is represented in AU1 ("Tone") and AU2 (e.g., "Low"). Memory contains chunks like ["Associate," "Low," "One"], which will end up in MEM1, MEM2, and MEM3, respectively. The vocal module can take commands such as ["say," "one"], represented in VOC1 and VOC2.

Operator 1:	Operator 2:
AU1 = C1	MEM1 = C2
C2 → MEM1	C3 → VOC1
AU2 → MEM2	MEM3 → VOC2

Although operators are like production rules, they are not directly linked to a particular task or skill. This makes it possible to reuse operators in many different contexts. To promote reuse of operators, all the particulars are represented in the "C" part of the workspace. Another property of operators, contrary to productions, is that they do not bind variables; instead, they assume relevant variables are supplied in the "C" subspace. Thus, the variable binding problem does not need to be solved at this level of abstraction. The next level of abstraction, the skill, is responsible for supplying values to that space.

Skills

To perform all but the most elementary tasks, several processing steps have to be taken. In other words, several operators are needed that are carried out in sequence. Given that operators only transfer information in the global workspace, operator activity is interleaved with module activity (as in the neural example in Figure 6.4). To get from operators to tasks, we need an intermediate level—the skill level—because even moderately complicated tasks need many operators, so connecting operators straight to tasks is too large a jump.

From "Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions," edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26, J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.

Skills organize which operators are needed and are responsible for instantiating particular values (i.e., binding is carried out at the level of skills). In the aural-vocal example, we specify that the skill 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 skill's behavior, providing a means to use the same skill for different purposes.

For the elementary aural-vocal task, one skill is enough. However, to perform the simple tasks depicted in Figure 6.1, multiple skills are needed in combination to explain why we are able to solve these tasks without prior learning.

Skill-level learning is important in ITL if the learner has not yet mastered all skills required to perform a task. Referring to the paperclip example given in the introduction, a learner might respond: "But I don't know how to make an outline." To remedy this, the teacher could demonstrate how to make an outline of a house using a pencil or verbally offer detailed step-by-step instructions. An effective, but not unique way to teach a skill is to show by example.

Tasks

At the uppermost level of MLA, multiple skills are linked, often in unique combinations, to perform tasks successfully. The underlying idea derives from the concept of compositionality in language, where words and grammar allow us to produce and understand sentences that we have never before heard. This level offers the same versatility: skills and instantiations of skills are used to accomplish tasks never before encountered.

Within a task, skills can be organized in several ways: they can follow each other sequentially, be active in parallel, or organized hierarchically. Consider the simple task in Figure 6.1: "Which aquarium has more red fish?" To respond, we need a *compare* skill, a *count* skill, and a *has-property* skill, organized hierarchically from top to bottom, each instantiated with the appropriate bindings.

To illustrate the compositionality property, consider the other task in Figure 6.1: "Count the number of spoons." This task can be carried out with some of the same skills: the top skill is now an *add-all* skill, but the sub-skills are the same as in the aquarium task, albeit with different bindings.

Figure 6.5 shows the two tasks across several levels of abstraction. Not only do the two tasks share two out of three skills, the two skills which they do not share still overlap in terms of operators as well as in the sequences of primitive operations.

In an ideal ITL situation, the instruction is enough to mobilize the appropriate skills and instantiate those skills to create a task structure. Accordingly,

From "Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions," edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26, J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.

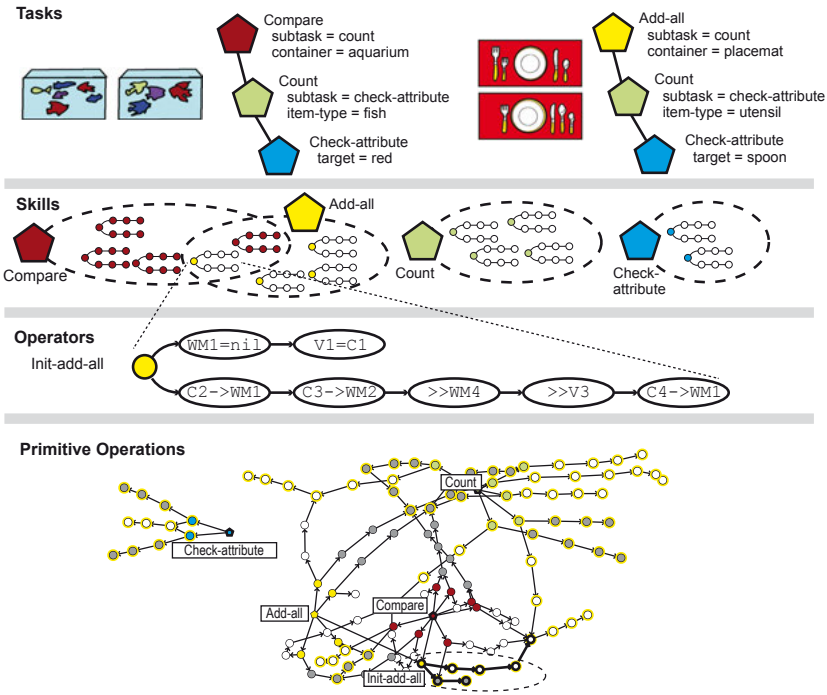

Figure 6.5 Illustration of the representation of the two tasks from Figure 6.1 across levels of abstraction. At the task level each task has a hierarchy of three skills with particular bindings. At the skill level, each skill is associated with a set of operators, which can potentially have overlap. 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 (https://github.com/ntaatgen/PRIMs-Tutorial) in which both tasks are modeled. It shows all the operators with their strings of primitive operations, and where these overlap. For example, the Init-add-all operator shown at the Operators level has two conditions and five actions. This operator is also shown at the Primitive Operations level, where it is outlined with bold lines. Some of the primitive operations at that level are shared with other operators; for example, the two conditions are also used by one of the compare operators.

learning a new task is straightforward, similar to the situation in which a subject in an experiment receives and carries out instructions. A misunderstanding of instructions may be due to a problem of connecting the language of the instruction to the appropriate skill, or because the subject lacks a skill. In that case, the learning process has to drill down to the appropriate level of abstraction in order to identify and fill in the gaps, and the mode of instruction has to be adjusted appropriately.

From "Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions," edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26, J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.

Multilevel Processes for ITL

An interesting property of the MLA is that at the bottom levels, the system is quite mechanical and syntactic in nature (i.e., it lacks "meaning" of what it is doing) whereas the highest levels closely resemble 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? As described above, learning and transfer happen at all levels of abstraction, and thus leaving out levels results in an incomplete system.

Learning

In machine learning, three types of learning are typically 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 type to this: learning by instruction and explicit reasoning. Learning involves 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 be 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 distinguish correct from incorrect. Instead, co-occurring patterns are associated with each other, creating links which the next level above it (primitive operations) needs to function. Moving up in levels, reinforcement learning combined with genetic algorithms provides 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, or by imitation), past experiences, or through reflection (by mentally simulating what the outcome of a certain skill will be). For high-level learning, there is much to learn from Soar (Laird et al. 1986), a cognitive architecture that has 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; knowledge is interconnected at all levels of abstraction. Thus, at every level, there are opportunities for knowledge transfer.

At the neuronal level, similarity between activation patterns signals that these patterns are processed in similar or even identical ways. This similarity originates in perceptual and motor systems, which are connected to the real world. It can, however, also result from associative learning processes that preserve some of the pattern similarities in higher-level representations.

From "Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions," edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26, J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.

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

The largest potential for transfer is in the reuse of skills. By instantiating and connecting skills that it has already acquired, the system is able to address novel tasks.

Conclusion

Because learning and transfer happen at all levels of the proposed MLA, it is important to remember that all levels of abstraction are necessary. There is no natural intermediate level under which one can say that everything is simply implementation detail. The MLA framework has 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. Some of these ideas have already been implemented in the PRIMs architecture (Taatgen 2013), but many open questions remain to be explored with respect to representation, learning, and transfer.

From "Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions," edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26, J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.

From “Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions,”
edited by K. A. Gluck and J. E. Laird. Strüngmann Forum Reports, vol. 26,
J. R. Lupp, series editor. Cambridge, MA: MIT Press. ISBN 978-0-262-03882-9.