Integrating Cognitive Models Based on Different Computational Methods

Nicholas L. Cassimatis (cassin@rpi.edu)
Rensselaer Polytechnic Institute
Department of Cognitive Science
110 8th Street
Troy, NY 12180 USA

Abstract

The mind integrates cognitive and perceptual processes that are currently best modeled using many different computational mechanisms. These are difficult to integrate into a single cognitive model and are thus a challenge to developing a unified model of human cognition. This paper presents two computational principles that identify common elements among these mechanisms and uses these principles, together with empirical findings about human cognition, to motivate a theory of human cognitive architecture. This theory has been embodied in the Polyscheme cognitive architecture, which enables models based on qualitatively different computational mechanisms, including Bayesian Networks, search and production systems, to be integrated into a single cognitive model and thus provides an explanation of how the mind integrates multiple cognitive structures and processes.

Keywords: integrated cognitive models; unified theories of cognition.

Introduction

There are many precise models of individual cognitive processes. These models are based on various formal or computational methods that are often very difficult to integrate with each other. Since thinking in many situations involves multiple cognitive processes, the lack of integration among cognitive models is a severe impediment to achieving a comprehensive model of human cognition.

For example, Bayesian Networks are used (for example in Charniak, 1991) to model how human cognition in situations where events or states have probabilistic relationships to other events or states. ACT-R (Anderson & Lebiere, 1998) uses a production rule-system to model how memory retrieval, the focus of attention and action selection interact during a task. It is not obvious how Bayesian Networks can be used to model many of the processes ACT-R models and vice versa, yet much human cognition in many situations involves uncertainty, memory, attention and action selection. Integrating models from both frameworks is difficult because the algorithms and data structures associated with Bayesian Networks are so different from those in ACT-R.

The importance of this problem is underscored by how much of human cognition, even, for example, the basic and ubiquitous activity of keeping track of one's physical surroundings involves multiple cognitive processes that are currently best modeled using qualifiedly different frameworks. For example, in keeping track of the physical world, young infants, use their visual system to receive information about physical objects; they use memory and mental imagery to keep track of objects they do not see; and they use reasoning to choose interpretations of events that are consistent with physical laws (Spelke, 1990). Perceptual processes are generally modeled by statistical and/or numerical methods, different aspects of memory and imagery have been modeled using multiple computational methods and reasoning is generally modeled using probabilistic, logical or mental model methods. A comprehensive model of physical reasoning would be very difficult to achieve without some advance in the integration of cognitive modeling methods.

Pending this advance in integration, it is difficult to rule out the possibility that models within one of or each of these frameworks do not accurately or completely reflect the mechanisms of human cognition.

This paper provides a three-part solution to this problem. First, two computational principles are describe which find common themes among very different computational methods used to model human cognition. Second, these principles motivate several hypotheses about human cognitive architecture that help explain how the mind integrates very different cognitive processes into one task. Finally, a cognitive architecture that embodies these principles, called Polyscheme, is presented which enables models based on very different computational methods to be integrated into one system, providing an explanation for how the mind integrates these mechanisms.

Computational principles

Two computational insights into the computational methods that underlie many computational cognitive modeling paradigms enable an account of how these can be combined into an integrated theory of human cognitive architecture:

- **Common function principle**. Many reasoning and problem solving strategies can be composed of sequences of the same set of common functions.
- **Multiple implementation principle**. Each procedural unit can be executed using algorithms based on multiple representations.

Common function principle (CFP)

A survey of algorithms from many different subfields of computational cognitive science reveals that many of the same functions form the basis of many different kinds of algorithms. A first approximation of the list of these *common functions* includes:

- *Forward inference*. Given a set of beliefs, infer beliefs which follow from them.
- *Subgoaling*. Given the goal of establishing the truth of a proposition, P, make a subgoal of determining the

truth values of propositions which would imply or falsify P.
- *Simulate alternate worlds*. Represent and make inferences about alternate, possible, hypothetical or counterfactual states of the world.
- *Identity matching*. Given a set of propositions about an object, find other objects which might be identical to it.

The CFP will be justified later by showing in detail how these common functions can implement a wide variety of algorithms. The following rough characterizations of two widely used algorithms in cognitive modeling illustrates how methods from different branches of formal cognitive science can be implemented using the same set of common functions (which are underlined). They will be more precisely characterized later.

Search. "When uncertain about whether A is true, represent the world where A is true, perform forward inference, represent the world where A is not true, perform forward inference. If forward inference leads to further uncertainty, repeat."

Stochastic simulation (used widely in Bayes Network propagation). "When A is more likely than not-A, represent the world where A is true and perform forward inference in it more often than you do for the world where not-A is true."

This illustrates how algorithms from different cognitive modeling communities can be conceived of as different strategies for selecting the same basic set of operations.

Multiple implementation principle (MIP)

Each of the common functions can be implemented using multiple algorithms and representations. For example forward inference can conducted using at least these three mechanisms:

- *Production rule firing* can involve matching a set of rules against a set of known facts to infer new facts.
- *Feed-forward neural networks* take the facts represented by the activation of the input units, propagate these activations forward and output the facts represented by the values of the output units.
- *Memory*. The value of a slow-changing attribute at time t2 can be inferred by recalling its value at t1 when the interval between t1 and t2 is sufficiently brief.

This principle will help explain how lower-level cognitive and perceptual processes make up and influence higher-order reasoning and problem solving.

These insights motivate a computational framework that achieves a significant level of integration. High-order reasoning and problem solving algorithms can (thanks to the CFP) be integrated with each other by implementing each of them as sequences of common function executions. Executing these algorithms together in the same situation would simply require interleaving and or overlapping the sequences of common functions that make up each algorithm. Implementing common functions that make up higher-order reasoning and problem solving using multiple lower-order reasoning and problem-solving methods (thanks to the MIP) enables each step of lower-level computation to be integrated with and influence higher-level computation.

Architectural principles

We begin with the following hypothesis, enabled by the common function principle, and explore its consequences for mental architecture.

Higher-order cognition-common function hypothesis. The mind implements higher-order reasoning and problem-solving strategies by executing sequences of common functions.

The rest of this section describes how this hypothesis, together with what is empirically known about many cognitive processes, motivates a theory of human cognitive architecture that explains how the mind integrates many qualitatively different computational mechanisms.

First, there are many reasons (reviewed by Baars, 1988) to believe that the mind has specialized processors, which are here called **specialists**, for perceiving, representing and making inferences about various aspects of the world. The MIP implies that each function can be implemented using multiple mechanisms. The next hypothesis suggests that the mind implements common functions with these specialists.

Specialist-common function implementation hypothesis. The mind is made up of specialized processors that implement the common functions, using computational mechanisms that are different from specialist to specialist.

For example, the mind might have a place memory specialist that uses cognitive maps to keep track of the location of objects. It might also have a spatial relation specialist that uses a constraint mechanism to keep track of relations among mechanisms. The specialist common function hypothesis states that specialists such as these make forward inferences that constitute reasoning and problem solving, create subgoals, simulate alternate worlds and generate identity matches using their own specific computational mechanisms.

This theory takes no definite position regarding whether the computation within these specialists is "encapsulated" from that within other specialists, though it does imply that in general specialists communicate through a central mechanisms about to be described.

The CFP states that algorithms can be composed into sequences of common function executions. The MIP states that each function can be executed using multiple computational mechanisms. Assuming that the mind does execute higher-order inference algorithms using common functions, how many of its computational mechanisms (embodied in specialists) does it use at any moment? There are at least two reasons to believe the following answer to this question:

Integrative cognitive focus of attention hypothesis. The mind uses all specialists simultaneously to execute each common function and the mind has an integrative cognitive focus of attention which at once forces the specialists to execute a particular common function on the current focus, integrates the results of this computation and distributes these results to each of the specialists.

First, interference in the Stroop Effect (Stroop, 1935) between multiple kinds of processing (e.g., word and color recognition) suggests that multiple mental processes (i.e, specialists) engage sensory input simultaneously. That Stroop-like interference can be found with emotional, semantic and many other nonperceptual aspects of stimuli suggests that all, not just perceptual, specialists, simultaneously process the same information. Second, if interference in Stroop-like tasks is a result of the mind's attempt to integrate information from multiple cognitive processes, then it is possible that that the mechanism for achieving this integration is a focus of attention. Treisman and Gelad (1980) have demonstrated that visual attention is the main medium for integrating information from multiple perceptual modalities. Polyscheme is based on the notion that just as the perceptual Stroop effect can be generalized to higher-level nonperceptual cognition, that integrative perceptual attention suggests the existence of a higher-level cognitive focus of attention that is the mind's principle integrative mechanism. Whether the mind's perceptual and higher-order focus of attention are the same is left for now as an open question.

Since we are assuming that the mind implements higher-order reasoning and problem solving strategies by sequences of the individual functions specified in the CFP that are implemented by the computation that occurs during each focus of attention, the following hypothesis is implied:

Higher order cognition as attention selection hypothesis. The mind's mechanisms for choosing the cognitive focus of attention decide which higher-order reasoning and problem solving strategies it executes.

As will be describe below, these hypotheses collectively allow an explanation of how the mind integrates computational methods currently best modeled using very different computational mechanisms.

Polyscheme

Polyscheme embodies the theory of cognitive architecture outlined in the last section.

Formal preliminaries

Much of Polyshceme's operation involves communication among specialists that uses a simple, familiar propositional formalism. The actual details of the formalism itself are fairly arbitrary and not essential to the theory, though the existence of such a formalism is important.

The formalism consists of propositions and truth values for those propositions. A proposition denotes that a predicate holds over an ordered set of objects during a temporal interval in a "world". `R(x,y,t,w)` states that the relation denoted by `R` holds over the entities denoted by `x` and `y` over the temporal interval denoted be `t` in the world `w`. Worlds are entities that enable propositions about hypothetical worlds. For example, one can say that in the world, `w`, in which there is a hole in a cup, that the cup is now empty with the proposition, `Empty(cup,now,w)`. Although propositions may seem cognitively implausible, it is possible to map propositions onto more the more familiar ACT-R chunk representation. For example, the most recent proposition could be reformulated as the following chunk: `(c ISA empty-predicate object cup time now world w)`.

A specialist can indicate its belief or doubt in a proposition, called its *stance* on it, with a two-dimensional truth value. The first dimension is the degree of confidence it had in the proposition's truth and the second is the degree of confidence that it is false. The four confidence levels are `C`, `L`, `l`, `m` and indicate respectively that the truth or falsehood is certain, very likely, likely, or maybe the case. "`?`" indicates that there is no evidence either way. Thus, to say that a specialist takes the stance, `(L,m)`, on a proposition, `P`, is to say that the specialist has evidence that `P` is very likely to be true and evidence that it might be false. The confidence levels do not correspond to numerical probabilities; only their relative order of confidence matters. This two-dimensional scheme enables Polyscheme to differentiate between the case where there is some evidence that `X` is true, say, `(l,?)`, and the case where there is very good evidence that it is both true and false, e.g., `(L,L)`. Both could be denoted with `P(X) = `.5 in a scalar probabilistic framework and would not capture the difference between the first situation of lukewarm confidence and the second, which indicates the potential of serious inconstancy. The truth value which represents the overall evidence all the specialists have about the truth of a proposition is called their *consensus* on that proposition.

Specialists

All specialists implement the following functions:

- `OpinonOn(Proposition)`. Specialists specify their stance on a `Proposition`.
- `ReportOpinions(Proposition,Opinions)` Specialists learn about the opinion of other specialists on a proposition.
- `PossibleMatches(Proposition)`. For each entity in the proposition, specialists find entities that might be identical to it.
- `RequestAttractions()`. Specialists return a set of "attractions" (described later in this section) which specify propositions they would like to focus on.

Specialists implement the common functions thus: thus:

- *Forward Inference*. When, from new information, a specialist is able to infer that a proposition, `P`, has truth value, `TV`, its `RequestAttractions` function will request attention for `P`. `OpinionOn(P)` will return the opinion that P has truth value, `TV`.
- *Subgoaling*. When a specialist can give information about the truth value of `P` if it knows the truth value of a set of propositions, `S`, it will include those in the set of propositions it requests through `RequestAttractions`.
- *Identity matching*. Identity hypotheses are generated by `PossibleMatches`.
- *Representing alternate world*. Since the propositions that are the input and output of these functions can regard alternate worlds, the specialists must be able to represent and make inferences about these possible worlds.

Focus of attention.

Polyscheme embodies the integrative cognitive focus of attention hypothesis by including a focus of attention in the form of a proposition that all specialists focus on, store in memory and make inferences about simultaneously. At every time step, Polyscheme does the following:

1. Polyscheme's *focus manager* (described below) chooses a proposition to make the focus of attention.
2. Polyscheme collects the stances of the specialists on the proposition by calling their `OpinionOn` functions.
3. Polyscheme reports these stances to the other specialist using their `ReportOpinions` functions.
4. Using their own computational mechanisms, the specialists process this new information, make inferences and decide which propositions they would help them make better information.
5. The focus manager collects the propositions specialists request to focus on by calling the specialists' `GetAttractions` functions.

Focus Manager

Polyscheme models the guiding of attention using two kinds of mechanisms. First, each specialist can indicate through its `GetAttractions` a set of *attractions* which each indicate a proposition it requests to focus on and a *charge function* that indicates for which times and how strongly this request is made. Second, Polyscheme has a *focus manager* which at each time step uses the attractions to focus on a proposition.

Attractions can be conceived of as tuples `(p,c)`, where `p` is a proposition to focus on and `c` is a charge function which indicates how strongly a proposition is to be focused on at a given time. For example, the charge function $c = 1/e^t$, indicates that the desire to focus on `p` is immediate and fades over time.

The focus manager chooses the focus of attention at each time step using a two-step process:

1. It computes the charge of all the attractions.
2. It chooses the proposition, `P`, from the attraction with the highest charge.

Modeling qualitatively different reasoning strategies in Polyscheme

It is now possible to illustrate how Polyscheme enables cognitive models based on different computational mechanisms, in this case backtracking search, production rules and Bayesian networks, to be implemented such that they can be combined into one cognitive model.

Backtracking search has been used widely to model reasoning and planning in cognitive science. It can be implemented easily in Polyscheme by adding an *uncertainty specialist* that behaves thus:

- During time steps where the specialists' consensus on `P` includes evidence it is true and evidence that it is false, `GetAttractions` includes the following two attractions in the set of attractions it returns:
- `(P1,c1)`, where `P1` is the (trivially true) proposition that P is true in the world where `P` is true.
- `(P2,c2)`, where `P2` is the proposition (trivially false) that `P` is true in the world where `P` is false.

In other words, when there is conflicting evidence on proposition, `P`, imagine the world where `P` is true and the world where `P` is false. If, when imagining the world where `P` is true, there is conflicting evidence on `Q`, then imagine the world where `P` and `Q` are true and the world where `P` is true and `Q` is false, etc. Should the specialists infer a contradiction in one of these worlds, no further search will be made in this direction. This is precisely how backtracking search is conducted. Variations in charge functions `c1` and `c2` determine which particular flavor of backtracking search.

The benefit of implementing search thus is that:

- Each step in the search can be evaluated using the cognitive resources of all the specialists.
- Inference methods (e.g., production rules, backward chaining, case-based reasoning, etc.) can interrupt by taking over focus at each step to elaborate the particular world being searched at the moment.
- Perceptual and memory processes influence every step of the search, by providing information helpful for elaborating a particular imagined world.

Production rules can be implemented in Polyscheme by adding a *rule specialist* that contains a collection of production rules. Describing different ways of implementing this specialist will show how Polyscheme enables features of the ACT-R and SOAR (Laird, et al, 1987) production systems to be implemented together in one cognitive model and demonstrates how doing so with Polyscheme makes it easier to model the integration of production rule firing with other cognitive processes.

- The rule specialist contains a set of production rules, `R`.
- When Polyscheme focuses on a new proposition `P`, the rule specialist matches all rules which match it.
- If more than one rule matches, either
    - All can be fired.
    - Or some internal conflict resolution mechanisms can pick one (e.g., the production matching conflict resolution system in ACT-R).
- If a proposition in the right hand side of a rule match is an open proposition, O, (i.e., one that has an unbound variable) during a time step, `GetAttractions` includes an attraction for `O` in the attractions it returns during that time step.
- When Polyscheme focuses on `O`, the specialists' `PossibleMatches` functions will find propositions which provide bindings for the open variables in `O`.
    - To include ACT-R's conflict resolution mechanisms for retrieving chunks would only require adding a specialist that implements this functionality in its `PossibleMatches` function to choose from multiple possible matches.
    - To treat the problem of more than one binding for a proposition as a SOAR impasse would only require making a specialist which makes a possible world (i.e., an to a analogue state in SOAR) based on each possible match and requests these to be focused on using its `RequestAttractions` method.

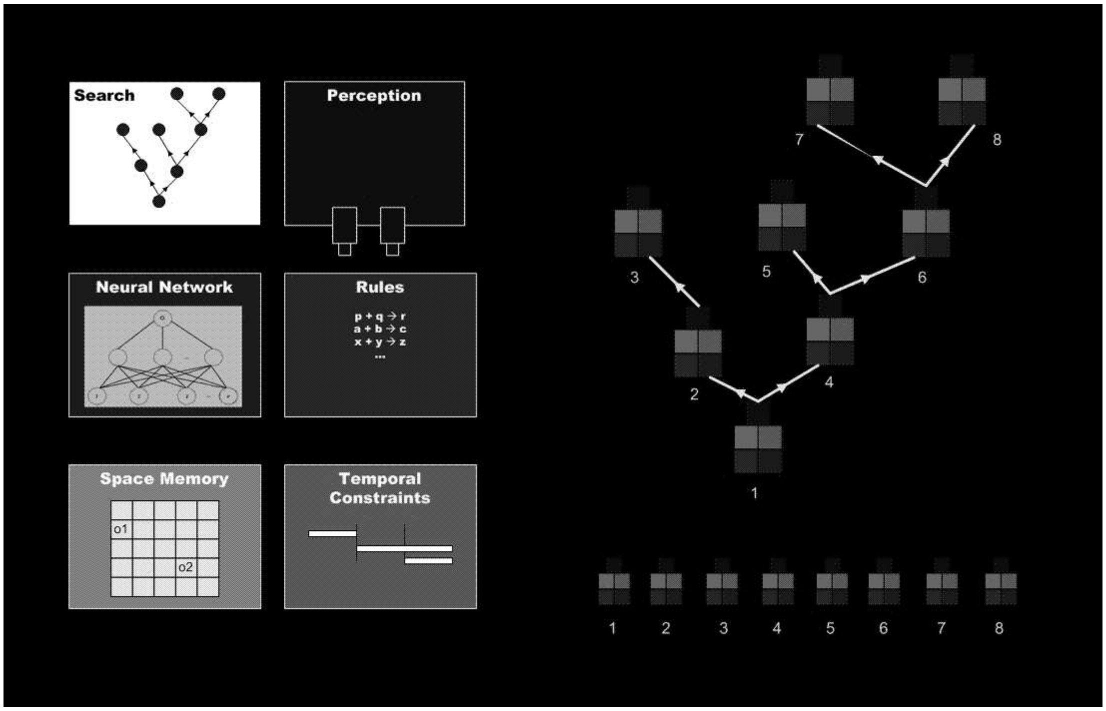

Figure 1. In purely modular cognitive models (illustrated on the left) integration is an explicitly provided for exception, while in Polyscheme models (illustrated on the right), the integration of multiple computational methods and algorithms is constant and automatic.

Bayes Network Propagation can be implemented in Polyscheme using *stochastic simulation*. Since precisely propagating probabilities in Bayesian Networks can be too computationally expensive in most practical situations, many cognitive models based on Bayesian Networks use a *stochastic simulation* algorithm to propagate these. They conduct multiple *simulations* by guessing (based on prior and conditional probabilities encoded in the network) at the value of variables. The probability of a variable value can be estimated by the proportion of simulations in which the variable is set to that value. Stochastic simulation can be implemented in Polyscheme easily, with the added benefit the every simulation can be conducted using all cognitive the cognitive and perceptual processes included in its specialists. Stochastic simulation can be implemented thus:

- When there is evidence that `P` in world `w` is N times more likely than `not-P`, the `GetAttractions` function of the uncertainty specialist includes in its return set N times more attractions of the form `A1` than it does `A2`:
  - `A1 = (P1,c1)`, where `P1` states that `P` is true in the world `w` modified by the assumption that P is true.
  - `A2 = (P2,c2)`, where `P2` states that `P` is false in the world `w` modified by the assumption that P is false.
- The uncertainty specialist keeps track of the truth value of `P` in all the worlds that have not lead to contradictions. It determines the truth value of `P` by computing the ratio of the number of worlds in which `P` is true vs. the number in which it is false. The higher the ratio, the more confidence its `OpinionOn` method assigns to `P`.

How Polyscheme explains the integration of multiple cognitive processes

Having described how models based on different computational mechanisms, e.g,. search, production rules and Bayesian networks, can be implemented in Polyscheme, it is now possible to explain how doing so explains how the mind integrates these mechanisms deeply in particular .

Integrating different reasoning and problem solving strategies. Polyscheme models the mind's execution of reasoning strategies as a sequence of foci and explains the integration of these strategies by the ability of sequences of foci from different strategies to be easily interleaved. For example, imagine a task requiring production rule firing and Bayesian network propogation and involves foci F1, F2, … , F11. Say production firing is implemented by focus F1, F3, F6, F7, F8, F9 and Bayesian network propagation through F2, F3, F4, F5, F6, F7, F10, F11. Notice that the execution of both algorithms is *interleaved* so that inference made in the middle of, say, production rule matching can be used immediately in network propagation. Notice also that F3 and F6 are *shared* by both algorithms. This shows how Polyscheme enables models of *opportunism* by allowing computation involved in one strategy to be incorporated into another if the opportunity exists. Thus, Polyscheme explains how the mind flexibly combines reasoning and problem solving strategies by executing them as sequences of foci that can be interleaved and shared with each other.

Integrating models of reasoning and problem solving with lower-level cognitive processes. If the mind implements reasoning and problem solving strategies using sequences of foci, all the computation is performed by the computation of the specialists during each focus. In other words, according to the Polyscheme theory, much higher-order cognition is nothing more then the guided focus of lower-level cognitive and perceptual processes. This helps explain how symbolic and serial cognitive processes are grounded (in the sense of (Harnad 1990)) in lower-level processes and to the extent these lower-level mechanism are sensorimotor, constitutes and embodied theory of higher-order reasoning. Also, since every focus of attention can be influenced by memory, perceptual and sensorimotor mechanisms, Polyscheme explains how reasoning can be interrupted or guided by these. Figure 1 illustrates the difference between integration in Polyscheme and more strictly modular approaches to integration. On the left of a figure is a hypothetical cognitive model that includes models based on particular algorithms and data structures. Communication between these modules must be explicitly provided for. However, on the right, the execution of backtracking search as the focus of attention of each module simultaneously illustrates how Polyscheme explains how the mind integrates multiple cognitive processes in every step of reasoning continually and automatically.

Existing Polyscheme models

Polyscheme was initially developed to build a model of infant physical reasoning (Cassimatis, 2002) that combined neural networks (for object recognition and classification), production rules (for causal inference), constraint propagation (for keeping track of temporal and spatial constraints), cognitive maps (for object location memory) and search (for finding plausible models of unseen events and for finding continuous paths). This model demonstrates how even apparently simple physical cognition can require sophisticated reasoning which could be modeled as the guided focus of attention of cognitive and perceptual processes empirically known to exist in infants. This model was adapted to construct a model of syntactic understanding (Cassimatis, 2004) and models of human-robot interaction (Cassimatis, et al. 2004). The model of human-robot interaction demonstrated that implementing every step of human reasoning as a focus explains how, for example, perceptual information, spatial cognition and social reasoning could be continually integrated during language use. For example, one model enabled human nominal references to be instantly resolved using information about the speaker's spatial perspective by implementing this linguistic process as a focus of attention on the location of an object that the models' spatial perspective specialist could refine. Thus, by implementing a language understanding algorithm, not as a process encapsulated in a module, but as a sequence of foci, every step of that algorithm could be refined by perception, providing a model of that same phenomenon in human dialog.

Conclusions

This paper began with the problem of combining models of cognitive processes based on difficult-to-integrate computational methods into a unified understanding of how the mind works. The computational function and multiple implementation principles, together with some empirical knowledge of how the mind works suggested a series of hypotheses about human cognitive architecture that together explain how the mind can integrate these different computational mechanisms. These principles have been embodied in a computational cognitive architecture, called Polyscheme. How accurately and comprehensively this theory explains human mental organization can only be determined by actually integrating cognitive models from various frameworks in Polyscheme and showing how this enables models of cognition in situations that are accurate and are difficult or impossible using individual frameworks alone. This paper's demonstration that models from three formerly difficult-to-integrates computational frameworks can be construed as an encouraging sign that this research program can achieve some success and insight.

References

J. R. Anderson and C. Lebiere (1988). *The Atomic Components of Thought*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1998.

B. J. Baars (1988). *A Cognitive Theory of Consciousness*. Cambridge: Cambridge University.

N. L. Cassimatis (2002). *Polyscheme: A Cognitive Architecture for Integrating Multiple Representation and Inference Schemes.* Doctoral Dissertation, Media Laboratory, Massachusetts Institute of Technology, Cambridge , MA .

N. L. Cassimatis (2004). Grammatical Processing Using the Mechanisms of Physical Inferences. In *Proceedings of the Twentieth-Sixth Annual Conference of the Cognitive Science Society*.

N. L. Cassimatis, J. Trafton, M. Bugajska, A. Schultz (2004). Integrating Cognition, Perception and Action through Mental Simulation in Robots. *Journal of Robotics and Autonomous Systems*. Volume 49, Issues 1-2, 30 November 2004, Pages 13-23.

E. Charniak (1991). Bayesian networks without tears. AI Magazine 12(4): 50-63.

S. Harnad (1990). The Symbol Grounding Problem. *Physica* D 42: 335-346.

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

E. S. Spelke (1990). Principles of Object Perception. *Cognitive Science* 14: 29-56.

J.R. Stroop (1935). Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*, 18, 622-643.

A. M. Treisman & G. Gelade (1980). A feature integration theory of attention. *Cognitive Psychology, 12*, 97-136.