Patterns, Hypergraphs and Embodied General Intelligence

Ben Goertzel

Abstract—**It is proposed that the creation of Artificial General Intelligence (AGI) at the human level and ultimately beyond is a problem addressable via integrating computer science algorithms and data structures within a cognitive architecture oriented toward experiential learning. A general conceptual framework for AGI is presented, beginning with a philosophy of mind based on the concept of pattern, then moving to a general mathematical and conceptual framework for modeling intelligent systems (SMEPH = Self-Modifying Evolving Probabilistic Hypergraphs), and finally to an overview of a specific design for AGI, the Novamente AI Engine. The problem of teaching an AGI system is discussed, in the context of Novamente's embodiment in the AGI-SIM simulation world. An educational program based loosely on Piaget's developmental stages is outlined, followed by more detailed consideration of the learning by Novamente in AGI-SIM of the Piagetan infant-level capability of "object permanence."**

I. Introduction

What is meant by "AI" these days generally bears scant resemblance to the grand visions that accompanied the founding of the field in the middle of the previous century. AI nowadays nearly always means "narrow AI" [1] – the creation of software programs that carry out highly specific functionalities that are typically considered "intelligent" when humans carry them out. Because of this I have lately adopted the terminology "Artificial General Intelligence" (AGI) to refer to the pursuit of software systems that display a wide variety of intelligent functionalities, including a reasonably deep understanding of themselves and others, the ability to learn how to solve problems in areas they've never encountered before, the ability to create new ideas in a variety of domains, and the ability to communicate richly in language. AI research flourishes; AGI research has been experiencing a long winter, from which it is now beginning to emerge.

In this paper I summarize a set of ideas that I have developed during the last two decades, which has led me to what I believe is a novel and productive way of thinking about general intelligence, and also to a specific design for an AGI system: the Novamente AGI design. I review a series of three closely interconnected topics: the patternist philosophy of mind, the SMEPH formalism for modeling intelligent systems, and then the Novamente design. Each of these in itself is too large a topic for a brief conference paper, and so the discussion will necessarily be somewhat abstract. However, at the end I will get concrete and describe some of the specific learning experiments we are now doing with the Novamente system, aimed at having it learn the sorts of things that a human infant learns when interacting with the world, and considered as the first steps in a coherent educational program with the end goal of general intelligence at the human level and beyond.

B. Goertzel is with the Applied Research Laboratory for National and Homeland Security at Virginia Tech, 2000 N. 15th St., Suite 503 , Arlington VA 22201; and also with Novamente LLC, 1405 Bernerd Place, Rockville MD 20851 (phone: 240-505-6518; e-mail: ben@goertzel.org).

II. The Surprising Unpopularity of AGI

Why does AGI get so little attention nowadays? By and large, I suggest, it's not because AI researchers believe AGI is impossible. The philosophy literature contains a variety of arguments against the possibility of generally intelligent software, but none are very convincing. Perhaps the most philosophically sensible counterargument is the Penrose/Hameroff speculation that human intelligence is based on unspecified quantum gravity based dynamics operating within brain dynamics [2]; but scientific evidence in favor of this conjecture is nonexistent. My impression is that most contemporary scientists believe that AGI at the human level and beyond is possible in principle.

The most articulate argument so far created *in favor* of the in-principle possibility of AGI is Marcus Hutter's [3] theoretical work on algorithmic information theory and decision theory, which involves positing a very general mathematical definition of intelligence and then proving rigorously that arbitrarily high degrees of intelligence are possible given arbitrarily large amounts of computational power. This theoretical work rigorously shows what has been obvious to many researchers for a long time: that AGI is at bottom a problem of processing and memory efficiency. With enough computing power, making AGI is trivial and can be done in a few dozen lines of easily-formulated LISP code. But this insight doesn't help very much in creating practical AGI systems using tractable amounts of computational power. In effect, the human brain consists of a collection of more or less clever tricks for achieving various sorts of more or less general intelligence within rather strict computation-power constraints.

Many AI researchers seem to take the position that, while AGI is in principle possible, it lies far beyond our current technological capability. This is a reasonable enough contention, since according to the best available estimates [4] (which are quite speculative), current computing hardware at AI researchers' disposal falls significantly short of the computing power of a single human brain. But even if it's true that current computers are much less powerful than the human brain, this isn't necessarily an obstacle to creating powerful AGI on current computers using fundamentally non-brain-like architectures.

I believe the core reason there has been so little detailed research work on AGI is that there have been so few even moderately convincing general ideas in the area of AGI design. The paucity of plausible AGI designs has been so severe that a number of highly knowledgeable researchers have effectively given up hope, opining that the only or most likely path to AGI is going to be the emulation of the human brain. Eric Baum has presented this perspective very articulately in terms of the concept of "inductive bias" [5]. Much of human intelligence, he argues, is based on tacit knowledge accumulated over generations of evolution, which cannot feasibly be explicitly encoded in software – for similar reasons to those underlying the inability of linguists to fully articulate the tacit rules we use in processing language, in spite of decades spent trying to spell out formal grammars for natural language. Ray Kurzweil [6] has argued that brain imaging will yield a reasonably complete understanding of human brain structure and dynamics by the middle of this century, and that the achievement of AGI via human brain emulation will likely follow not long thereafter.

I find the Baum/Kurzweil perspective a plausible one, yet intuitively I don't believe it. The research project reported here seeks to refute this perspective via providing a detailed, thorough, high-quality design for a non-human-like AGI system, the Novamente system. In-depth publications on aspects of Novamente are in preparation [7]-[8], and a few prior papers have touched on a variety of aspects of the system [9]-[10]-[11]. Here I will by and large remain at a higher level of abstraction and talk about what sort of AGI design I think it makes the most sense to pursue. I will use the "patternist philosophy of mind" underlying Novamente as a way of structuring the analysis and discussion of the AGI problem.

While there has been a surprisingly long "AGI winter," at the moment there seem to be some signs of a renaissance, and the Novamente project is not the only one to emerge lately addressing the AGI problem. A complete review of the current literature would be out of place here but among the recent projects with the closest relationship to Novamente must be listed Pei Wang's NARS project [12]-[13]-[14], John Weng's SAIL architecture [15], Nick Cassimatis's PolyScheme [16], Stuart Shapiro's SnEPs [17], and Robert Hecht-Nielsen's confabulation approach [18]. All these are current projects actively addressing AGI.

To fully explore the relation between these projects and Novamente would take us too far afield, but suffice it to say that the relationships exist and are interesting. For instance: NARS is based on an uncertain logic closely related to Novamente's PTL inference system. SnEPs is based on paraconsistent logic, whereas Novamente's PTL logic is also paraconsistent; furthermore, both SnEPs and Novamente have been used to control an agent in a simulation world based on the CrystalSpace game engine, based on somewhat similar approaches to embodied perception and action. Hecht-Nielsen's "confabulation" operation occurs naturally within Novamente as a consequence of PTL inference.

In spite of the various similarities, however, there are also significant differences between these other recent approaches and Novamente; and some of these differences are foundational and conceptual rather than technical. Novamente embodies a particular conceptual understanding of mind and intelligence; in this brief overview my goal is to get across a few important aspects of this conceptual understanding and explain how they manifest themselves in the AI architecture and in our plan for teaching Novamente.

III. PATTERNIST PHILOSOPHY OF MIND

The ultimate conceptual foundation of Novamente is patternist philosophy of mind: a general approach to thinking about intelligent systems, which is based on the very simple premise that "mind is made of pattern." This in itself is not a very novel idea – it is present, for instance, in the 19'th-century philosophy of Charles Peirce [19], in the writings of contemporary philosopher Daniel Dennett [20], in Benjamin Whorf's [21] linguistic philosophy and Gregory Bateson's [22] systems theory of mind and nature. Bateson spoke of the Metapattern: "that it is pattern which connects." In a series of prior writings [23]-[24]-[25]-[26]-[27] and in a forthcoming book [28] I have sought to pursue this theme more thoroughly than has been done before, and to articulate in detail how various aspects of human mind and mind in general can be well-understood by explicitly adopting a patternist perspective. This work, which has previously been labeled the "psynet model of mind," includes attempts to formally ground the notion of pattern in mathematics such as algorithmic information theory [29] and probability theory, beginning from the conceptual notion that "a pattern is a representation as something simpler" and then utilizing appropriate mathematical concepts of representation and simplicity.

In the patternist perspective, the mind of an intelligent system is conceived as the set of patterns in that system, and the set of patterns emergent between that system and other systems with which it interacts. The latter clause means that the patternist perspective is inclusive of notions of "distributed intelligence" [30]. Intelligence is conceived, similarly to in Hutter's work, as the ability to achieve complex goals in complex environments; where complexity itself may be defined as the possession of a rich variety of patterns. A mind is thus a collection of patterns that is associated with a persistent dynamical process that achieves highly-patterned goals in highly-patterned environments.

An additional hypothesis made within the patternist philosophy of mind is that reflection is critical to intelligence. This lets us conceive an intelligent system as a dynamical system that recognizes patterns in its environment *and itself*, as part of its quest to achieve complex goals.

While this approach is quite general, it is not vacuous; it gives a particular structure to the tasks of analyzing and synthesizing intelligent systems. About any would-be intelligent system, we are led to ask questions such as:

--How are patterns represented in the system? That is, how does the underlying infrastructure of the system give rise to the displaying of a particular pattern in the system's behavior?

--What kinds of patterns are most compactly represented within the system?

--What kinds of patterns are most simply learned?

--What learning processes are utilized for recognizing patterns?

--What mechanisms are used to give the system the ability to introspect (so that it can recognize patterns in itself?)

Now, these same sorts of questions could be asked if one substituted the word "pattern" with other words like "knowledge" or "information." However, I have found that asking these questions in the context of pattern leads to more productive answers, because the concept of pattern ties in very nicely with the details of various existing formalisms and algorithms for knowledge representation and learning.

IV. SELF-MODIFYING, EVOLVING PROBABILISTIC HYPERGRAPHS

Patternist philosophy is extremely general, which is both a strength and a weakness. In order to more effectively apply it to the AGI problem, I have created an intermediate formalism called SMEPH (Self-Modifying, Evolving Probabilistic Hypergraphs). SMEPH is a more specific formalism for describing intelligent systems, which is consistent with patternist philosophy but provides more guidance regarding the analysis and construction of particular intelligent systems.

The basic ideas underlying SMEPH are threefold, as the acronym would suggest:

--To use a specific mathematical structure called a "generalized hypergraph" to model intelligent systems.

--To study the way hypergraphs change over time (i.e. they way they evolve" -- the word "evolution" is used here in a general sense, rather than specifically in the sense of evolution by natural selection, although that is an aspect of SMEPH as well when one delves into the details)

--To use probability theory to study the relationships between the parts of the hypergraph

A hypergraph is an abstract mathematical structure [31], which consists of objects called Vertices and objects called Edges, which connect the Vertices. In computer science, a "graph" traditionally means a bunch of dots connected with lines (i.e. "vertices" connected by "edges", or "nodes" connected by "links"). A hypergraph, on the other hand, can have Edges that connect more than two Vertices; and SMEPH's hypergraphs extend ordinary hypergraphs to contain additional features such as Edges that point to Edges instead of Vertices; or Vertices that, when you zoom in on them, contain little hypergraphs. Properly, SMEPH's hypergraphs should always be referred to as "generalized hypergraphs," but this is cumbersome, so we will persist in calling them "hypergraphs" instead. In a hypergraph of this sort, edges and vertices are not as distinct as they are within an ordinary mathematical graph (for instance, they can both have edges connecting them), and so it is useful to have a generic term encompassing both Edges and Vertices; for this purpose, in SMEPH and Novamente, we use the term "Atom."

A "weighted, labeled hypergraph" is a hypergraph whose Edges and Vertices come along with labels, and with one or more numbers that are generically called "weights." The label associated with an Edge or Vertex may sometimes be interpreted as telling you what "type" of entity it is. On the other hand, an example of a weight that may be attached to an Edge or Vertex is a number representing a probability, or a number representing how important the Vertex or Edge is to the system.

Hypergraphs may come along with various dynamics. For instance, one may think about:

--Dynamics that modify the properties of Vertices or Edges in a hypergraph (such as the weights attached to them)

--Dynamics that add new Vertices or Edges to a hypergraph, or remove existing ones.

The SMEPH approach to intelligence is centered on a particular collection of Vertex and Edge types. The key Vertex types are ConceptVertex and SchemaVertex, the former representing an idea or a set of percepts, and the latter representing a procedure for doing something (perhaps something in the physical world, or perhaps an abstract mental action). The key Edge types are ExtentionalInheritanceEdge (ExtInhEdge for short: an edge which, linking one Vertex or Edge to another, indicates that the former is a special case of the latter), ExtensionalSimilarityEdge (ExtSim: which indicates that one Vertex or Edge is similar to another), and ExecutionEdge (a ternary edge, which joins {S,B,C} when S is a SchemaVertex and the result from applying S to B is C). So, in a SMEPH system, one is often looking at hypergraphs whose Vertices represent ideas or procedures, and whose Edges represent relationships of specialization, similarity or transformation among ideas and/or procedures.

ExtInh and ExtSim Edges come with probabilistic weights indicating the extent of the relationship they denote (e.g. the ExtSimEdge joining the "cat" ConceptVertex to the "dog" ConceptVertex gets a higher probability weight than the one joining the "cat" ConceptVertex to the "washing-machine" ConceptVertex). The mathematics of transformations involving these probabilistic weights becomes quite involved -- particularly when one introduces SchemaVertices corresponding to abstract mathematical operations, a step that enables SMEPH hypergraphs to have the complete mathematical power of standard logical formalisms like predicate calculus, but with the added advantage of a natural representation of uncertainty in terms of probabilities, as well as a natural representation of networks and webs of complex knowledge.

SMEPH hypergraphs may be used to model and describe intelligent systems (such as human mind/brains, for example). One can (in principle) draw a SMEPH hypergraph corresponding to an individual intelligent system, with Vertices and Edges for the concepts and processes in that system's mind. This leads to what is called the "derived hypergraph" of that system. More specifically, a ConceptVertex in the derived hypergraph of a system corresponds to a structural pattern that persists over time in that system; whereas a SchemaVertex corresponds to a multi-time-point dynamical pattern that recurs in that system's dynamics. Drawing the derived hypergraph of an intelligent system is one way of depicting the mind of that system – this follows from the definition of a mind as the set of patterns in an intelligent system, and the fact (which follows from mathematical pattern theory) that the patterns in the system can be read off from the derived hypergraph.

Pattern theory enters more deeply here when one thoroughly fleshes out the Inheritance concept. Philosophers of logic have extensively debated the relationship between "extensional" inheritance (inheritance between sets based on their members) and "intensional" inheritance (inheritance between entity-types based on their properties). A variety of formal mechanisms have been proposed to capture this conceptual distinction; see [12]-[13]-[14] for a review along with a novel approach utilizing uncertain term logic. Pattern theory provides a novel approach to defining intension: one may associate with each ConceptVertex in a system's derived hypergraph the set of patterns associated with the structural pattern underlying that ConceptVertex. Then, one can define the strength of the IntensionalInheritanceEdge between two ConceptVertices A and B as the percentage of A's pattern-set that is also contained in B's pattern-set. According to this approach, for instance, one could have

```
ExtInhEdge whale fish <0>

IntInhEdge whale fish <.6>
```

(where "R A B" denotes an Edge of type R between Vertices A and B; and the numbers in <>s denote strength values associated with Edges).

As well as being used to conceptually model intelligent systems, SMEPH hypergraphs may also be used as the foundation of an AGI design. In this case, a SMEPH hypergraph is used explicitly as the medium for the (long and short term) memory of an intelligent system, and its thought processes are explicitly described and implemented as dynamics modifying this hypergraph. Such a SMEPH-based intelligence will also have a derived hypergraph, which will not be identical to the hypergraph it uses for explicit knowledge representation. However, an interesting feedback loop arises here, in that the intelligence's self-study will generally lead it to recognize large portions of its derived hypergraph as patterns in itself, and then embody these patterns within its concretely implemented knowledge hypergraph. The Novamente AI system, which I will discuss here, is the second in a series of AGI-oriented AI systems specifically based on the SMEPH framework; the first was the Webmind AI Engine [27]-[32].

V. KNOWLEDGE REPRESENTATION, LEARNING AND REASONING IN NOVAMENTE

The remainder of the paper focuses on the Novamente AI Engine, a specific software design and software system aimed at powerful AGI. The Novamente project is ongoing, and the current software implementation of the Novamente AI design is somewhere between 20% and 60% complete, depending on how you measure it. What is discussed here is mainly the AGI design rather than the state of the current implementation; however I will occasionally insert comments regarding what has currently been implemented and tested and what has not.

Novamente's knowledge representation must be considered on two levels: implicit and explicit. The explicit knowledge representation is a SMEPH-style generalized hypergraph, which I will refer to here as a hypergraph of Nodes and Links, to distinguish it from the Vertices and Edges in the SMEPH derived hypergraph of a Novamente system. This includes ConceptNodes and SchemaNodes, where SchemaNodes are represented as mathematical objects using arithmetic, logical and combinatory operators to combine elementary data types and Novamente Nodes and Links, designed to enable compact expression of useful cognitive procedures. It also includes a number of other node types including PredicateNodes (SchemaNodes that produce truth values as their outputs) and various kinds of Nodes representing particular kinds of concrete information, such as NumberNodes, WordNodes, PolygonNodes, and so forth. A moderately extensive list is given in [11] and will not be repeated here.

In addition to explicit knowledge representation in terms of Nodes and Links, Novamente also incorporates implicit knowledge representation in the form of what are called "maps": collections of Nodes and Links that tend to be utilized together within cognitive processes. To see the need for maps, consider that even a Node that has a particular meaning attached to it – like the "Iraq" Node, say – doesn't contain much of the meaning of "Iraq" in it. The meaning of "Iraq" lies in the Links attached to this Node, and the Links attached to their Nodes – and the other Nodes and Links not explicitly represented in the system, which will be created by Novamente's cognitive algorithms based on the explicitly existent Nodes and Links related to the "Iraq" Node. This halo of Atoms related to the "Iraq" node is called the "Iraq" map. In general, some maps will center around a particular Atom, like this "Iraq" map, others may not have any particular identifiable center. Novamente's cognitive processes act directly on the level of Nodes and Links, but they must be analyzed in terms of their impact on maps as well. In SMEPH terms, Novamente maps may correspond to SMEPH ConceptVertices, and for instance bundles of Links between the Nodes belonging to a map may correspond to a SMEPH Edge between two ConceptVertices

SMEPH ExtInhLinks and IntensionalInheritanceLinks exist in Novamente, along with a variety of related link types; Novamente contains a probabilistic reasoning engine called Probabilistic Term Logic (PTL) which exists specifically to carry out reasoning on these relationships, and will be described in a forthcoming publication [8]. The mathematics of PTL contains many subtleties, and there are relations to prior approaches to uncertain inference including NARS [12]-[12]-[14] and Walley's theory of interval probabilities [33]. An essentially complete software implementation of PTL exists within the current Novamente codebase and has been tested on various examples of mathematical and commonsense inference.

In addition to PTL, Novamente's other main learning mechanism is a modification of evolutionary learning called MOSES (Meta-Optimizing Semantic Evolutionary Search), a descendant of the Bayesian Optimization Algorithm Programming (BOAP) algorithm described in [34]-[35]. MOSES is an algorithm for learning PredicateNodes or SchemaNodes satisfying specified criteria. For instance if a goal G and context C are given then it may be used to learn a compact SchemaNode S so that the statement

```
"When context C holds and schema S is
executed, goal G is achieved"
```

holds with a high truth value. (Typically the goal and context would be specified as PredicateNodes.) MOSES is a modification of the genetic programming algorithm [36], but with some substantial differences, including:

--As in BOA [37], crossover and mutation are augmented by a probabilistic modeling process in which the population of candidate solutions is studied statistically and then new candidate solutions are generated from the inferred probability distribution

--Candidate "programs" (PredicateNodes/ SchemaNodes) are normalized using algebraic simplification routines prior to being probabilistically modeled or crossed over

BOAP was integrated into the Novamente codebase in 2003 and tested on a number of examples in the domain of quantitative and relational data mining. MOSES is currently under active development within the Novamente system and at time of writing is being tested on various relevant problems on its own and in combination with PTL.

MOSES complements PTL: whereas PTL's job is to extrapolate existing knowledge and build new Nodes and Links that directly follow from old ones in an incremental way, MOSES's job is to create complex combinations of Nodes and Links "out of the blue," via heuristic, evolutionary/probabilistic exploration of the large space of possibilities. The interrelation between these two learning algorithms was described in a little more depth in prior overview papers on the Novamente system [9]-[10]-[11].

Pattern theory enters here in a very direct way, via the standard inclusion of a "compactness criterion" in the fitness function used to guide MOSES's evolutionary learning. Without a compactness criterion on the fitness function, MOSES would simply learn complex patterns "overfit" to its historical training data; the compactness criterion means that MOSES has to actually recognize patterns in its training data (keeping in mind the definition of a pattern as "a representation as something simpler"). MOSES searches for patterns out of the blue; PTL takes existing patterns and uses them to incrementally infer new ones.

The use of PTL and MOSES to recognize patterns among the Nodes and Links in the AtomTable, and to create new Nodes and Links based on these patterns, is a specific manifestation of the general idea introduced above of an intelligent system studying itself, recognizing its own derived SMEPH hypergraph, and thus embodying this derived hypergraph in its own explicit structure. Novamente's learning algorithms may recognize patterns in Nodes and Links representing perceptions and actions, but also in Nodes and Links representing abstract ideas and even self-models: in this way the architecture is built with introspection at its foundation.

The final critical aspect of Novamente learning is "attention allocation" or "assignment of credit." This has to do with regulating the system's own cognitive activities, an issue that has many different aspects. Firstly, in practice a Novamente instance can't maintain an arbitrarily large hypergraph in memory, so prioritization decisions must be made regarding which Nodes and Links to remove from RAM and save to disk. Next, among those Atoms remaining in RAM, decisions must be made regarding which ones to think about: which ones to feed to PTL reasoning, and which ones to consider as goals for the guidance of MOSES learning. A variety of schemes for making these sorts of decisions exist in the AI literature [38], but Novamente takes a somewhat novel approach. Special Links called HebbianLinks are created, indicating the degree to which the utility of one Atom implies the utility of another. PTL and MOSES are then used to infer new HebbianLinks and new PredicateNodes involving HebbianLinks, from the original HebbianLinks learned via direct experience. In short, these "meta-level" learning processes are handled via the same cognitive mechanisms used for ordinary learning. This approach to credit assignment has not yet been implemented and is scheduled for coding and testing in mid-2006.

VI. NOVAMENTE'S COGNITIVE ARCHITECTURE

The cognitive architecture within which the representational, learning and reasoning mechanisms above exist is a fairly simple one. A Novamente instance is divided into a set of Units, each of which contains an AtomTable containing a hypergraph of Nodes and Links, and also a set of MindAgent objects embodying various cognitive processes (see [11] for a fairly comprehensive list of MindAgents). Example MindAgents include Clustering, Spontaneous First-Order Inference, Goal-Directed Inference, Object Recognition, and Credit Assignment. The MindAgents are perpetually cycled through, carrying out recurrent actions and creating Task objects that carry out processor-intensive one-time actions. Different Units deal with different high-level cognitive functions and may contain different mixes of MindAgents, or at least differently-tuned MindAgents.

In [10] we describe a specific Novamente configuration, intended for “experiential learning” – and more specifically, for a Novamente system that controls a real or simulated body that is perceiving and acting in some world. Currently we are not working with physical robotics but are rather using Novamente to control a simple simulated body in a 3D simulation world called AGI-SIM [39]. It would also be possible to construct Novamente configurations unrelated to any kind of embodiment; for instance, we have designed a configuration intended specifically for mathematical theorem-proving. However, as argued in [40], we believe that pursuing some form of embodiment is likely the best way to approach AGI. This is not because intelligence intrinsically requires embodiment, but rather because physical environments present a host of useful cognitive problems at various levels of complexity, and also because understanding of human beings and human language will probably be much easier for AI’s that share humans’ grounding in physical environments.

This experiential learning configuration centers around a Unit called the Central Active Memory, which is the primary cognitive engine of the system. There is also a Unit called the Global Attentional Focus, which deals with Atoms that have been judged particularly important and subjects them to intensive cognitive processing. There are Units dealing with sensory processing and motor control; and then Units dealing with highly intensive PTL or MOSES based pattern recognition, using control mechanisms that are not friendly about ceding processor time to other cognitive processes. Each Unit may potentially span multiple machines; the idea is that communication within a Unit must be very rapid, whereas communication among Units may be slower.

Psychologically, one may think of the Novamente system’s activities as falling into two categories: goal-driven and ambient. Ambient cognitive activity includes for instance

--MindAgents that carry out basic PTL operations on the AtomTable, deriving obvious conclusions from existing knowledge

--MindAgents that carry out basic perceptual activity, e.g. recognizing coherent objects in the perceptual stimuli coming into the system

--MindAgents related to attention allocation and assignment of credit

--MindAgents involved in moving Atoms between disk and RAM.

Goal-driven activity, on the other hand, involves an explicitly maintained list of goals that is stored in the Global Attentional Focus and Central Active Memory. Two key processes are involved:

--Learning SchemaNodes that, if activated, are expected to lead to goal achievement

--Activating SchemaNodes that, if activated, are expected to lead to goal achievement

--The goal-driven learning process is ultimately a form of “backward-chaining learning,” but subtler than usual instances of backward chaining due to its interweaving of PTL and MOSES and its reliance on multiple cognitive Units.

VII. Developmental Stages

So far I have very loosely described a cognitive architecture, a knowledge representation and a set of learning mechanisms. These merely set the stage for the self-organization and reflective learning processes that are what really make a mind: they lead us to the fascinating and critical topic of AGI education. The basic principle underlying any reasonable AGI educational program must be the hierarchical composition of (conceptual, perceptual and behavioral) patterns. Advanced intelligence requires the recognition of complex patterns, but the search space of possible complex patterns is very large, and so a mind must work up to learning complex patterns via starting out with simple patterns and then incrementally building more and more complex patterns from the ones it already knows. PTL and MOSES are designed to be good at this kind of hierarchical building. The point of teaching an AGI is to present it with a series of learning problems that require it to learn to recognize more and more complex patterns, in an order that matches naturally with the logical buildup of more and more complex patterns from initially simple elements.

The teaching program we are using for Novamente is based on a loose adaptation of Jean Piaget’s classic development psychology ideas [41] to the context of the AGI-SIM simulation world. Our approach to developmental psychology is based on an attempt to integrate Piaget's conceptual insights with more recent developmental psychology theories [42, 43] in a manner consistent with Novamente, SMEPH and patternist philosophy.

Piaget conceived of child development as falling into four stages, each roughly identified with an age group: infantile, preoperational, concrete operational, and formal.

--*Infantile*: Basic world-exploration; instinctive actions; reward-driven repetition of actions; imitation of others' actions; simple associations between words and object, actions and images. One of the major learning achievements here is object permanence – infants learn that objects persist even when not being observed.

--*Preoperational*: The formation of mental representations, mostly poorly organized and un-abstracted; mostly intuitive rather than logical thinking. Word-object and image-object associations become systematic rather than occasional. Simple syntax is mastered, including an understanding of subject-argument relationships.

--*Concrete Operational*: More abstract logical thought applied to the physical world. Among the feats achieved here are: reversibility -- the ability to undo steps already done; conservation -- understanding that properties can persist in spite of appearances; theory of mind – an understanding of the distinction between what I know and what others know. (If I cover my eyes, can you still see me?) Concrete operations such as putting items in height order are easily achievable. Classification become more sophisticated.

--*Formal*: Abstract deductive reasoning, the process of forming then testing hypotheses, etc. This is full, adult human-level intelligence. Note that the capability for formal operations is intrinsic in the PTL component of Novamente, but in-principle capability is not the same as pragmatic, embodied, controllable capability.

Inspired by Piaget's general ideas we have created our own series of developmental stages, defined roughly as follows:

--*Infantile*: Able to recognize patterns in and conduct inferences about the world, but only using simplistic hard-wired (not experientially adapted) inference control schemata

--*Concrete Operational*: Able to carry out more complex chains of reasoning regarding the world, via using inference control schemata that adapt their behavior based on experience (reasoning about a given case in a manner similar to what worked in prior similar cases).

--*Formal*: Able to carry out arbitrarily complex inferences (constrained only by computational resources) via including inference control as an explicit subject of abstract learning.

--*Reflexive*: Capable of thorough self-modification of all internal structures.

Here Piaget's pre-operational phase appears as transitional between the infantile and concrete operational phases. We suspect this approach to cognitive development has general value beyond Novamente, though to argue this point would bring us too far afield here. We have designed specific Novamente / AGI-SIM learning tasks based on all the key Piagetan themes. Currently our concrete work is near the beginning of this list, at Piaget's infantile stage.

VIII. Learning Object Permanence

Next I will discuss the specific task of learning object permanence, a topic which will require a brief digression into the simple visual system via which Novamente interfaces with the AGI-SIM world. Rather than perceiving individual pixels or voxels within AGI-SIM, Novamente perceives AGI-SIM in terms of polygons. A PolygonNode represents a polygon observed at a point in time. A PersistentPolygonNode (PPNode) then represents a series of PolygonNodes that are heuristically guessed to represent the same PolygonNode at different moments in time. Before object permanence is learned, the heuristics for recognizing PPNodes will only work in the case of a persistent polygon that, over an interval of time, is experiencing relative motion within the visual field, but is never leaving the visual field. For example some useful heuristics are: If P1 occurs at time t, P2 occurs at time s where s is very close to t, and P1 are similar in shape, size and color and position, then P1 and P2 should be grouped together into the same PPNode.

Next, AdjacencyLinks are created between PPNodes, via a special formula that maps the relative positions of two polygons into a "strength" value in [0,1]. Then a Clustering MindAgent looks for clusters in the graph of AdjacencyLinks between PPNodes: these clusters become AGISIMObjectNodes. These mechanisms are relatively straightforward – all they do is recognize an object as a set of persistent polygons that cohere together within the visual field during some continuous interval of time. If an observed object leaves the visual field and then re-enters, then these low-level in-built mechanisms don't tell Novamente anything about it. If a ball disappears behind a chair and then reappears, then upon reappearance it is classified as a new object! The Piagetan task of object permanence requires Novamente to learn that in fact it is still the same ball after it has reappeared.

This is not a very hard reasoning task. For instance, if the system is given multiple balls to play with, with different (unique) markings on them, then it can learn via experience that if a ball with marking X goes behind the chair, and it then goes behind the chair, it will find a ball with marking X rather than some other marking. Simple though it seems, this knowledge is represented in Novamente via a predicate involving a couple dozen different Nodes and Links, and learning it either requires a lengthy MOSES run or some fairly intensive backward-chaining inference. And a more interesting sort of inference occurs after this. Suppose the system has learned that balls retain markings: can it then extend this knowledge to infer the permanence of other sorts of objects? This requires what in PTL theory is called abductive inference [11]-[12].

This example illustrates the difference between AGI research and narrow-AI research. In this case, we are making Novamente learn something that we could very easily tell it instead (information regarding what objects exist in AGI-SIM is there explicitly in the AGI-SIM server, and could merely be passed to Novamente). We take this approach because we believe that minds most naturally learn complex things via analogy to simple things, and that analogies are most easily drawn to concepts and procedures about which a rich network of patterns has been formed. When Novamente reaches the concrete operational stage and needs to learn conservation laws, its job will be easier because it will be able to draw on its experience learning object permanence. Conservation of mass is basically "mass permanence," and the procedures it has developed for "learning about permanence" in the context of object permanence will be useful for it in learning about mass permanence. This simple example illustrates the general principle of composition of patterns, via which learning algorithms build complex patterns from simple ones.

IX. Conclusion

Creating AGI at the human level and ultimately beyond is, I suggest, a very difficult but not impossible problem to solve. There is no reason to believe that emulating the human brain is the only viable solution. General intelligence is complex but not all that mysterious.

General intelligence requires a robust mechanism for the representation of general patterns, which gives compact representations to particular patterns of use to a particular system adapted to in a particular environment. It then requires learning algorithms for extrapolating new patterns from existing ones (i.e. for recognizing aspects of the derived SMEPH hypergraph implicit in a system and explicitly embodying them within the system's knowledge base). General learning algorithms are needed, both incremental ones (like PTL) and global, speculative ones (like MOSES). Specialized learning algorithms are needed, in order to address frequently encountered resource-intensive learning problems in an efficient way (the specific heuristics for dealing with polygons mentioned above are an example of this). A flexible cognitive architecture is needed, able to incorporate ambient and goal-directed learning and to integrate various general and specialized learning mechanisms. Attention allocation and assignment of credit must be carried out effectively, which can be done if they are taken seriously and treated as difficult pattern recognition problems on par with others. Finally, recognizing complex patterns right from the start is too hard -- a mind must receive a sensible education that encourages the build-up of more and more complex patterns in a meaningful order; and one natural way to structure this educational process is to embed the mind in a body perceiving and acting in a world.

Patternist philosophy and SMEPH are not the only way to think about general intelligence; and Novamente is not the only possible AGI design consistent with patternist philosophy and SMEPH, let alone the only feasible AGI design. Almost surely there are many viable routes to AGI – and my suspicion is that there are many viable routes that are achievable using current computer science knowledge and current computing hardware. My goal here has been to present a high-level overview of one viable approach.

Acknowledgment

The author would like to thank the members of the Novamente technical team, especially Cassio Pennachin, Moshe Looks, Ari Heljakka, Andre' Senna, Izabela Freire Goertzel, Matt Ikle' and Welter Silva.

References

[1] R. Kurzweil. *The Age of Spiritual Machines*. New York: Penguin 1999.

[2] R. Penrose. *Shadows of the Mind*. Oxford: Oxford University Press, 1996.

[3] M. Hutter. *Universal Artificial Intelligence*. New York: Springer, 2004.

[4] R. Merkle. "How Many Bytes in Human Memory?" *Foresight Uptate,* No. 4, October 1998

[5] E. Baum. *What Is Thought?* Cambridge MA: MIT Press 2004.

[6] R. Kurzweil. *The Singularity is Near*. New York: Penguin. 2005.

[7] B. Goertzel. *The Foundations of Emergent Digital Intelligence*. In preparation.

[8] B. Goertzel, I. F. Goertzel, M. Ikle', A. Heljakka. *Probabilistic Term Logic*. In preparation.

[9] B. Goertzel, C. Pennachin, A. Senna, T. Maia, G. Lamacie. Novamente: An Integrative Architecture for Artificial General Intelligence, *Proceedings of IJCAI 2003 Workshop on Cognitive Modeling of Agents.* Acapulco, Mexico, 2004

[10] M. Looks, B. Goertzel and C. Pennachin. Novamente: An Integrative Architecture for Artificial General Intelligence, *Proceedings of AAAI 2004 Symposium on Achieving Human-Level AI via Integrated Systems and Research,*, Washington DC, 2004

[11] B. Goertzel and Cassio Pennachin. The Novamente AI Engine. In *Artificial General Intelligence*, Ed. by B. Goertzel and C. Pennachin, New York: Springer Verlag 2005.

[12] P. Wang. *The Logic of Intelligence*. Unpublished manuscript, online at http://www.cogsci.indiana.edu/farg/peiwang/papers.html

[13] P. Wang. *Non-Axiomatic Reasoning System.* PhD Thesis, Indiana University, 1995

[14] P. Wang. Experience-grounded semantics: a theory for intelligent systems. *Cognitive Systems Research*, Vol. 6, No. 4, Pages 282-302, 2005.

[15] J. Weng. Developmental Robotics: Theory and Experiments". *International Journal of Humanoid Robotics*, vol. 1, no. 2, 2004.

[16] N. Cassimatis. Grammatical Processing Using the Mechanisms of Physical Inferences. In *Proceedings of the Twentieth-Sixth Annual Conference of the Cognitive Science Society*, 2004.

[17] R. Hecht-Nielsen Mechanization of Cognition. In *Biomimetics* (ed. Bar-Cohen, Y.) CRC Press, Boca Raton, FL, 2005

[18] J. Santore and S. Shapiro. Crystal Cassie: Use of a 3-D Gaming Environment for a Cognitive Agent. In R. Sun, Ed., *Papers of the IJCAI 2003 Workshop on Cognitive Modeling of Agents and Multi-Agent Interactions*, IJCAII, Acapulco, Mexico, August 9, 2003, 84-91.

[19] C.,Peirce. *Collected Works, Vol. 8: Scientific Metaphysics.* Cambridge MA: Harvard Press, 1935

[20] D. Dennett. *Brainstorms.* Cambridge MA: MIT Press 1991

[21] B. Whorf. *Language, Thought and Reality*. Cambridge MA: MIT Press 1964

[22] G. Bateson. *Mind and Natur*e. New York: Bantam 1980.

[23] B. Goertzel. *The Structure of Intelligence*. New York: Springer-Verlag 1993

[24] B. Goertzel. *The Evolving Mind.* New York: Gordon and Breach 1993

[25] B. Goertzel. *Chaotic Logic*. New York: Plenum 1994

[26] B. Goertzel. *From Complexity to Creativity*. New York: Plenum 1997

[27] B. Goertzel. *Creating Internet Intelligence*. New York: Plenum 2001

[28] B. Goertzel. *The Hidden Pattern.* In preparation.

[29] G. Chaitin. *Algorithmic Information Theory*. New York: Addison-Wesley 1986

[30] E. Hutchins. *Cognition in the Wild*. Cambridge MA: MIT Press 1996

[31] B. Bollobas. *Modern Graph Theory*. New York: Springer 1998

[32] B. Goertzel, S. Bugaj, C. Hartley, M. Ross and K. Silverman. The Baby Webmind Project. *Proceedings of AISB 2000*, London, England. 2000.

[33] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. New York: Chapman and Hall 1991

[34] M. Looks, B. Goertzel, snd C. Pennachin, "Learning Computer Programs with the Bayesian Optimization Algorithm," *Proceedings of Genetic and Evolutionary Computation Conference (GECCO)*, Washington DC, 2005

[35] M. Looks. *Learning Computer Programs with the Bayesian Optimization Algorithm*, MS Thesis, Washington University in St. Louis, 2005

[36] Koza, John. *Genetic Programming.* New York: MIT Press. 1993.

[37] M. Pelikan. *Bayesian Optimization Algorithm: From Single-Level to Hierarchy*. PhD Thesis, University of Illinois at Urbana-Champaign

[38] R. Sutton and A. Barto. *Reinforcement Learning*. Cambridge MA: MIT Press 1998

[39] AGI-SIM open-source software project, based at http://sourceforge.net/projects/agisim/

[40] B. Goertzel. *Post-Embodied AI.* Frontier Number 4, 2003 (http:/frontiernumber4.com)

[41] S. Opper and H. Ginsburg. *Piaget's Theory of Development*. New York Prentice-Hall 1987

[42] P. Miller. Theories of Developmental Psychology. New York: Worth Publishers 2001

[43] E. Thelen and E. Bates, E. Connectionism and dynamic systems: Are they really different*?* Introduction to J. Spencer & E. Thelen (Eds.), *Connectionism and dynamic systems. Special Section, Developmental Science*, 6(4), 378–391., 2003