Advances in Cognitive Systems 6 (2018) 3–16 Submitted 9/2018; published 9/2018

Theories and Models in Cognitive Systems Research

Pat Langley PATRICK.W.LANGLEY@GMAIL.COM
Institute for the Study of Learning and Expertise, Palo Alto, California 94306 USA
Department of Computer Science, University of Auckland, Private Bag 92019, Auckland 1142 NZ

Abstract

In this essay, I consider the distinction between theories and models in research on cognitive systems. I start by reviewing some classic examples of theories from the history of science, and then discuss four accounts of increasing specificity that focus on intelligent behavior. In each case, I present the framework's postulates concerning cognitive structures, its assumptions about mental processing, and the form of models that connect it to particular domains. After this, I examine some implications for papers on cognitive systems that follow from the difference between theories and models, along with broader issues about the sources of explanatory power.

1. The Science of Cognitive Systems

The field of cognitive systems aims to understand the nature of intelligence and to reproduce it in computational artifacts. By focusing on high-level mental abilities like problem solving, reasoning, and language processing, it shares the goals and style of early artificial intelligence (Langley, 2012). There are many different responses to this challenge, but researchers share a common interest about those facets of the human mind that make us unique in the animal kingdom. However, our pursuit suffers from a problem that has plagued AI since its inception over 60 years ago. If we hope to claim status as a scientific discipline, then we must develop a body of theory, but cognitive systems is an example of what Simon (1969) has called a *science of the artificial*. In such fields, researchers study artifacts they construct, in this case programs that run on digital computers.

What form should theoretical accounts take in our discipline? They should not be the 'theories' associated with algorithmic complexity and other formal branches of computer science, as such results are actually contributions to mathematics. Scientific theories cannot be proved correct; they make empirical claims that can be refuted by observations. Some readers will be tempted to view computer programs themselves as theories, since they may not behave as intended, but this stance falls to a simple counterargument. Suppose a graduate student develops an innovative cognitive system that exhibits some challenging ability and receives a PhD for the work. Two years later, another student reimplements the core ideas in a different programming language and submits a dissertation that describes it. Most would agree that the second person does not deserve a doctorate because the contribution was not the code itself, but rather the more abstract insights it instantiates. The program serves mainly to demonstrate that these ideas actually support the target abilities. When I refer to 'theory' in cognitive systems, I am talking about this abstract level.

In this essay, I examine more fully the distinction between theories and the models that are stated within them. I start by reviewing some classic theories and associated models from the history of science. After this, I consider four theoretical frameworks from the cognitive systems paradigm, some of them dating back to the earliest days of the AI revolution. Next I discuss implications of this analysis for research papers, which should state problems, describe theories and associated models, and present evidence for them. Finally, I examine broader issues that arise when developing accounts of complex phenomena, including those studied in the field of cognitive systems.

2. Theories and Models in Science

The conventional wisdom states that science examines relations between theory and data. In this account, a theory leads to predictions that we compare to observations, which in turn let us evaluate the theory's adequacy and suggest ways to improve it. This closed-loop characterization of science is alluring but incorrect because theories, by themselves, are not operational or testable. Their abstract specification offers many advantages, but it means that, by themselves, they cannot produce predictions or explain observations without additional assumptions. We often refer to a collection of such assumptions as a *model*.

Some examples from the history of science will clarify this point. Newell and Simon (1976) reviewed a number of important theories, although they used the phrase *laws of qualitative structure*. One is Dalton's (1808) *atomic theory*, which states that macroscopic objects are composed of many tiny molecules, each of which contains one or more atoms of elements that cannot be decomposed further. Chemical reactions transform molecules of some types into ones of different types by rearranging their elements. Another is the *germ theory* (Pasteur, 1880), which says that diseases are caused by microscopic organisms that invade and attack the body, and which are spread from one host to another. A third is the geological theory of *plate tectonics* (Hess, 1962), which claims that the Earth's surface comprises a set of plates that move very slowly under, over, and against each other, leading to mountains, trenches, and other formations.

Note that each of these theories is quite abstract, in that it describes generic types of entities and qualitative relations among them. The atomic theory, at this level, cannot explain particular substances like water or ammonia. This requires that one adopt more specific assumptions about their constituent elements and their numbers. In the same way, the germ theory cannot, on its own, predict which microorganisms lead to consumption or measles, or how these diseases are communicated. This requires additional statements that particular species are responsible and that they are transferred, say, by contact, air, or water. Similarly, the idea of plate tectonics does not suffice to explain particular landforms or specific continental motions. For this, one needs postulates about the size and location of plates, as well as the direction and rate of movement. Such modeling assumptions must bridge the gap from abstract theory to testable predictions.

Many scientific theories are qualitative in character, but the same points hold for quantitative accounts like Newton's theory of gravitation. This revolves around two key postulates. The first is that an object will continue to move in a straight line, at a fixed velocity, unless some force is applied. The second is that the gravitational force between two objects is directly proportional to the product of their masses and inversely proportional to the square of their distance. The actual motion of any object is the resultant of these two factors. However, despite its explicit numeric

equations, we cannot use these relations alone to anticipate the orbits of the moon or planets. For this purpose, we must introduce assumptions about the masses, positions, and velocities of the bodies whose trajectories that we want to predict. The standard description of the solar system is a model that specifies these quantities for the Sun, the planets, and their satellites. Such models are stated *within* a theory, but they extend it in ways that make it testable and refutable.

Note that each of these theories includes statements about *structures*, which specify a set of entities and relations among them, and *processes*, which operate to maintain or alter them. The atomic doctrine states that molecules are made up of elements and that reactions rearrange elements to produce new ones. The germ theory assumes that microscopic organisms reside in some human bodies and that disease spreads by transferring them to other bodies, where they reproduce. Plate tectonics postulates that the Earth's surface is composed of large, interconnected segments, which shift over time to form mountain chains and trenches. We will see that a similar division between structures and processes arises in theories of intelligent behavior.

3. Theories and Models in Cognitive Systems

Now that I have introduced the distinction between theories and models in science, I can clarify its relevance to cognitive systems research by analyzing a series of examples. In each case, I review the core postulates of a familiar theory, first discussing its mental structures and then the processes that operate over them. The section also provides instances of models stated within each theoretical framework. I start with an abstract theory of intelligence and present more specific ones that retain their predecessors' assumptions and introduce new constraints.

3.1 Physical Symbol Systems

Artificial intelligence and cognitive systems are concerned with the structures and processes that enable intelligent behavior. Research on these topics has relied on list processing, a distinctive approach to computing that has played a central role in the design and construction of AI systems. In revisiting their early contributions to the field, Newell and Simon (1976) recast their ideas in more general terms, leading to the claim that *physical symbol systems offer the means for general intelligent action*.[1] This idea, which they called the *physical symbol system hypothesis*, has been a core enabler of AI progress since its introduction, in less explicit terms, in the late 1950s.

Newell and Simon elaborated on what they mean by physical symbol systems, starting with their structural characteristics. Such a system incorporates:

- *Symbols* – physical patterns that remain stable unless they are modified by some activity.
- *Symbol structures* – organized sets of such symbols that, taken together, form *expressions*.
- *Designations* – symbol structures which denote or link to other structures or processes that reside in internal stores or that occur in the external world.

These ideas are agnostic about the substrate of the physical patterns; equivalent structures can reside in radically different media, including neurons, vacuum tubes, silicon chips, paper, and blackboards. Any persistent set of physical patterns can serve in this capacity.

1. The presentation here differs slightly from Newell and Simon's version. For instance, they referred to 'necessary and sufficient' means, but these modifiers are not important to the aims of the current discussion.

The theory of physical symbol systems also includes postulates about processes that operate over these building blocks. These include:

- *Interpretation*. A symbol system can 'execute' or 'run' structures that designate processes by carrying out their specified steps.
- *Creation and modification*. Interpreting such designated processes can create new symbol structures and modify existing ones.
- *Extended operation*. A symbol system operates over time to produce an evolving collection of symbol structures.

Newell and Simon pointed to four developments during the 20th century that led to the theory of physical symbol systems: formal logic and symbol manipulation, Turing machines and digital computers, the concept of a stored program, and list processing languages like IPL and Lisp. List processing played a key role in AI's development because it could encode arbitrarily complex structural descriptions, create new structural descriptions dynamically, use them to designate other structures, and interpret the structures to produce behavior. These abilities supported the abstraction of complex content beyond the specific hardware on which they were implemented, and they proved crucial in the construction of early cognitive systems.

We can view any program implemented on a digital computer as a model stated within the theory of physical symbol systems. Each such program specifies details that, when run on some hardware platform, will generate behavior. Not all such computer programs will exhibit intelligence, but the many examples of implemented cognitive systems – from the early Logic Theorist (Newell, Shaw, & Simon, 1957) to the recent AlphaGo (Silver et al., 2016) – provide evidence of the theory's ability to explain and reproduce many aspects of the mind. These successes do not prove that the theory's tenets are the only such account, or that it can explain every aspect of intelligence, but the number, breadth, and abilities of implemented cognitive systems offer strong support for them.

3.2 Production Systems

Although widely adopted, the physical symbol system hypothesis is a weak theory that provides few constraints on the construction of intelligent systems. Consider some well-established features of human cognition: they exhibit flexible, conditional behavior; they balance stimulus-driven and goal-driven activity; and they acquire new expertise in an incremental, piecemeal manner. These suggest that human intelligence, at least, does not rely on the types of procedural constructs adopted by mainstream programming languages, despite them being examples of physical symbol systems. In response, Newell (1966) introduced a more constrained version of the theory – *production systems* – that addresses these observed characteristics of the human mind.

The framework includes several new theoretical assumptions about the memories that underlie cognition and the structures they contain:

- *Modularity*. Each memory comprises a set of separate, modular elements that are encoded as symbol structures.
- *Working memory* contains concrete, descriptive elements that encode beliefs, goals, or other specific structures and that change rapidly over time.
- *Production memory* contains generic rules that specify the conditions under which to take certain actions and that remain static or change very slowly.

These representational tenets are very different from those adopted by traditional, procedure-oriented programming languages. There is a much stronger connection to behaviorism's notion of stimulus-response pairs that link perceptions to actions, although production systems replace perceptions with elements in working memory and responses with commands to alter them.

The production system framework also makes assumptions about the mechanisms that operate over these mental structures:

- *Recognize-act cycle*. Cognitive processing alternates among finding productions whose conditions match working memory, selecting a subset of them, and executing their associated actions.
- *Pattern matching*. Production rules are accessed by matching their conditions against elements in working memory.
- *Conflict resolution*. When there are multiple matches, features of the matched elements or the rules themselves determine which ones to select.
- *Dynamic composition*. Production rule application alters working memory in ways that enable other matches on later cycles, including composition of matched elements into new entities.

Despite these additional constraints, production systems have the same computational power and generality as many other frameworks, although they make it easier to produce some types of behaviors and more difficult to handle others.

The introduction of production systems had a major impact on both AI and cognitive psychology starting in the late 1970s. They became widely used for the creation of expert systems (Hayes-Roth, Waterman, & Lenat, 1983), which were connected to AI's first commercial successes. They also played a key role in early research on machine learning, especially in areas like problem solving and language processing. Many cognitive architectures (Langley, Laird, & Rogers, 2009) have either been cast in the production system framework or adopted its assumptions about the recognize-act cycle and pattern matching. Neches, Langley, and Klahr (1987) trace the evolution of production systems during the framework's first two decades. This period saw many specializations of the theory, often descended from OPS (Forgy & McDermott, 1978), that incorporated further constraints on representation and processing. More recent well-known variants include Soar (Laird, Rosenbloom, & Newell, 1987; Laird, 2012), ACT-R (Anderson, 1993), and EPIC (Kieras & Meyer, 1997).

Each of these architectures comes with a programming language whose syntax embeds theoretical assumptions about memories and the representation of their contents. The syntax itself is no more part of the theory than Newton's notation for integrals was part of universal gravitation, but it nevertheless reflects key intellectual commitments. One can also describe the production system framework at different levels of abstraction. Traditional architectures in this paradigm take positions about conflict resolution and learning mechanisms, but Langley's (1983) PRISM environment included a number of parameters whose combined settings determined system behavior and defined a space of candidate architectures.

Even specialized architectures like Soar and ACT-R remain abstract theories that must be combined with models to produce behavior, with different sets of production rules and initial contents of working memory playing this role. Researchers often develop such models to demonstrate their theory's implications for behavior in different domains or on various problems in a given domain, but they can also encode alternative strategies for solving a class of problems. Assumptions about the contents of production and working memory effectively specify a program that we can run us-

ing the production system interpreter to produce behavior traces, which we can then compare with target behaviors to evaluate their adequacy. The status of architecture implementations is less clear cut, as they must introduce choices about data structures and methods, such as whether to use iteration or recursion when defining a function. At first glance, these choices seem similar to modeling assumptions. However, as long as different implementations produce the same behaviors, they are better treated as alternative statements of the same architectural theory than as distinct models.

3.3 Heuristic Search

Human intelligence includes the ability to solve novel problems never before encountered. Plato first posed this apparent conundrum in his *Meno*: How can we find solutions to problems when we do not already know the answers? One response, familiar to all AI researchers, is that we separate the generators of candidate solutions from the mechanisms that test them. This division eliminates the apparent paradox, but it requires the ability to represent candidate solutions mentally and to explore the resulting space of possibilities. Yet such spaces can be incredibly large, which in turn means that we cannot in practice store them in advance or enumerate them explicitly.

This insight and challenge led Newell and Simon (1976) to propose their *heuristic search hypothesis*, which, despite its name, is actually another theory. We can paraphrase its structural postulates as stating that a problem solver relies on:

- *Candidates*, which describe patterns that may or may not be acceptable solutions;
- *Test criteria*, which specify how to determine if a candidate is an acceptable solution;
- *Generators*, which indicate how to generate new candidates from scratch or from existing ones;
- *Heuristics*, which characterize the quality of generator steps or candidate solutions.

One can represent candidate solutions, tests, generators, and heuristics in different ways, but they always take the form of symbol structures, making it an elaboration of the physical symbol system hypothesis. Another central element of the theory is a *problem space*, which is the set of candidate solutions. Strictly speaking, this is not itself a symbol structure, as it is defined implicitly in terms of the other elements. A subset of the problem space may be constructed during problem solving, but it often contains so many candidates that it is best treated as virtual.

Naturally, Newell and Simon's theory of problem solving also postulated a set of mental processes that inspect and manipulate these symbol structures. Their framework included three interacting component mechanisms:

- *Searching* through the implicit problem space by generating candidate solutions;
- *Testing candidates* to determine whether they constitute acceptable solutions; and
- *Using heuristics* to guide search through the problem space, making it tractable.

There are many routine tasks that do not require one to carry out search through a problem space, but nearly any nonroutine task can be viewed usefully in these terms, provided it does not have trivial or obvious solutions.

The heuristic search framework has been applied repeatedly in the construction of AI systems and has led to many successes. Application areas have included automated reasoning, planning, game playing, design, scheduling, and language processing. Each of these adopt modular representations for candidate solutions, involve large problem spaces, specify criteria that solutions must

meet, and use heuristics to guide search. Even approaches that emphasize numerical processing, like many methods for statistical learning, almost invariably rely on search as one of their core techniques. In fact, the theory of heuristic search has become so widely adopted in AI that many researchers have difficulty imagining any other account of problem solving, despite alternatives offered by Gestalt theory (Koffka, 1935), which emphasized the role of spontaneous restructuring.

Although the heuristic search theory is certainly more constrained than the framework of physical symbol systems, it remains very general, so it is not surprising that researchers have developed more specialized accounts that make additional assumptions. These include:

- *Planning*, in which candidate solutions are partial plans that attempt to link an initial problem state to a goal description and the generator adds domain operators to existing candidates. Both partial-order and total-order approaches fall in this category.
- *Problem-reduction search*, in which candidate solutions are AND trees that decompose a problem into subproblems and the generator uses domain rules to elaborate existing candidates. Most theorem provers take this approach, but means-ends analysis also fits in the framework.
- *Constraint satisfaction*, in which candidate solutions have values associated with each target variable and the generator assigns possible values to those variables. Candidates that violate specified constraints are eliminated from consideration.
- *Repair-space search*, in which one starts with a candidate solution that is unsatisfactory in some way and the generator produces variant structures in an attempt to find a solution. Case-based reasoning, 'local' search methods, and evolutionary techniques all incorporate this idea.[2]

These formulations makes different commitments about the character of candidate solutions, the generators that produce them, and the heuristics that guide choice. Each is a specialization of the heuristic search theory, but they are not mutually exclusive. For instance, constraint satisfaction is often combined with repair-space search, and planning systems may invoke problem reduction. One can also combine production systems and heuristic search in different ways, using the former to specify both generators and heuristics (e.g., Langley & Ohlsson, 1984; Laird et al., 1987).

As before, we must introduce modeling assumptions before a given heuristic search theory can produce behavior. This means that we must specify some notation for candidate solutions and commit to particular generators, test criteria, and heuristics for use during processing. For a planning system, we must specify an initial state, a goal description, and a set of operators for generating new states, as well as heuristics for guiding search. A problem-reduction system also requires a set of decomposition rules that it can use to break tasks into subtasks. Heuristics play a crucial role in these specialized theories because exhaustive search is impractical for large problem spaces.[3] These may take the form of symbolic rules that propose, reject, and order solution candidates or their transformations; alternatively, they may be stated as numeric functions that evaluate structures. Some theories make commitments about the form of heuristics, but their content is part of the model. Once we have specified these details, we can treat the model as a computer program and run it to produce behavior, which we can then examine for effectiveness during the search process.

2. One can also view hill-climbing and gradient-descent methods as examples of repair-space search, although generators for the latter compute the next candidate directly, rather than considering a number of alternatives.

3. As Langley (2017) has noted, the term *heuristic* originally denoted criteria that do not guarantee finding the best solution or, indeed, any solution at all, but that in practice lead to reasonable results with reasonable effort.

3.4 The HPS Architecture

The heuristic search theory reflects important insights about problem solving, but we know much more about the structures and processes involved that it enumerates. For example, there is strong evidence that humans often rely on means-ends analysis (Newell, Shaw, & Simon, 1960) to tackle novel problems, although they do not use this strategy in all settings. The HPS architecture (Langley, Barley, & Meadows, in press) offers an extended theory of problem solving that builds on Newell etal.'s insights. The framework incorporates both the *physical symbol system* hypothesis and the *heuristic search* hypothesis, but also introduces four structural postulates:

- Problems are specified as an initial state that comprises a set of relational literals and a goal description that includes a set of generic literals that denote a class of states.
- Domain operators describe the generic conditions under which a given action will have particular effects on states; these serve as generators for new candidate solutions.
- Candidate solutions are structured as hierarchical decompositions of problems, with each decomposition including an associated operator and optional 'down' and 'right' subproblems.[4]
- Candidates are organized as nodes in an OR tree in which each child elaborates on its parent by introducing one operator and zero, one, or two associated subproblems. Alternative children specify different elaborations on their parent's partial solution.

The first and second statements identify the HPS theory as a specialization of the planning paradigm discussed earlier, although it is not limited to action-oriented tasks. The third assumption reveals its close kinship with means-ends analysis, which introduced the idea of using domain operators to decompose problems into subproblems. The final assumption shows its relation to refinement approaches to plan generation (Kambhampati, 1997), which emphasizes this idea.

The HPS theoretical framework also makes assumptions about the processes that inspect, create, and modify these mental structures:

- Problem solving involves search through a space of alternative hierarchical decompositions that aim to transform the initial state into one that satisfies the goal description.
- Search operates in discrete cycles that either return found solutions, mark a candidate *C* as a solution or unacceptable, attempt to retrieve an operator for *C*, create and evaluate a child of *C* with an operator, use an operator to create subproblems, or mark a subproblem as solved.
- Strategic parameters are the locus of heuristic control during search, determining the evaluation and selection of operators and nodes, as well as criteria for success and failure.

The final postulate lets it retain the key ideas of means-ends analysis without its commitment to chaining only off operators that achieve goals. HPS replaces this control scheme with parametric options, with the traditional approach being a special case. Otherwise, it makes very similar assumptions to PRODIGY (Carbonell et al., 1990) and ICARUS (Choi & Langley, 2018), architectures that combine means-ends problem solving with logical formalisms for states and goals.

Modeling assumptions for the HPS theory are similar to those adopted for planning systems. This includes specifying representations for the states and goal descriptions used in problems, along

4. A given candidate solution maps onto a unique sequential plan, although a particular plan may be decomposed in different ways. Not all decomposition trees solve the top-level problem fully, so they are best viewed as partial plans.

with one for the operators used to generate new states and decompositions. Any model must also indicate settings for parameters that control the evaluation of operators and nodes, how the system uses these scores to select among candidates, and how it decides whether it should reject a candidate or accept it as a solution. Given these details, we can run the HPS architecture on particular problem-solving tasks and examine its search behavior.

3.5 Other Examples from Cognitive Systems

I should also consider briefly some other well-known classes of theories with relevance to cognitive systems. One widely adopted paradigm, which I will call *derivation systems*, assumes that knowledge is encoded as inference rules and that beliefs and queries take the form of relational literals. Given some query, a reasoning mechanism attempts to find derivations or proofs that connect answers to the initial beliefs through the available rules. Derivations are organized as AND trees in which nonterminal nodes correspond to inferred beliefs that follow directly from instantiated rules with antecedents that match other derived or given beliefs. Constructing derivations typically involves search through a space of possible proof structures, making it a special case of both physical symbol systems and heuristic search. Models take the form of particular sets of inference rules, initial beliefs, and queries, along with assumptions about their representation. This framework, one of the oldest in AI, has seen many successful applications, from proving theorems about logic (Newell et al., 1957) to answering questions posed in natural language (Waldinger et al., 2018).

Another theoretical paradigm – *analogical reasoning* – is less widely reported but equally general. This postulates that knowledge is stored as a set of cases, each comprising a set of relational literals that may share arguments. Given a new, partially described case C, the analogical process retrieves similar structures from memory and determines how each of these candidates maps onto C. Based on the quality of these mappings, it selects one of the candidates and uses its content to make inferences that elaborate on C's material. Each model in this framework is stated as a specific set of cases, along with assumptions about their encoding. Analogical reasoning has been used to understand narratives, solve novel problems, and generate scientific explanations. This theory is a specialization of physical symbol systems, but it assumes larger-scale structures than production systems and it views reasoning not in terms of search through a problem space but rather as a variety of retrieval from episodic or long-term memory.

A final class of theories – *recurrent neural networks* – has received increased attention due to its association with the 'deep learning' movement. This paradigm assumes that knowledge is encoded as a multilayer neural network, with short-term memory corresponding to activations on its nodes. The network structure links the output nodes to some of the input nodes, ensuring they have the same activations. Processing occurs in cycles, with activation spreading through the network from inputs to outputs, which in turn determine some inputs on the next round. Recent variants have included a 'long short-term memory' that uses specialized elements that retain previous activations across many cycles. Learned models associate specific weights with links in the network, with details depending on the training cases encountered. Recurrent neural networks have been applied successfully to speech recognition, language translation, and robotic control. Despite claims to the contrary, the framework is a special case of physical symbol systems and, although it adopts a distributed representation, it also bears a strong resemblance to production system architectures.

4. Reporting Research on Cognitive Systems

Now that we have examined the character of theories and models that arise in cognitive systems research, we can turn to the implications for publications that report them. This section considers briefly how authors can productively specify the abilities they aim to reproduce, describe the theory that attempts to address them, present models cast within that theoretical framework, and report evidence about their joint adequacy and plausibility.

Scientific papers often start by presenting a *problem* that motivates the authors' research. In the natural sciences, the problem typically involves explaining some interesting or challenging phenomena or behavior. In sciences of the artificial (Simon, 1969), the problem involves replicating or generating some desired behavior in a construct. In cognitive systems, we are concerned with explaining and reproducing aspects of intelligence in computational terms. One natural way to describe cognitive abilities and behaviors is to specify their inputs and associated outputs. For instance, we can specify the ability to understand a language in terms of inputs – knowledge about the language and a sentence in that language – and results – one or more meanings of the sentence. Similarly, the inputs for planning include knowledge about domain actions, an initial state *I*, and a goal description *G*, whereas the output is one or more plans that transform state *I* into another satisfying *G*. We can specify learning abilities in much the same way, except they must occur in the context of some performance task on which improvement should occur. Such problem descriptions set the stage for theoretical accounts of the abilities they specify.

Once an author has stated a problem in terms of desired abilities, he can present a *theory* designed to address them as a set of linked assumptions. Some statements will be definitions, while others will postulate relations or interactions that hold among the defined elements. Not all tenets need be novel; theories often builds on earlier ones, incorporating key ideas from predecessors. Some researchers will choose to convey their theory in formal terms, using logical expressions or equations to define elements and describe relations, but natural language is also acceptable. Either way, they should state their assumptions clearly enough that they are unambiguous to readers. As noted earlier, it makes sense to focus on structures before processes. For papers on cognitive systems, it is natural to first note the memories a theory assumes, such as a dynamic short-term store and a stable long-term one, along with the types of entities that populate them, such as beliefs, goals, and production rules. Authors can then present postulates about processes that act upon these structures, including statements about high-level control and others about component mechanisms. Thus, authors might describe the recognize-act cycle for a production system, along with processes for pattern matching and conflict resolution.

As we have seen, theories are intentionally abstract, so we must combine them with *models* to produce particular behaviors, which we can then compare with targets to gain evidence that supports or refutes the abstract account. This is distinct from describing the theory's implementation in some programming language, although that is also important; authors should provide some information about their software, but high-level facts will typically suffice. However, papers should offer more details about modeling assumptions for particular domains and scenarios. Some information will encode general knowledge about the domain, such as the grammar for a given linguistic theory or operators for a given planning framework. Other content will describe particular tasks, such as sentences that the parser should interpret or problems that the planner must solve. For the-

ories of learning, authors should describe observations or instructions that the system encounters. Publications on cognitive systems should state clearly what form these elements take, what content they contain, and how they connect the implemented theory to particular scenarios or test cases. Of course, papers do not have enough space to give details about every problem or domain, but they should include examples that clarify such design decisions for readers.

The abstract character of theories makes them difficult to evaluate directly, as we can only measure directly the adequacy of models stated within them. This does not mean such evaluation is impossible, only that it is seldom an open and shut case. Papers on cognitive systems should provide evidence that a theory supports the desired abilities, but presenting results on multiple domains or test cases is more compelling because, to the extent they involve different models, it suggests the theory provides the source of explanatory power. Theories are difficult to refute outright, as this would mean showing that they fail to explain the target phenomena with any possible modeling assumptions. However, if researchers must complement their theory with complicated models, analogous to Ptolemaic epicycles, they are less plausible. Authors should state the criteria they will use to evaluate their theory, present results that buttress or undermine it, and explain how they reached their conclusions. Detailed analysis of system behavior on a few cases can often reveal more about the contribution of postulates than aggregate results on many tasks. The distinction between theories and models raises many issues for evaluation that deserve more extended treatment, but generally authors should be wary of drawing overly strong conclusions in a single publication.

5. Concluding Remarks

Before closing, I should discuss some issues about theories of cognitive systems that the earlier treatment did not address. The first concerns the character of postulates about structures that underlie intelligent behavior. The theories we have examined emphasized the *form* of individual items, such as production rules or working memory elements, and their organization in memory, such as search trees. However, some accounts instead make claims about the *content* encoded in an intelligent system's memories. An early example was Schank's (1972) conceptual dependency framework, which posited 12 'primitive' relations responsible for encoding sentence meanings. In a similar vein, Cassimatis (2004) reported an account of language understanding that involved reasoning over spatial, temporal, componential, and other relations. Such content theories may not take stances about the details of processing, making them closer to Newell's (1982) 'knowledge level' than to classical cognitive architectures. However, the two approaches are complementary, and researchers should consider bringing them both to bear in efforts to explain intelligence.

Another issue, already discussed in passing, is that theories occur at different levels of abstraction. For example, we have seen that the HPS account of problem solving is a specialization of heuristic search, which in turn elaborates the physical symbol system hypothesis. More abstract theories are less constrained, which effectively means they have more degrees of freedom to reproduce target behaviors and place a greater explanatory load on modeling assumptions. This makes an abstract theory more difficult to refute, as its proponents can always claim the problems lie with the model. Examples of such reasoning are common in the history of science, from Ptolemy's addition of epicycles to preserve the assumption of circular motion to variations on the phlogiston theory to defend its view of heat as a fluid. As a discipline progresses, it seems natural for theories to become

less abstract and more constrained. If the extended accounts remain consistent with a broad set of phenomena as this takes place, it lends them greater credibility than their vaguer predecessors.[5]

This leads naturally to a third issue. In some cases, the main source of explanatory power lies not with the theory itself, but with modeling assumptions. Consider the NETtalk system (Sejnowski & Rosenberg, 1987), which learned to predict the phonemes for letters in English sentences from their surrounding context. Although word pronounciation is a sequential process, the system used a moving window to transform it into a classification task that it mastered with backpropagation, a supervised method for learning multilayer neural networks. Many in the research community concluded that the latter was responsible for NETtalk's success, but it seems likely that other techniques, such as decision-tree induction, could have done just as well when coupled with a moving window, which was a modeling assumption. Another example involves the interactive activation model (McClelland & Rumelhart, 1981) and EPAM (Richman & Simon, 1989) accounts of word and letter recognition. These theories adopted quite different postulates about representation and processing, yet both fit experimental results very well, possibly because they assumed the same hierarchical encoding for words. Researchers should be careful when drawing conclusions about the sources of their theories' explanatory power.

A special case of this problem arises with reductionist accounts of complex behaviors. This paradigm typically combines a simple, abstract theory with a model that includes many elements, then shows how target behaviors follow from their interactions. The field of meterology adopts this approach to explain and predict weather from a few basic equations and a fine-grained spatial grid. Reductionist accounts are also popular with advocates of statistical induction, especially ones in the 'deep learning' movement, which relies on training systems with many parameters on very large data sets. The learning theory itself is abstract and simple, but different methods often produce similar results on the same data. This suggests that the explanatory power resides not in the theory of learning but in the training cases, which effectively serve as modeling assumptions. Researchers who adopt such reductionist schemes usually view them as superior to accounts that incorporate stronger constraints, but this reveals a misunderstanding of theories' role in science.

In the previous pages, I claimed that theories and models play different roles in scientific research and that it is important to delineate them. I reviewed some examples from the history of science, arguing that theories propose abstract principles and models introduce enough additional assumptions to make them operational. After this, I described four theories relevant to cognitive systems, in each case stating their postulates about mental structures, the processes that operate over them, and the form taken by associated models. I also examined implications of the theory-model distinction for papers in our discipline, along with issues related to the sources of explanatory power. Theories come in different forms and occur at many levels of abstraction, but even the most detailed accounts remain incomplete without models to elaborate them. At the same time, models always exist in the context of some theory that they serve to make operational. Both have central roles to play in the field of cognitive systems, but it is essential that we keep these roles distinct in both our research activities and our publications.

5. This may not always be possible. Modeling assumptions are more complex in biology than in physics or chemistry, partly from variations among species and partly from the contingent character of evolution. Given the variability of human cognition, there are good reasons to expect similar complexity of modeling assumptions in our field.

Acknowledgements

This essay was supported by Grants N00014-15-1-2517 and N00014-17-1-2434 from the Office of Naval Research, which is not responsible for its contents. I thank Will Bridewell, Ben Meadows, and Stephanie Sage for discussions that helped refine the ideas in this paper.

References

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

Carbonell, J. G., Knoblock, C. A., & Minton, S. (1990). PRODIGY: An integrated architecture for planning and learning. In K. VanLehn (Ed.), *Architectures for intelligence*. Hillsdale, NJ: Lawrence Erlbaum.

Cassimatis, N. L. (2004). Grammatical processing using the mechanisms of physical inference. *Proceedings of the Twentieth-Sixth Annual Conference of the Cognitive Science Society* (pp. 192–197). Chicago, IL: Cognitive Science Society.

Choi, D., & Langley, P. (2018). Evolution of the ICARUS architecture. *Cognitive Systems Research*, *48*, 25–38.

Dalton, J. (1808). *A new system of chemical philosophy* (Part 1). London, UK: R. Bickerstaff.

Forgy, C. L.., & McDermott, J. (1978). *The OPS2 reference manual*. Technical Report, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.

Gentner, D., & Forbus, K. (1991). MAC/FAC: A model of similarity-based retrieval. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society* (pp. 504–509). Chicago: Lawrence Erlbaum.

Gabaldon, A., Langley, P., & Meadows, B. (2014). Integrating meta-level and domain-level knowledge for task-oriented dialogue. *Advances in Cognitive Systems*, *3*, 201–219.

Hayes-Roth, F., Waterman, D. A. & Lenat, D. B. (Eds.) (1983). *Building expert systems*. Boston: Addison-Wesley.

Hess, H. H. (1962). *History of ocean basins*. In A. E. J. Engel, H. L. James, & B. F. Leonard (Eds.), *Petrologic studies: A volume to honor A. F. Buddington*, 599–620. Boulder, CO: Geological Society of America.

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

Koffka, K. (1935). *Principles of Gestalt psychology*. New York: Harcourt, Brace, & Co.

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

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

Langley, P. (1983). Exploring the space of cognitive architectures. *Behavior Research Methods and Instrumentation*, *15*, 289–299.

Langley, P. (2012). The cognitive systems paradigm. *Advances in Cognitive Systems*, *1*, 3–13.

Langley, P. (2017). Heuristics and cognitive systems. *Advances in Cognitive Systems*, *5*, 3–12.

Langley, P., Barley, M., & Meadows, B. (in press). Adaptive search in a hierarchical problem-solving architecture. *Advances in Cognitive Systems*.

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

Langley, P., & Ohlsson, S. (1984). Automated cognitive modeling. *Proceedings of the Fourth National Conference on Artificial Intelligence* (pp. 193–197). Austin, TX: Morgan Kaufmann.

McLelland, J. L. & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, *88*, 375–407.

Neches, R., Langley, P., & Klahr, D. (1987). Learning, development, and production systems. In D. Klahr, P. Langley, & R. Neches (Eds.), *Production system models of learning and development*. Cambridge, MA: MIT Press.

Newell, A. (1966). *On the analysis of human problem solving protocols*. Technical Report, Department of Computer Science, Carnegie Institute of Technology, Pittsburgh, PA. Reprinted in J. C. Gardin & B. Jaulin (1968), *Calcul et formalisation dans les sciences de l'homme*, 146–185. Paris.

Newell, A. (1982). The knowledge level. *Artificial Intelligence*, *18*, 87–127.

Newell, A., & Simon, H. A. (1976). Computer science as empirical enquiry: Symbols and search. *Communications of the ACM*, *19*, 113–126.

Newell, A., Shaw, J. C., & Simon, H. A. (1957). Empirical explorations of the Logic Theory Machine. A case study in heuristics. *Proceedings of the Western Joint Computer Conference* (pp. 218–230) New York: Institute of Radio Engineers.

Newell, A., Shaw, J. C., & Simon, H. A. (1960). Report on a general problem-solving program for a computer. *Proceedings of the International Conference on Information Processing* (pp. 256–264). UNESCO House, France: UNESCO.

Pasteur, L. (1880). On the extension of the germ theory to the etiology of certain common diseases. *Comptes rendus, de l'Academie des Sciences*, *xc*, 1033–44.

Richman, R., & Simon, H. A. (1989). Context effects in letter perception: Comparison of two theories. *Psychological Review*, *96*, 417–432.

Schank, R. C. (1972). Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, *3*, 552–631.

Sejnowski, T. J. & Rosenberg, C. R. (1987). Parallel networks that learn to pronounce English text. *Complex Systems*, *1987*, 145–168.

Silver, D. et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*, 484–489.

Simon, H. A. (1956). Rational choice and the structure of the environment. *Psychological Review*, *63*, 129–138.

Simon, H. A. (1969). *The sciences of the artificial*. Cambridge, MA: MIT Press.

Waldinger, R., Condoravdi, C., Richardson, K., & Suenbuel, A. (2018). Natural language access: When reasoning makes sense. *Advances in Cognitive Systems*, *6*, 17–29.