

# A Cognitive Substrate for Achieving Human-Level Intelligence

*Nicholas L. Cassimatis*

■ Making progress toward human-level artificial intelligence often seems to require a large number of difficult-to-integrate computational methods and enormous amounts of knowledge about the world. This article provides evidence from linguistics, cognitive psychology, and neuroscience for the *cognitive substrate hypothesis* that a relatively small set of properly integrated data structures and algorithms can underlie the whole range of cognition required for human-level intelligence. Some computational principles (embodied in the Polyscheme cognitive architecture) are proposed to solve the integration problems involved in implementing such a substrate. A natural language syntactic parser that uses only the mechanisms of an infant physical reasoning model developed in Polyscheme demonstrates that a single cognitive substrate can underlie intelligent systems in superficially very dissimilar domains. This work suggests that identifying and implementing a cognitive substrate will accelerate progress toward human-level artificial intelligence.

## The Profusion Problem

A major challenge to achieving human-level artificial intelligence is the apparent enormity of the problem. Creating an effective intelligent system today generally requires severe constraints either on its func-

tionality or robustness. This article describes a research program based on the premise that a major obstacle to intelligent systems that are at once more broadly functional and more robust is the profusion of knowledge, data structures, and algorithms that must be integrated into a system to achieve this goal. I call this the *profusion problem*.

Enormous amounts of knowledge are required in even very simple domains. For example, imagine a natural language question-answering system that answers simple queries about football schedules. Queries such as “When is the first Pittsburgh Steelers game after (the World Series, Thanksgiving, my daughter’s birthday, the next full moon)?” require knowledge and information about baseball, national holidays, personal information about the user and his family, celestial movements, and so on. A substantial fraction of human knowledge could potentially be relevant to such queries, and it is often difficult to restrict the domain in such a way that neatly rules out most of the relevant information. One indication of the enormity of the knowledge profusion problem is that while the very large compilations of assertions in knowledge bases such as Cyc (Lenat and Guha 1990) or ThoughtTreasure (Mueller 1998) have been quite important and useful, they have yet to provide a comprehensive store of all the knowledge that might be relevant to such queries.

Corresponding to the knowledge profusion problem is the procedural profusion problem: most domains involve computational problems that are best dealt with using a wide variety of difficult-to-integrate computational methods. For example, when a speaker makes an utterance, a natural language spoken dialogue system must convert a continuous acoustic signal into a discrete representation of the phonemes, morphemes, and words. It must (at least partially) identify the syntactic structure of the utterance. It must use knowledge about the world, including the beliefs and intentions of conversants, to infer the speaker's intention. It must take each of these aspects of language and world knowledge into account to plan a response that will achieve or advance the goals of the dialogue. Each of these aspects of dialogue is currently best dealt with using different classes of algorithms. For example, hidden Markov models are often used for speech recognition; chart- and search-based algorithms are used for syntactic parsing; logical, case-based, and probabilistic methods are used for reasoning about the world and other people's intentions. Since evidence (Tanenhaus and Trueswell 1995) shows that each of these aspects of language use constrains the others, designing a system that is capable of human-level natural language dialogue involves the difficult problem of tightly integrating these algorithms (based on very different control and data structures) into one system. All this work is required for dialogue, only a single aspect of human-level intelligence. The number of computational methods required and the difficulty of the integration problem they pose would seem only to multiply for systems that integrate more of human-level intelligence.

If achieving human-level AI requires a store of knowledge and collection of algorithms that is not practical to compile or implement manually, perhaps computers could be programmed to automatically acquire these. The first problem with this approach is that most machine-learning algorithms target and are built around one or a few specific computational formalisms. For example, using neural network connection update algorithms commits one to neural networks as an execution algorithm. Thus, learning methods can confine research to a relatively narrow subset of algorithms. There are techniques for combining existing learning methods and learning to delegate between them, (Cox and Ram 1999, Jordan and Jacobs 2002), but these do not create new algorithms altogether; they only learn to delegate among existing algorithms. Thus, existing machine-learning methods by them-

selves are not sufficient to solve the procedural profusion problem.

Even if there were a way of automatically accumulating the knowledge and algorithms required for human-level intelligence, a problem of effectively integrating these into one system remains. Merely encapsulating this machinery into modules that pass messages among each other does not allow for the internal operation of one algorithm to be influenced by another. For example, the choice of an action to explore in a motion planning algorithm should be able to be influenced by computation performed by Bayesian networks, partially observable Markov decision processes (POMDPs), case-based reasoners, or logic theorem provers. Yet, the control and data structures of these different classes of algorithms are very difficult to integrate and therefore it is difficult to construct a computational architecture in which the operation of one algorithm can be influenced by that of many others.

Thus, the profusion problem makes achieving human-level intelligence not merely a problem of marshalling the time and resources necessary to accumulate the required set of knowledge and algorithms, but also a genuinely difficult integration problem.

## The Cognitive Substrate Hypothesis

The generality of human cognition motivates a solution to the profusion problem. Many of the domains in which human cognition is effective involve technology, cultural practices, and language that did not exist when the human brain evolved to its present state. The pace of evolution is so slow that whatever cognitive mechanisms humans use when thinking about, say, airplanes, computers, parliaments, and equity markets are the same mechanisms they used in earlier cognition. This suggests that this set of earlier mechanisms (which presumably initially evolved in part for dealing with the physical world and for coordinating social relationships), in addition to the mechanisms that adapt them for use to new domains, are sufficient to achieve human-level intelligence in all domains. This article explores a strong version of this line of thought, called the *cognitive substrate hypothesis*. The hypothesis states that there is a relatively small set of computational problems such that once the problems of artificial intelligence are solved for these, that is to say, once a machine, called here a "cognitive substrate," is created that effectively solves these problems, then the rest of human-level intelligence can be achieved by the relatively

simpler problem of adapting the cognitive substrate to solve other problems.

To understand what follows, it will be helpful to begin with a very brief example of what a cognitive substrate might be. As explained later, an initial guess at a cognitive substrate can be derived from the computational problems that need to be solved for basic social and physical reasoning. These include reasoning about temporal intervals, causal relations, identities between objects and events, ontologies, beliefs, and desires. The cognitive substrate hypothesis suggests that once a set of computational mechanisms, in other words, a cognitive substrate, that solves the problems of human-level AI for this set of problems is constructed, achieving the rest of human-level AI will be a relatively easy problem.

The cognitive substrate hypothesis being true would benefit work in human-level AI in three ways:

*Smaller problem.* Progress toward human-level intelligence would not halt until or be dependant on the identification of enormous amounts of commonsense knowledge and a seemingly endless number of algorithms and their combination into one system. Progress on the comparatively small (but not trivial) set of problems required to implement a cognitive substrate would constitute progress toward human-level intelligence in all domains.

*Quicker intelligent system development.* Developing intelligent systems for new domains becomes much easier, since they can be based on the same mechanisms used in other domains.

*Easier integration across domains.* Two systems designed for different domains are easier to integrate when they are based on the same set of mechanisms.

The cognitive substrate hypothesis is important for AI research even if it is not universally true. If, for example, a cognitive substrate could underlie cognition in most, but not all, domains of human cognition, then, much, but not all, of the problem of human-level cognition becomes more tractable, intelligent system development will be accelerated in most, but not all, domains, and integration would be eased across most, but not all, domains.

Several lines of research in artificial intelligence, cognitive psychology, linguistics, and neuroscience support the cognitive substrate hypothesis and motivate a first guess at what would constitute such a substrate.

### Implicit Substrate Hypothesis in Much Artificial Intelligence Research

First, for most any major class of computational method in AI, there have been researchers

who have believed that most or all AI problems could be solved by that method. Examples include characterizing AI problems as search through a state space (Newell and Simon 1972) or updating probabilities in a Bayesian network (Pearl 1988). That each such method has been applied with success in many domains suggests that the superficial dissimilarity among those domains hides deeper similarities.

In addition to an implicit version of a cognitive substrate hypothesis embodied in these subfields of artificial intelligence, there have been attempts in the history of the field to reduce this superficial variety to a small number of primitives. Schank (1975), for example, identified a modest set of primitives that he proposed could represent much of the semantics of human language. This research program has not achieved human-level AI, not because the cognitive substrate hypothesis is incorrect, but because the reasoning problems associated with these primitives have never been solved. For example, Schank's primitives (and those from other research programs (for example, Jackendoff [1990]) involve causality and space, but human-level spatial and causal reasoners do not yet exist. It is not enough, therefore, to find a set of primitives that can represent all human knowledge. One must also solve the computational problems associated with these primitives that humans can solve.

Thus, while the ability of single AI methods to support reasoning in a very wide variety of domains supports the cognitive substrate hypothesis, the research presented here is based on the hypothesis that a fully successful implementation of a substrate will require that the benefits of each specific class of AI methods must somehow be integrated into one system (Minsky 1986).

### Linguistic Semantics

Research in linguistic semantics has found that the structures used to represent the semantics of a relatively small set of semantic fields (such as physical motion and causation) can be used to represent the semantics of many other semantic fields. Jackendoff's treatment of motion transfer verbs illustrates this point (Jackendoff 1990).

Jackendoff introduces a set of primitives to explain semantic regularities that occur in multiple, not-necessarily-physical domains. These primitives include *cause*, *go*, *path*, *to*, and *from* and are common in many other frameworks.

For example, the meanings of "John entered the room" and "John left the room" can be formalized:

GO (John, [*path* [to: room]])

“John moved along a path that ended in the room.”

GO (John, [*path* [from: room]])

“John moved along a path that originated in the room.”

With the same primitive notions, one can represent a change of state:

GO<sub>state</sub> (John, [*path* [to: drunk]]) “John became drunk.”

GO<sub>state</sub> (John, [*path* [from: drunk]]) “John sobered.”

Or, temporal extent:

GO<sub>time</sub> (class, [*path* [to: 9pm]]) “Class extended to 9pm.”

GO<sub>time</sub> (class, [*path* [from: 9pm]]) “Class began at 9pm.”

Or, transfer of possession:

GO<sub>possession</sub> (\$100, [*path* [to: John]]) “John received \$100.”

GO<sub>possession</sub> (\$100, [*path* [from: John]]) “John lost \$100.”

The only difference in each example is GO's subscript, representing the (potentially abstract) domain in which such motion occurs. Jackendoff's work describes the semantics of a large set of word classes with only a few more primitives, each of which are part of the domain of commonsense physical reasoning.

This work suggests underlying similarities exist among the mechanisms the human mind uses to think in multiple domains, thus giving support to the notion that a relatively small set of mechanisms can lead to human-level intelligence in all domains. That many of the primitives enable so many semantic fields to be represented and have origins in representations of the physical world suggests that the cognitive structures humans use in physical reasoning may be a place to look for a first guess at the contents of a cognitive substrate. Finally, this work is also consistent with the evolutionary motivation for the cognitive substrate hypothesis since dealing with the physical world has been an important concern for human and nonhuman primates throughout their evolutionary development.

### Cognitive Psychology and Neuroscience

Research in cognitive psychology and neuroscience provides evidence that much nonspatial or physical thought involves mechanisms that the brain normally uses for spatial and physical cognition. This evidence supports the cognitive substrate hypothesis that a relatively

small set of computational mechanisms can underlie a wide variety of cognition.

Cognitive psychologists have found (see Barsalou et al. [2003] for a review) that humans consistently map certain visual and motor representations onto abstract, nonphysical concepts. Further, when people engage in tasks with visual and motor components, they think more or less easily about the abstract concepts involved to the extent these are more or less consistent with the visual-motor aspects of the task. For example, D. C. Richardson, M. J. Spivey, S. Edelman, and A. D. Naples (Richardson et al. 2001) found that people consistently associate an image of people situated horizontally with the verb “push” and an image with people situated vertically with the verb “respect.” One explanation of this is that people associate *X* respecting *Y* with *X* looking up at *Y*, whereas most pushing is along a horizontal plane. Further, D. C. Richardson, M. J. Spivey, L. W. Barsalou, and K. McRae (Richardson et al. 2003) found that people had to work harder to understand sentences that involved verbs associated with a vertical orientation (such as “respect”) when viewing images that had objects in a horizontal orientation. That cognition with abstract concepts is so affected by the visual configuration of objects in the environment suggests that abstract conceptual cognition involves visual and motor representations. Further, these conclusions have been confirmed by repeated cases (see, for example, Warrington and Shallice [1984], Humphreys and Forde [2001], and Cree and McRae [2003]) of visual and motor regions of the brain becoming active during nonperceptual cognition.

### Conclusions from This Evidence

All this work on the evolution and functioning of human cognition both supports the cognitive substrate hypothesis and provides some first steps at identifying the contents of the substrate. The work in evolution, linguistics, psychology, and neuroscience suggests that the mechanisms the human mind uses to think about a relatively small set of domains underlies cognition in many other domains. This fact about human cognition, together with the underlying similarities AI researchers have already found among domains, suggests that there is a relatively small subset of domains such that if researchers solved the problems of human-level AI for them, then achieving the rest of human-level AI will be a relatively easy problem.

Also, because so much of the work in the cognitive sciences suggests that the mechanisms of physical cognition underlie cognition in many domains, these mechanisms can form

the basis of a first guess at what is in the cognitive substrate. Through work in cognitive models of physical reasoning (Cassimatis 2002) and some preliminary work extending this elsewhere (Cassimatis et al. 2004, Cassimatis 2004), I have arrived at a preliminary guess at what can constitute a cognitive substrate. This includes reasoning about time, space, part-whole, categories, causation, uncertainty, belief, and desire. This is just a preliminary list. Part of the research program described here involves refining the understanding of what can constitute a cognitive substrate.

### Implied Research Program

The preceding line of reasoning suggests the following research program: (1) identify and implement a cognitive substrate; (2) find mappings from multiple domains onto the cognitive substrate; and (3) automate the process of adapting a cognitive substrate so that it can solve problems in other domains.

The stages of this research program of course can and should be executed in parallel. For example, the process of mapping a cognitive substrate to many domains will suggest and constrain elements of that substrate, and the process of automating the adaptation of a cognitive substrate to many domains could constrain how that substrate is implemented.

A preliminary guess at the contents of a cognitive substrate have already been provided. The following section describes some of the problems involved in implementing a cognitive substrate and a framework for addressing them. Next, a mapping between the problems of syntactic parsing and physical reasoning will be described that demonstrates how intelligent systems in superficially very different domains can be implemented using the same substrate and how this makes constructing intelligent systems in new domains faster. The problem of automating mappings is briefly discussed, but for now, we are assuming that much existing work in analogy will be a major part of the solution.

## Implementing a Cognitive Substrate

As discussed previously, the procedural profusion problem entails the problem of integrating computational methods with very different control and data structures that are difficult to combine into one system. Recognizing that a cognitive substrate can underlie inference in many domains does not completely resolve this issue because the computational problems that a substrate must solve are themselves best

dealt with using very different computational techniques. For example, the first guess at the contents of a cognitive substrate involve reasoning about time and the beliefs and desires of other agents. Yet, the best temporal reasoners are often based on different computational methods than the best social reasoners. Most frameworks for integrating multiple computational methods either attempt to reduce them to one computational formalism or encapsulate them in modules. Both approaches have achieved some success but have limitations. This implementation of the substrate described in this article is motivated by the assessment that the reductive approaches can never fully escape the limitations of the computational method they are based on and that the modular approach provides too loose an interaction among the different methods. This section outlines an approach to integration, embodied in the Polyscheme cognitive architecture, which enables multiple computational methods to be implemented such that the interaction between them is much more ubiquitous. Finally, I describe some work with Polyscheme at Rensselaer's Human-Level Intelligence Laboratory and the Naval Research Laboratory that demonstrates how these principles enable an implementation of a cognitive substrate that greatly simplifies the procedural profusion problem.

Polyscheme (Cassimatis 2005) is based on two principles that enable very different computational methods to be integrated into one system. The *common function principle* (CFP) states that many AI algorithms can be implemented in terms of the same basic set of common functions, and the *multiple implementation principle* (MIP) states that each common function can be implemented using multiple computational methods. These computational principles enable the level of integration among the various data structures and algorithms needed to implement a cognitive substrate.

The following is the current best guess at the set of common functions:

*Forward inference.* Given a set of beliefs, infer other beliefs that 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 that 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 that might be identical to it.

One way to justify the CFP is to show how these common functions can implement a wide variety of algorithms. For example, backtracking search and stochastic simulation (often used in Bayesian network inference) can be roughly characterized using the same set of common functions (which are underlined below).

*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.* “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.”

One way to illustrate the MIP is to show how multiple computational mechanisms can implement forward inference:

*Neural Networks.* The activation of input units of a feedforward neural network leads to a change in the activation of the output units of the network. These activations represent facts that can be inferred from the facts represented by the input units.

*Forward rule changing.* Production systems can be constructed to match the left-hand sides of production rules against a set of currently known facts to infer new facts represented by the right hand sides of rules.

*Ontologies.* When an object, *o*, is a member of category *C* in a category hierarchy, one can infer that *o* is a member of *C1 ... Cn*, the ancestors of *C*.

Examples for other common functions are provided in Cassimatis (2005). The upshot of all these examples is that each of the common functions can be implemented by many different computational techniques and that the apparent diversity of many computational methods obscures the fact that they are each solving the same small set of computational problems in their own way.

The CFP and the MIP motivate a cognitive architecture, Polyscheme, that enables an advance in the level of integration of computational methods based on different control and data structures and hence also an implementation of a cognitive substrate. Polyscheme is not intended to supplant existing cognitive architectures and computational frameworks. Instead, it was designed to provide a framework for integrating the best features of many existing architectures and frameworks that have heretofore not been integrated. For example, Polyscheme enables production rules to be integrated with neural networks and stochastic

simulation algorithms for probabilistic inference into a single intelligent system. The integration of these various algorithms is tighter in Polyscheme than in conventional modular, multiagent systems because these algorithms are implemented in such a way that every step of the execution of every algorithm can be influenced by multiple other algorithms and data structures.

A Polyscheme system consists of a set of modules, called specialists, a focus of attention for fusing the results of the specialists' inferences together, and a set of attention-control strategies that shape the flow of computation. Each of these features is motivated in detail elsewhere (Cassimatis 2005), but the following is a summary of Polyscheme.

Each specialist in Polyscheme is based on its own specialized computational mechanisms and implements each of the common functions. For example, an object-recognition specialist can be based on a neural network and implement the common functions thus:

*Forward inference.* The input units to the network represent the features of an object and the output units represent the category of an object. Forward inference happens when the propagation of the activation of the input units leads to new values of the output units.

*Subgoalting.* When asked for propositions whose truth value might help determine the category of an object, the specialist returns propositions representing values of the input units.

*Identity Matching.* When, for object *O*, the specialist is asked for other objects that might be identical to *O*, it determines *O*'s category, *C*, through forward propagation, and returns all objects it has classified as belonging to category *C* in the past.

*Representing counterfactual worlds.* The specialist makes inferences about a counterfactual world in which an object has certain features by setting the input units of its network to those features even though the object does not necessarily have those features in reality.

Specialists in Polyscheme communicate through a *focus of attention*. Polyscheme includes a propositional language that all specialists use to communicate with each other. Although they represent knowledge and make inferences using their own specific data structures, specialists must be able to translate back and forth into this language in order to inform other specialists of their inferences and receive knowledge from other specialists. The focus of attention in Polyscheme is a proposition, and at every time step specialists focus on the same proposition. They do so in order to learn about the beliefs of other specialists about a proposi-

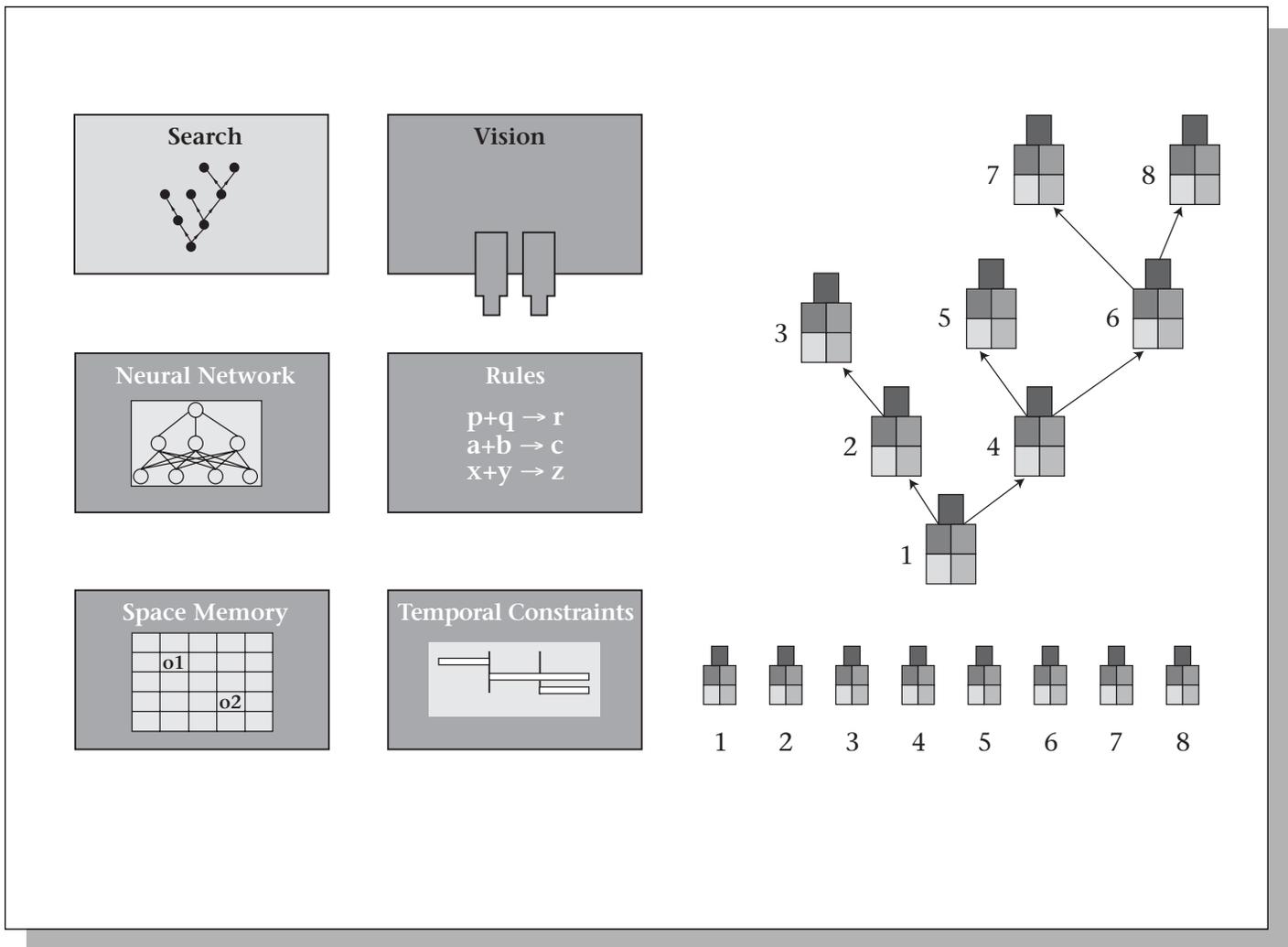


Figure 1. The Contrast between Modular Integration and Integration in Polyscheme.

tion to make sure they do not proceed with certain assumptions about a proposition when another specialist might have reason to contradict them. Cassimatis (2005) presents several forms of evidence that suggest humans have a similar focus of attention that is more general than merely visual attention.

Through various attention control strategies (described in Cassimatis [2002] and Cassimatis et al. [2004]) Polyscheme implements reasoning and planning algorithms such as search, stochastic simulation, means-ends analysis, and logical deduction, through a sequence of attention fixation. The execution of any algorithm in Polyscheme is thus a set of attention fixations. This enables Polyscheme to achieve integration in two ways.

First, since very different algorithms can all be reformulated in terms of sequences of atten-

tion fixations, integrating these algorithms is as easy as interleaving and combining the sequences that constitute their execution. For example, imagine a task requiring production rule firing and Bayesian network propagation. Suppose Polyscheme's sequence of attention fixation in this task involves foci F1, F2, ..., F11. Say production firing is implemented by foci F1, F3, F6, F7, F8, and F9 and Bayesian network propagation through F2, F3, F4, F5, F6, F7, F10, and F11. Notice that the execution of both algorithms is *interleaved* so that an inference made in the middle of, say, production rule matching can be used immediately in network propagation.

Second, since every step (that is, attention fixation) of these algorithms involves all the specialists (by focusing their attention on the same proposition), reasoning and planning are

constantly integrated with perception and knowledge in multiple forms of representation. For example, if a logic theorem proving algorithm makes a subgoal of finding whether  $P$  is true, a perceptual specialist attached to a video camera or a specialist encapsulating a relational database can assert or deny  $P$ 's truth. Thus, every step of reasoning and planning in Polyscheme is integrated with a wide array of information and computation. This increased level of integration is illustrated in figure 1. The left side of the figure depicts a set of modules based on different representations and algorithms within a traditional modular system. Backtracking search is just one method encapsulated inside a module. The algorithms in the modules execute in parallel, communicating on occasion. The right side of the figure depicts backtracking search in Polyscheme. Backtracking search is implemented as sequences of attention fixations, each of which involves the representations and data structures in each of the specialists. In this way, every step of search is automatically and continually integrated with multiple data structures and algorithms.

Polyscheme's ability to integrate multiple forms of reasoning and planning with perception and diverse sources of information has been demonstrated in a robotic framework (Cassimatis et al. 2004) for addressing the tension between the apparent rigidity and inflexibility of the sense-plan-act loop of traditional planning algorithms and the autonomy and reactivity required of real-world robots. This framework implemented a search-based motion planner, a truth maintenance system, and a physical world simulator as attention control strategies. Since every step of these algorithms was an attention fixation that involved in part information from a video camera, reasoning and problem solving were constantly informed by perceived changes in or new information about the robot's environment. The quickness of each attention fixation and its close connection with the environment gave the robot the robustness of reactive control systems. At the same time, the ability of the attention fixations to implement algorithms combined the ability to reason and formulate plans while preserving the robustness normally associated with purely reactive systems.

The integration demonstrated in this work is enabled by the recognition that inference algorithms originally based on very different computational formalisms can be executed as sequences of a small set of common functions (according to the CFP) that can be easily interleaved and that these common functions can be implemented using many different algo-

gorithms (according to the MIP) that can thus potentially contribute to every step of every inference. As work with Polyscheme illustrates, these approaches enable great progress in solving the integration problems associated with implementing a cognitive substrate. However, a question remains: does this work generalize, or do the CFP, MIP, and Polyscheme address the procedural profusion problem only in the few domains they have been designed for? The next section addresses this issue.

## Leveraging a Cognitive Substrate

In order to confirm that a computational system can act as a cognitive substrate, it is necessary to show that its mechanisms can support intelligent behavior in many domains. This section describes how a substrate based on a Polyscheme model of human physical reasoning can be used to construct a natural language parser. This demonstrates that physical reasoning mechanisms can serve as a cognitive substrate and that the computational principles underlying Polyscheme address the integration and profusion problems involved in creating a substrate. Also, since physical theory (with notions such as force, mass, collision, and movement) is apparently so different from syntactic theory (with notions such as empty categories, c-command, and binding principles), demonstrating that the same computational mechanisms can underlie reasoning in both of these domains makes it at least more plausible that intelligent systems in a great many superficially diverse domains can be created with a cognitive substrate.

The key to this work is to recognize that the structures of grammar and of naive physics appear more similar when a verbal utterance is conceived as an event that is composed of a sequence of word utterance subevents. Like physical events, verbal events belong to categories, combine to form larger verbal events, and are ordered in relation to other verbal events according to lawful regularities. This section examines these dualities in detail and shows that many grammatical structures have analogues to nonlinguistic cognitive structures.

## Notation

In order to explain the mapping between syntactic structure and cognitive structures used to represent the physical world, it will be helpful to use a formal notation for representing physical events. This article uses a notion based on the propositions used in Cassimatis (2002) to present problems to his model of physical reasoning. Although there is no claim that the no-

tation resembles the mind's representations for syntactic or physical structure, the next section will show how to use this formalism to present sentences to a model of physical reasoning so that the model can use its own representations and processes to infer the syntactic structure of sentences.

In this formalism, events, objects, and places have names. Predicates describe attributes on and relations among named entities. For example, an event in which an object,  $x$ , moves from  $p1$  to  $p2$  during the temporal interval,  $t$ , is indicated with the following propositions: *Category(e, MotionEvent)*, *Agent(e, x)*, *Origin(e, p1)*, *Destination(e, p2)*, *Occurs(e, t)*. Intervals are ordered using Allan's (1983) temporal relations. For example, *Before(t1, t2)* indicates that  $t1$  finishes before  $t2$  begins and *Meets(t1, t2)* indicates that  $t1$  ends precisely when  $t2$  begins. Category hierarchies are described using subcategory relationships, for example, *Subcategory(Fly, Motion-Event)*. *PartOf(e1, e2)* indicates that event  $e1$  is part of event  $e2$ . That two names for events, objects, or places refer to the same object is indicated using an identity relationship. For example, *Same(o1, o2)* indicates that " $o1$ " and " $o2$ " name the same object. Finally, regularities between physical events can be expressed using material implication. For example, that an unsupported object falls is indicated by:

*Location(o, p1, t1) + Below(p2, p1) + Empty(p2, t1)*

*Category(e, MotionEvent) + Origin(e, p1) + Destination(e, p2) + Occurs(e, t2) + Meets(t1, t2).*

With this background, it is now possible to describe several dualities between syntactic and physical structure.

### Utterances Are Events

The philosophical tradition of "speech act theory" holds that linguistic utterances are actions used to achieve goals. In this way, words are similar to other nonlinguistic actions such as gesturing or tool use. Other people's actions are events we must perceive in order to interpret their intent. Both verbal and nonverbal events occur over temporal intervals. Like nonverbal events, verbal utterances can be executed with various manners (hastily, carefully, loudly, softly). Thus, the same concepts used to describe physical events can be used to describe verbal utterances. For example, using the present notation, the utterance of the word "dog" at time,  $t$ , may be represented, *Category(e, dog-utterance)*, *Occurs(e, t)*.

### Word Order Is Temporal Order

The temporal order of a set of physical events has important consequences for their ultimate result. For example, pulling a gun's trigger *be-*

*fore* loading it results in a much different event from pulling its trigger *after* loading it. This is also a fundamental feature of grammar: the result (in terms of its effect on the listener) of uttering "The dog", uttering "bit" and then uttering "John" is much different from the result of uttering "John", "bit", and then "the dog". In our notation, "John bit the dog" is represented as a sequence of utterance events:

*Category(e1, JohnUtterance), Occurs(e1, t1)*

*Category(e2, BitUtterance), Occurs(e2, t2)*

*Meets(t1, t2)*

...

### Physical and Linguistic Events both Belong to Categories Organized Hierarchically

Word and phrase categories are an important component of almost every serious syntactic theory. Categories are also an essential part of most every other domain of cognition. The previous subsection demonstrated that the same *Category* predicate that represents the category of a physical event can represent the category of a word or phrase utterance. Likewise, just as physical categories exist in hierarchies (for example, *Subcategory(RunningEvent, Motion-Event)*), so do verbal and phrasal categories (for example, *Subcategory(CommonNoun, Noun)* and *Subcategory(TransitiveVerbPhrase, VerbPhrase)*). Just as the category of a physical event determines which other events it occurs with (for example, a gun-firing event tends to be preceded by a trigger-pulling event), so does the category of a word or phrase determine the distribution of words and phrases (for example, transitive verbs are often followed by noun phrases).

### Constituency Is a Parthood Relation

Physical events combine into larger events, which themselves can combine into even larger events. Word utterance events combine into phrase utterance events, which combine into larger phrase utterance events. Parthood relations are thus a feature of both physical and verbal events. Predicates for representing physical event parthood relations can capture phrasal constituency. For example, the noun phrase "the dog" can be represented thus: *Category(e, CommonNounPhrase)*, *Category(e1, Determiner)*, *Occurs(e1, t1)*, *Category(e2, CommonNoun)*, *Occurs(e2, t2)*, *PartOf(e1, e)*, *PartOf(e2, e)*, *Meets(e1, e2)*.

The same notation for expressing physical regularities can be used to represent phrase structure rules and constraints. For example, a rule for a transitive verb's arguments can be expressed thus:

*Category(verb, TransitiveVerb) + Occurs(verb, t-verb)*

*Exists(object) + Category(object, NounPhrase) + Occurs(object, t-object) + Before(t-verb, t-object).*

### Coreference and Binding Are Object-Identity Relationships.

Coreference and binding are perhaps the most obvious identity relationships in language. Consider the following sentence, where “the dog” refers to an object, *d*, “the cat” refers to an object, *c*, and “it” refers to an object, *i*:

The dog chased the cat through the park where it lives.

The reference to “it” is ambiguous. It can refer to the dog (*Same(i, d)*), to the cat (*Same(i, c)*) or to some other object in the conversation or the environment (*Same(i, ?)*). In each case, the coreference is just a special kind of identity relationship. Identity is an extremely widespread and important relationship in everyday physical reasoning. When we lose visual contact with an object because we turn our gaze or because it is occluded and then see a similar object, we must decide whether the sightings are of the same object.

### Phrase Attachment Is an Event Identity Relationship.

The occurrence of a physical event often implies the occurrence of another physical event. For example, when an object resting on a shelf falls to the floor (event *f*), there must have been an event (*p*) that pushed the object off the shelf. One can infer the pushing event from the falling event even if the pushing event is not visible. Later, after observing marks left by a cat’s claws on the shelf, we can infer a cat walking event (*w*). If this cat walking event occurred near the original location of the object that fell, then the cat walking event might be *identical* to the pushing event, that is, *Same(p, w)*.

Event identity is an important feature of grammar as well. For example, the existence of a prepositional phrase utterance within a sentence utterance implies the existence of a noun or verb utterance that the prepositional phrase is an argument or adjunct of. For example, in the sentence “John saw the man with the telescope”, the “with the telescope” utterance event implies the existence of an utterance event, *pp-head*, which takes “with the telescope” as an argument or adjunct. In this case, *pp-head* might be the “John” or “man” utterance event. More formally, either one of the following propositions might be true: *Same(“John”, pp-head)* or *Same(“the man”, pp-*

*head)*. Thus phrase attachment and attachment ambiguity are instances of event identity and uncertainty about event identity.

**Dualities Demonstrate Power of Substrate Hypothesis.** These dualities between physical and linguistic structure have enabled the construction of a natural language syntactic parser (Cassimatis 2004). The parser takes a sentence represented using the physical event language described above and infers propositions that correspond to a parse of a sentence. Figure 2 illustrates such a parse.

The dualities and the parser they enable illustrate several of the benefits of the cognitive substrate hypothesis. First, once you have an implementation of a cognitive substrate, then solving other AI problems becomes easier. It took very little extra work to design a parser once the structural dualities between syntax and physical law were found. No new algorithms need to be designed, thus making the procedural profusion problem less severe. Second, that the substrate used here was a model of physical reasoning suggests that the domain can serve as guidance in identifying the contents of a cognitive substrate. Finally, since the laws of physical motion and syntactic theory are so superficially different, finding a productive mapping between these two domains makes it more plausible that such mappings of a cognitive substrate onto many other domains can be found.

## Conclusions

The work described in this article demonstrates the following benefits of the cognitive substrate approach:

First, creating intelligent systems for new domains is accelerated. The small amount of work needed to create a syntactic parser from a physical reasoner demonstrates that the substrate can make it significantly easier to create an intelligent system for new domains.

Second, integration among domains. An intelligent system that must integrate reasoning between two domains will be easier to construct if reasoners based on a substrate in those two domains already exist, since they will be both based on the same data structures and algorithms.

Third, the problem of achieving human-level AI is reduced and simplified. Instead of needing enormous databases of knowledge and hundreds or thousands of algorithms to achieve human-level intelligence, researchers can focus on solving the problems of human-level AI for a relatively small (but still difficult) set of problems knowing that other domains can be addressed by mapping them onto a substrate.

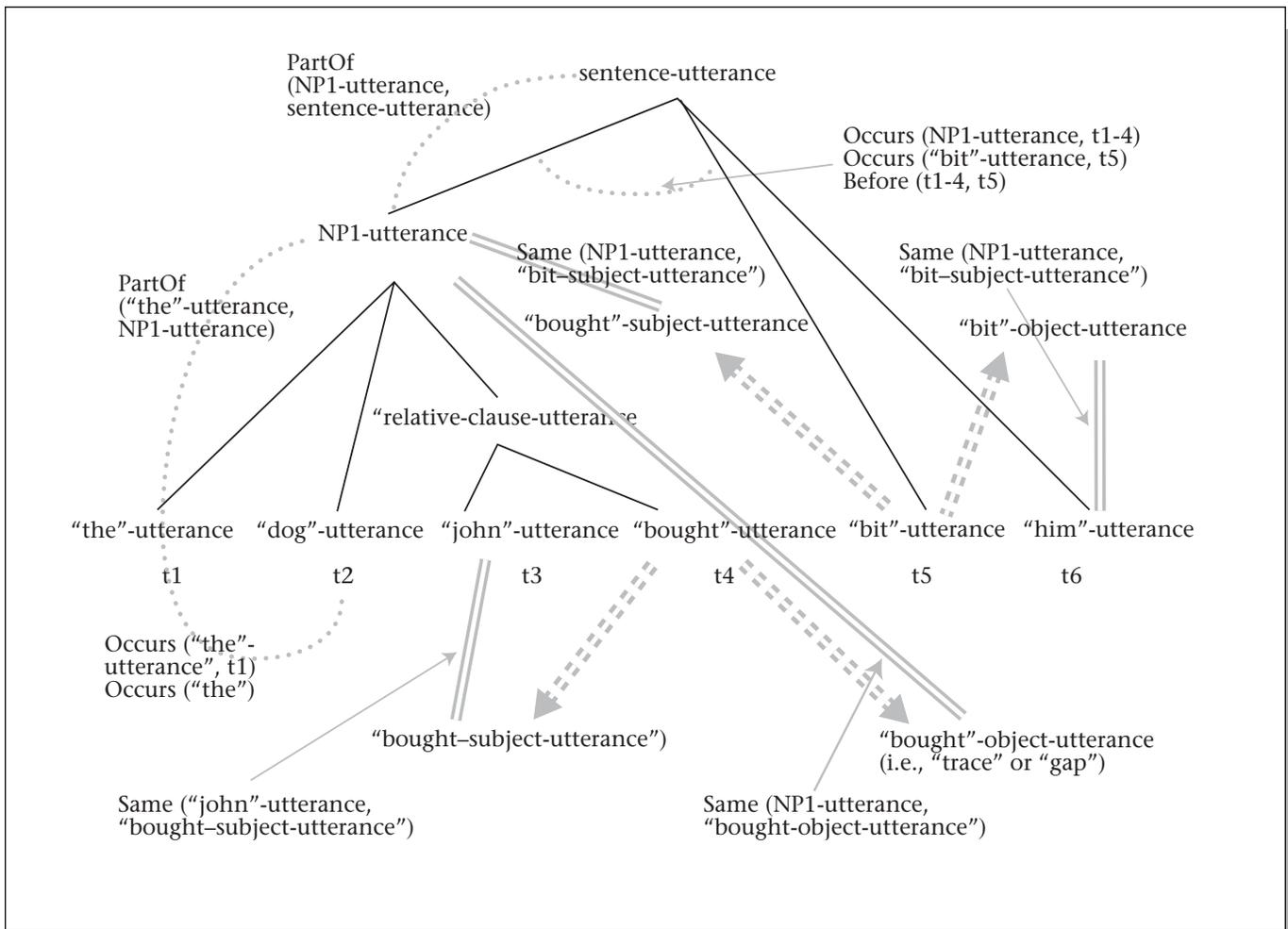


Figure 2. The Syntactic Structure of a Sentence Represented Using Concepts from Infant Physical Reasoning.

As this research program unfolds and a more comprehensive and powerful cognitive substrate is implemented it should in turn take less work to adapt a cognitive substrate to ever more domains and even to automate this process. This would be a significant advance toward achieving human-level intelligence in our lifetime.

## References

- Allen, J. F. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26(11): 832–843.
- Barsalou, L. W.; Simmons, W. K.; Barbey, A.; and Wilson, C. D. 2003. Grounding Conceptual Knowledge in Modality-Specific Systems. *Trends in Cognitive Sciences* 7(2): 84–91.
- Cassimatis, N. L. 2002. *Polyscheme: A Cognitive Architecture for Integrating Multiple Representation and Inference Schemes*. Cambridge, MA: Massachusetts Institute of Technology.
- Cassimatis, N. L. 2004. Grammatical Processing Using the Mechanisms of Physical Inferences. Paper presented at the Twentieth-Sixth Annual Conference of the Cognitive Science Society.
- Cassimatis, N. L. 2005. Integrating Cognitive Models Based on Different Computational Methods. Paper presented at the Twenty-Seventh Annual Conference of the Cognitive Science Society.
- Cassimatis, N. L.; Trafton, J. G.; Bugajska, M.; and Schultz, A. C. 2004. Integrating Cognition, Perception, and Action through Mental Simulation in Robots. *Robotics and Autonomous Systems* 49(1–2): 13–23.
- Cox, M. T.; and Ram, A. 1999. Introspective Multi-strategy Learning: On the Construction of Learning Strategies. *Artificial Intelligence* 112(1–2): 1–55.
- Cree, G. S.; and McRae, K. 2003. Analyzing the Factors Underlying the Structure and Computation of the Meaning of Chipmunk, Cherry, Chisel, Cheese, and Cello (and Many Other Such Concrete Nouns). *Journal of Experimental Psychology: General* 132(2): 163–201.
- Humphreys, G. W.; and Forde, E. M. E. 2001. Hierar-



JULY 13-15, 2006

DARTMOUTH COLLEGE  
ARTIFICIAL INTELLIGENCE CONFERENCE  
THE NEXT FIFTY YEARS

Commemorating the 1956 founding at  
Dartmouth College of AI as a research discipline

**With Original 1956  
Conference Participants:**

- John McCarthy (Stanford)
- Marvin Minsky (MIT)
- Ray Solomonoff (London)
- Oliver Selfridge (MIT)

Visit website for complete program

# Explore the Future of Artificial Intelligence with Today's Foremost Intelligences

## July 13-15, 2006

prior to the AAAI national annual meeting in Boston, MA

[www.dartmouth.edu/~ai50](http://www.dartmouth.edu/~ai50)



[ai50@dartmouth.edu](mailto:ai50@dartmouth.edu)



603-646-8172

chies, Similarity, and Interactivity in Object Recognition: "Category-Specific" Neuropsychological Deficits. *Behavioral Brain Sciences* 24(3): 453-509.

Jackendoff, R. 1990. *Semantic Structures*. Cambridge, MA: The MIT Press.

Jordan, M. I.; and Jacobs, R. A. 2002. Learning in Modular and Hierarchical Systems. In *The Handbook of Brain Theory and Neural Networks*, 2nd ed., ed. M. Arbib. Cambridge, MA: The MIT Press.

Lenat, D. B.; and Guha, R. V. 1990. *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project*. Reading, MA: Addison-Wesley.

Minsky, M. L. 1986. *The Society of Mind*. New York: Simon and Schuster.

Mueller, E. T. 1998. *Natural Language Processing with ThoughtTreasure*. New York: Signiform.

Newell, A.; and Simon, H. A. 1972. *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann.

Richardson, D. C.; Spivey, M. J.; Barsalou, L. W.; and McRae, K. 2003. Spatial Representations Activated during Real-Time Comprehension of Verbs. *Cognitive Science* 27(5): 767-780.

Richardson, D. C.; Spivey, M. J.; Edelman, S.; and Naples, A. D. 2001. "Language Is Spatial": Experi-

mental Evidence for Image Schemas of Concrete and Abstract Verbs. Paper presented at the 23rd Annual Meeting of the Cognitive Science Society.

Schank, R. C. 1975. *Conceptual Information Processing*. Amsterdam: Elsevier, North-Holland.

Tanenhaus, M. K.; and Trueswell, J. C. 1995. Sentence Comprehension. In *Handbook of Perception and Cognition* volume 11, ed. J. Miller and P. Eimas, 217-262. San Diego, CA: Academic Press.

Warrington, E. K.; and Shallice, T. 1984. Category Specific Semantic Impairments. *Brain* 107(3): 829-854.



**Nicholas Cassimatis** is an assistant professor of cognitive science at the Rensselaer Polytechnic Institute. He received his B.S. in mathematics from the Massachusetts Institute of technology, his M.A. in psychology from Stanford University, and his Ph.D. from the MIT Media Laboratory. The overall

goal of his research is to understand and reproduce in computers human-level intelligence. This has led him to do work in robotics, machine learning, cognitive development, physical reasoning, language understanding, and social reasoning.