NOVAMENTE: An Integrative Architecture for Artificial General Intelligence

Ben Goertzel
Cassio Pennachin
Andre Senna
Thiago Maia
Guilherme Lamacie
Artificial General Intelligence Research Institute
4005 Delancy Drive
Silver Spring MD 20906 USA
{ben,cassio}@agiri.org

Abstract

The *Novamente AI Engine*, a novel AI software system, is briefly reviewed. Unlike the majority of contemporary AI projects, Novamente is aimed at artificial *general* intelligence, rather than being restricted by design to one particular application domain, or to a narrow range of cognitive functions. Novamente integrates aspects of many prior AI projects and paradigms, including symbolic, neural-network, evolutionary programming and reinforcement learning approaches; but its overall architecture is unique, drawing on system-theoretic ideas regarding complex mental dynamics and associated emergent patterns.

1 Introduction

We describe here an in-development AI software system that confronts the "grand problem of artificial intelligence": Artificial General Intelligence (AGI). This software system is the *Novamente AI Engine*, or more compactly *Novamente*.

The Novamente design incorporates aspects of many previous AI paradigms such as genetic programming, neural networks, agent systems, evolutionary programming, reinforcement learning, and probabilistic reasoning. However, it is unique in its overall architecture, which confronts the problem of creating a holistic digital mind in a direct way that has not been done before.

The fundamental principles underlying the system design derive from a novel complex-systems-based theory of mind called the "psynet model", which was developed in a series of cross-disciplinary research treatises published during 1993-2001 [Goertzel, 1993b; 1993a; 1994; 1997; 2001]. What the psynet model has led us to is not a conventional AI program, nor a conventional multi-agent-system framework. Rather, Novamente aims to be an autonomous, self-organizing, self-evolving AGI system, with its own understanding of the world, and the ability to relate to humans on a "mind-to-mind" rather than a "software-program-to-mind" level. The Novamente project is based on many of the same ideas that underlay the Webmind AI Engine project carried out at Webmind Inc. during 1997-2001 [Goertzel *et al.*, 2000]; and it also draws to some extent on ideas from Pei Wang's Non-Axiomatic Reasoning System (NARS) [Wang, 1995].

At the moment, Novamente is partially implemented as a C++ software system, currently customized for Linux clusters, with a few externally-facing components written in Java. The overall mathematical and conceptual design of the system is described in a forthcoming paper [Goertzel and Pennachin, 2003b] and book [Goertzel and Pennachin, 2003a]. While the implementation is not yet complete, the design has matured throughout the years, and draws upon the many lessons learned by the authors in the design, implementation and testing of the Webmind AI Engine. The current, partially-complete codebase is being used by the startup firm Biomind LLC, to analyze genetics and proteomics data in the context of information integrated from numerous biological databases. Once the system is fully engineered, the project will begin a phase of interactively teaching the Novamente system how to respond to user queries, and how to usefully analyze and organize data. The end result of this teaching process will be an autonomous AGI system, oriented toward assisting humans in collectively solving pragmatic problems.

2 What is Integrative Artificial General Intelligence?

Our concept of AGI is motivated by the definition of intelligence given in [Goertzel, 1993b] – which, in verbal form, states simply that:

> *General intelligence is the ability to achieve complex goals in complex environments*

This definition ties in closely with the algorithmic information theory to general intelligence, as represented for example in the work of Marcus Hutter [Hutter, 2001].

A primary aspect of this definition is the plurality of the words *goals* and *environments*. A single narrow goal is not enough, and a single narrow environment is not enough. A chess-playing program is not a general intelligence, nor is a datamining engine that does nothing but seek for patterns in consumer information databases, and nor is a program that can extremely cleverly manipulate the multiple facets of a researcher-constructed microworld. A general intelligence must be able to carry out a variety of different tasks in a variety of different contexts, generalizing knowledge from one context to another, and building up a context and task independent pragmatic understanding of itself and the world.

Given this overall goal, three basic approaches to AGI are possible:

- Close emulation of the human brain in software
- Conception of a novel AGI architecture, highly distinct from the brain and also from narrow AI programs
- An integrative approach, synthesizing narrow AI algorithms and structures in a unique overall framework, perhaps guided to some extent by understanding of the human brain

The Novamente approach falls on the continuum between Category 3 and Category 2. Roughly two-thirds of the Novamente design is based on existing narrow AI approaches, and the rest was conceived de novo with AGI in mind. Novamente does not fall into Category 1, although it does incorporate some inspiration from study of human brain structure and dynamics.

Whenever appropriate, we have drawn upon existing techniques, which were chosen based on their applicability across multiple domains, and the ease of integration with other existing or novel components of the Novamente design. The most important examples of existing approaches that have made their way into the Novamente design are:

Genetic Algorithms and other Evolutionary Approaches which are used as a learning and optimization tool in several components of the design, due to their broad applicability, relative flexibility of representation (as exemplified by Genetic Programming [Koza, 1992] and Gene Expression Programming [Ferreira, 2001], for example), and conceptual compatibility with the theoretical framework behind Novamente (see [Goertzel, 1993a]).

Neural Networks The activation spreading algorithms in Novamente are heavily influenced by some nonlinear models found in associative neural networks [Amit, 1992]. Activation spreading helps determine the AI's *attentional focus*, i.e., the set of concept and entities which receive computational time and space at any given moment.

Combinatory Logic which allows the representation of complex higher order expressions in a conveniently variable-free format [Curry and Feys, 1958]. The elimination of variables is very useful for inference and schema/procedure/predicate learning.

Also, Novamente's internal inference engine, although novel, draws from Term Logic and Probability theory. The NLP model we use is also new, but feature-structure parsing, with some key modifications, is an important building block.

In principle, integrative AI could be conducted in two ways:

- Loose integration, in which different narrow AI techniques reside in separate software processes or software modules, and exchange the results of their analysis with each other
- Tight integration, in which multiple narrow AI processes interact in real-time on the same evolving integrative data store, and dynamically affect one another's parameters and control schemata

Novamente is based on a distributed software architecture, in which a distributed processing framework called DINI (Distributed Integrative Intelligence) is used to bind together databases, information-gathering processes, user interfaces, and "analytical clusters" consisting of tightly-integrated AI processes. The tight integration aspect is critical. Tight integration is more difficult to design, implement, test and tune, but provides the opportunity for greater intelligence via emergent, cooperative effects.

Tight integration also means that the knowledge representation has to be unified and amenable for processing by the multiple narrow AI processes. This requirement is met by Novamente's Atomspaces, but it also creates a constraint in the selection of narrow AI techniques. As an example, we use a variation of Genetic Programming for supervised learning of categories in biological data, even though GP isn't known to outperform run-of-the-mill Machine Learning alternatives. The key point is that GP creates models that are represented as combinator expressions. These models can then be processed by the probabilistic inference engine, which can refine them or create more understandable explanations of the decisions taken by the categorization system.

3 Experiential Interactive Learning

While we are avid strong AI boosters, we also believe that software and mathematics alone, no matter how advanced, cannot create an AGI. What we suggest they can do, however, is to set up a framework within which artificial general intelligence can emerge through interaction with humans in the context of a rich stream of real-world data. That is:

> *Intelligence most naturally emerges through situated and social experience.*

This leads us to the concepts of autonomy, experiential interactive learning, and goal-oriented self-modification – concepts that lie right at the heart of the notion of Artificial General Intelligence.

The Novamente system, once fully engineered, will gain its intelligence through processing practically-relevant data, answering humans' questions about this data, and providing humans with reports summarizing patterns it has observed. In addition to EIL through interactive data analysis/management, we have created a special *EIL user interface* called `ShapeWorld`, which involves interacting with Novamente in the context of a simple drawing panel on which the human teacher and Novamente may draw shapes and talk about what they're doing and what they see. Novamente will continually modify not only its knowledge base, but its control schemata based on what it has learned from its environment and the humans it interacts with. Ultimately it may even be able to modify the source code that makes it what it is.

While touching on these somewhat philosophical considerations, we feel it is important to emphasize that Novamente was not designed based on engineering and mathematical considerations alone. Rather, Novamente owes its ultimate origin to a conceptual, system-theoretic psychological model, the psynet model of mind. Based on the premise that a mind is the set of patterns in a brain, the psynet model describes a specific set of high-level structures and dynamics for mind-patterns, and proposes that these are essential to any sort of mind, human or digital. Chief among these are the dual network (a combined hierarchical/heterarchical control/information structure) and the fractal self (a portion of an AI system's knowledge network that resembles the whole). These are not structures that can be programmed into a system; rather they are structures that must emerge through the situated evolution of a system – through experiential interactive learning. Novamente's specific structures and dynamics tie in closely with the more general ones posited by the psynet model.

4 The Psynet Model of Mind

The psynet model of mind is the conceptual and philosophical foundation for the Novamente system. The psynet model does not aim to fully capture the human-language notion of "mind." Rather, it aims to capture a useful subset of that notion, with a view toward guiding AGI engineering and the analysis of human cognition.

The psynet model is based on what Ray Kurzweil calls a "patternist" philosophy [Kurzweil, 2000]. It rests on the assumption that a mind is neither a physical system, nor completely separate from the physical – rather, a mind is something associated with the set of *patterns* in a physical system. In the case of an intelligent computational system, the mind of the system is not in the source code, but rather in the patterns observable in the dynamic trace that the system creates over time in RAM and in the registers of computer processors.

The concept of pattern used here is a rigorous one, which may be grounded mathematically in terms of algorithmic information theory [Goertzel, 1997; Chaitin, 1988]. In essence, a pattern in an entity is considered as an abstract computer program that is smaller than the entity, and can rapidly compute the entity. For instance, a pattern in a picture of the Mandelbrot set, might be a program that could compute the picture from a formula. Saying "mind is pattern" is thus tantamount to positioning mind in the mathematical domain of abstract, nonphysical computer programs. As cautioned above, we are not asserting this as a complete explanation of all aspects of the concept of "mind" – but merely as a pragmatic definition that allows us to draw inferences about the minds of AGI systems in a useful way.

The "mind is pattern" approach to AI theory is not in itself original; similar ideas can be found in the thinking of contemporary philosophers such as Gregory Bateson [Bateson, 2002], Douglas Hofstadter [Hofstadter, 1996] and Daniel Dennett [Dennett, 1998]. The psynet model, however, takes the next step and asks how the set of patterns comprising a mind is structured, and how it evolves over time. It seeks to understand mind in terms of pattern dynamics, and the emergent structures arising from pattern dynamics.

According to the psynet model, the patterns constituting a mind function as semi-autonomous "actors," which interact with each other in a variety of ways. Mental functions like perception, action, reasoning and procedure learning are described in terms of interactions between mind-actors (which are patterns in some underlying physical substrate, e.g. a brain or a computer program). And hypotheses are made regarding the large-scale structure and dynamics of the network of mind-patterns.

Consistent with the "complex goals in complex environments" characterization of intelligence, an intelligent system, at a given interval of time, is assumed to have a certain *goal system* (which may be expressed explicitly and/or implicitly in the system's mind[1]). This goal system may alter over time, either through "goal drift" or through the system's concerted activity (some goals may explicitly encourage their own modification). It is important that an intelligent system has both general and specific goals in its goal system. Furthermore, one particular general goal is posited as critical to the goal system of any intelligent system: *the creation and recognition of new patterns*. With this goal in its goal system, an intelligence will seek to perceive and create new structures in itself, as it goes about the business of achieving its other goals; and this self-perception/creation will enhance its intelligence in the long term.

The pattern dynamics of a cognitive system is understood to be governed by two main "forces": spontaneous self-organization and goal-oriented behavior.

More specifically, several primary dynamical principles are posited, including:

Association in which patterns, when given attention, spread some of this attention to other patterns that they have previously been associated with in some way.

Differential attention allocation in which patterns that have been valuable for goal-achievement are given more attention, and are encouraged to participate in giving rise to new patterns.

Pattern creation in which patterns that have been valuable for goal-achievement are mutated to yield new patterns, and are combined with each other to yield new patterns.

Relationship reification in which habitual patterns in the system that are found valuable for goal-achievement, are explicitly reinforced and made more habitual.

For example, it is proposed that, for a system to display significant intelligence, the network of patterns observable in the system must give rise to several large-scale emergent structures:

Hierarchical network in which patterns are habitually in relations of control over other patterns that represent more specialized aspects of themselves.

[1] Parenthetically, it is important that a goal set be defined over an interval of time rather than a single point of time; otherwise the definition of "implicit goal sets" is more difficult.

Heterarchical network in which the system retains a memory of which patterns have previously been associated with each other in any way.

Dual Network in which hierarchical and heterarchical structures are combined, the dynamics of the two structures working together harmoniously.

"Self" structure, in which a portion of the network of patterns forms into an approximate (fractal) image of the overall network of patterns.

The psynet model is a very general construct. It does not tell you how to build an AGI system in the engineering sense; it only tells you, in general terms, "what an AGI system should be like." Novamente is the third AGI-oriented software system created with the psynet model in mind, and it is very different from the previous two efforts.

The relation between the psynet model of mind and Novamente is somewhat like the relationship between evolutionary theory and contemporary evolutionary programming algorithms. Evolutionary theory provides the conceptual underpinnings for evolutionary programming, and the first evolutionary programming algorithm, the traditional bit string GA [Goldberg, 1989], arose as a fairly direct attempt to emulate biological evolution by natural selection [Holland, 1975]. But contemporary evolutionary programming approaches such as the Bayesian Optimization Algorithm [Pelikan, 2002] and Genetic Programming [Koza, 1992] achieve superior pragmatic functionality by deviating fairly far from the biological model, and are only indirectly mappable back into their conceptual inspiration. Similarly, Novamente represents the basic concepts involved in the psynet model, but in an indirect form that owes equally much to issues of pragmatic functionality in a contemporary computing context.

5 The Novamente AGI Design

In this section we will describe the Novamente AI system, painting the design in very broad strokes, aiming only to get across the general gist.

Below we briefly describe the major aspects of Novamente design:

Nodes Nodes may symbolize entities in the external world, they may embody simple executable processes, they may symbolize abstract concepts, or they may serve as components in relationship-webs signifying complex concepts or procedures.

Links Links may be n-ary, and may point to nodes or links; they embody various types of relationships between concepts, percepts or actions. The network of links is a web of relationships.

Mind OS The Mind OS, living within the DINI framework, enables diverse MindAgents to act efficiently on large populations of Nodes and Links distributed across multiple machines.

Maps A map represents declarative or procedural knowledge, as a pattern distributed across many Nodes and Links.

MindAgents A MindAgent is a software object embodying a dynamical process such as activation spreading or first-order logical inference. It acts directly on individual Atoms, but is intended to induce and guide system-wide dynamical patterns.

Units A Unit is a collection of Nodes, Links and MindAgents, living on a cluster of machines, collectively devoted to carrying out a particular function such as: vision processing, language generation, highly-focused concentration, etc.

5.1 Knowledge Representation, Atoms, Nodes and Links

Novamente's knowledge representation scheme integrates symbolic and subsymbolic techniques, using two levels of representation:

Atoms software objects that come in two species: Nodes or Links. Nodes represent entities that stand on their own, and Links represent relationships between Atoms. Links involving only Nodes correspond to *First-order Relationships*, while links involving other Links represent *Higher-order Relationships*.

Maps sets of Atoms that tend to be activated together, or tend to be activated according to a certain pattern (e.g. an oscillation, or a strange attractor).

Generally speaking the same types of knowledge are represented on the Atom level and on the Map level. Atom level representation is more precise and more reliable, but Map level representation is more amenable to certain types of learning, and certain types of real-time behavior.

On the Atom level, the essential mathematical structure of Novamente's knowledge representation is that of a hypergraph (a graph whose edges can span more than 2 nodes), which is called the Atomspace. The Atomspace is a hypergraph with the special properties that:

- odes and links are weighted with complex weight structures (TruthValue and AttentionValue objects).
- nodes and links are labeled with different *type* labels.
- some of the nodes can contain data objects (characters, numbers, color values, etc.)
- some of the nodes can contain small hypergraphs internally.

Conceptually, the two weight structures associated with Novamente Atoms involved represent the two primary schools of AI research – logic (TruthValue) and neural networks (AttentionValue). A more detailed description is as follows:

Truth value indicates, roughly, the degree to which an Atom correctly describes the world. The object contains:

- a probability (*strength*) value;
- a *count* value indicating the amount of evidence used to derive the probability;
- optionally further information such as a probability distribution function.

Attention value a bundle of information telling how much attention of various kinds an Atom should get and is getting. This includes:

- Activation, which changes rapidly as in a neural network;
- Long-Term-Importance (LTI), an estimate of the value of keeping the Atom in memory instead of paging it to disk;
- Recent Utility, a measure of how much value has been obtained from processing the Atom recently;
- Importance, a measure of how much CPU time the Atom should get, which is computed based on activation, LTI, and recent utility.

This special hypergraph is used in many different ways. For instance:

- some of the nodes and links are intended to be interpreted logically, using probabilistic logic, and are used for inferences, including the creation of new logical Atoms which connect disjoint or loosely connected Atom-sets – this property is useful for knowlegde integration, and the amalgamation of heterogeneous data sources.
- all nodes and links can be used as neural net nodes and links, for activation spreading and related dynamics.
- some nodes and links can be interpreted as executable programs, storing Novamente's perception and action layers and, therefore, enabling and embodying communications with external agents (users, other AIs) and Novamente's digital environment.

The flexibility and high expression power of the Atomspace are invaluable in allowing the seamless integration of the multiple narrow AI algorithms, allowing Novamente's multiple cognitive modules and agents to cooperate, giving rise to the necessary emergent behavior that leads to intelligence.

Enabling all these interpretations simultaneously requires some care. For instance, the TruthValue object involves a strength value scaled into $[0, 1]$ for probabilistic interpretation by reasoning processes; but neural-net-like activation spreading processes transform these numbers into $[-1, 1]$ so they can do the inhibitory activation spreading required for complex map formation.

The network of maps in the system is also conceivable as a hypergraph – one whose nodes are maps, and whose links are "virtual links" defined as the bundles of links of a certain type going between Atoms in one map and Atoms in another. Emergent map-level links represent a more general, diffuse kind of knowledge, which interacts via Atom-level knowledge via a complex set of feedback effects.

The Atomspace-based representation described above is very flexible. It allows one to create multiple kinds of Nodes and Links, representing the different types of information in the mind. While this specialization is not necessary in theory, and may sound like a conceptual compromise to some, it proves invaluable in practice – the right degree of specialization saves us both space (encapsulating otherwise complex maps in single Atoms) and time (allowing simpler processing of these complex concepts).

Getting the right level of specialization is tricky, and the current design decisions were arrived at after a number of less successful alternatives were investigated. One needs special Node types to deal with perception, action, cognition, learning, and internal representation of the world. One needs links to represent logical and set relations, to aid learning of perceptors and actions, and to drive activation spreading and attention allocation. Tables 1 and 2 list the existing Node and Link varieties, respectively.

Node Variety	**Description**
Perceptual Nodes	These correspond to particular perceived items, like WordInstanceNode, CharacterInstanceNode, NumberInstanceNode, PixelInstanceNode.
Procedure Nodes	These contain small programs called "schema," and are called SchemaInstanceNodes. Action nodes that carry out logical evaluations are called PredicateNodes.
ConceptNodes	These represent categories of perceptual or action or conceptual nodes, or portions of maps representing such categories.
Procedure-Concept Nodes	These, called SchemaNode and PredicateNode, are ConceptNodes that are linked to particular SchemaInstanceNodes or PredicateInstanceNodes. They are used to represent complex patterns or procedures.
Psyche Nodes	These are GoalNodes and FeelingNodes, which play a special role in overall system control, in terms of monitoring system health, and orienting overall system behavior.

Table 1: Novamente Node Varieties

5.2 The Mind OS

The management of the tightly integrated AI processes involved in Novamente is handled by a software component that we call the Mind OS, or simply *the core*. The core may be considered as a generic C++ server-side framework for multi-agent systems, optimized for complex and intensive tasks involving massive agent cooperation. As we know from experience, efficiency is often a problem with multi-agent frameworks; but in designing and implementing the core, great care has been taken to avoid such issues.

Living within the larger distributed processing framework of the DINI architecture, the Mind OS is a specialized distributed processing system designed to operate across a cluster of tightly-coupled machines, in such a way that a node living on one machine in the cluster may have links relating it to nodes living on other machines in the cluster. In DINI, the Mind OS is intended to live inside a complex analytic cluster. A complex Novamente configuration will involve mul-

Link Variety	Description
Logical Links	These represent symmetric or asymmetric logical relationships among nodes (InheritanceLink, SimilarityLink) or among links and PredicateNodes (e.g. ImplicationLink, EquivalenceLink).
MemberLink	These denote fuzzy set membership
Associative Links	These denote generic relatedness, including HebbianLink learned via Hebbian learning, and a simple AssociativeLink representing relationships derived from natural language or from databases.
ApplicationLink	These indicate input-output relationships among SchemaInstanceNodes and PredicateInstanceNodes.
Action-Concept Links	Called ExecutionLinks and EvaluationLinks, these form a conceptual record of the actions taken by SchemaNodes or PredicateNodes.
ListLink and ConcatListLink	These represent internally-created or externally-observed lists, respectively.

Table 2: Novamente Link Varieties

tiple functionally-specialized analytic clusters, each one running the Mind OS.

On each machine in a Mind OS-controlled cluster, there is a table of Atoms (an AtomTable object, which comes with a collection of specialized indices for rapid lookup), and then a circular priority-aware queue of objects called MindAgents. The MindAgents are cycled through, and when one gets its turn to act, it acts for a brief period and then cedes the CPU to the next MindAgent in the queue. Most of the MindAgents embody cognitive processes, but some embody system-level processes, like periodically caching the AtomTable to disk, polling for external input (such as input from a UI), or gathering system statistics. On an SMP machine, the Mind OS may allocate different MindAgents to the different processors concurrently, obeying a fixed table of exclusion relationships in doing so.

Currently, communication with a Mind OS can be done in two ways:

- A customized Unix shell called the `nmshell` (appropriate for interactive communication, and submission of control commands)
- XML (a preferred method for loading in large amounts of data), through a specific DTD.

A third communication medium, via a Novamente-specific functional-logical programming language called *Sasha*, has been designed but not implemented. There is also a Novamente knowledge encoding language called *KNOW*, inherited from the Webmind system, which interacts with the core indirectly via XML. Existing enterprise applications also use the XML for management of the analytical clusters.

5.3 Maps and Map Dynamics

Many Atoms are significant and valuable in themselves, but some gain meaning only via their coordinated activity involving other Atoms, i.e. their involvement in *maps*. Maps come in many shapes and sizes; a general characterization of Novamente maps would be difficult to come by. However, a few rough map categories may be useful for understanding Novamente on the map level, in a very general way. These are given in Table 3.

Map Variety	Description
Concept map	A map consisting primarily of conceptual nodes.
Percept map	A map consisting primarily of perceptual nodes, which arises habitually when the system is presented with environmental stimuli of a certain sort.
Schema/Predicate map	A distributed schema or predicate.
Memory map	A map consisting largely of nodes denoting specific entities (hence related via MemberLinks and their kin to more abstract nodes) and their relationships.
Event map	A map containing many links denoting temporal relationships.
Feeling map	A map containing FeelingNodes as a significant component.
Goal map	A map containing GoalNodes as a significant component.

Table 3: Example Novamente Map Varieties

Combinations of these basic varieties will often exist. One can have a *Concept-percept map*, which contains both conceptual and perceptual Atoms. *Concept-schema maps* contain combinations of conceptual Atoms and schema. *Concept-percepct-schema maps* contain Atoms representing perceptions, actions and cognition.

The nonlinear dynamics [Devaney, 1989] of Novamente maps is a large and important subject.

Many concept maps will correspond to fixed point map attractors – meaning that they are sets of Atoms which, once they become important, will tend to stay important for a while due to mutual reinforcement. However, some concept maps may correspond to more complex map dynamic patterns. And event maps may sometimes manifest a dynamical pattern imitating the event they represent. This kind of knowledge representation is well-known in the attractor neural networks literature.

Schemata, on the other hand, generally correspond to transient maps. An individual SchemaNode does not necessarily represent an entire cognitive procedure of any significance – it may do so, especially in the case of a large encapsulated schema; but more often it will be part of a distributed schema. A distributed schema is a kind of mind map, and its map dynamic pattern is simply the system behavior that ensues when it is executed – behavior that may go beyond the actions explicitly embodied in the SchemaInstanceNodes contained in the distributed schema.

The maps in the system build up to form larger and more complex maps, ultimately yielding very large-scale emergent patterns, including patterns like the *dual network* (a specific statistical arrangement of logical links, associative links, and ApplicationLinks) and the *self* (a fractal pattern in which a subnetwork of the hypergraph comes to resemble the hypergraph itself), which are posited in the psynet model of mind.

5.4 MindAgents

The crux of Novamente intelligence lies in the MindAgents, which dynamically update the Atoms in the system on an ongoing basis. Regardless of what inputs are coming into the system or what demands are placed upon it, the MindAgents keep on working, analyzing the information in the system and creating new information based on it.

There is a number of "system maintenance" MindAgents, dealing with things like collecting system statistics, caching Atoms to disk periodically, updating caches related to distributed processing, handling queues of queries from users and other machines in the same analytic cluster or other Novamente analytic clusters. We will not discuss these further here, but will restrict ourselves to the "cognitive MindAgents" that work by modifying the AtomTable.

These are numerous. Logical groupings would give us agents that perform logical Atom creation (through first-order inference, higher-order inference, or other methods); learning agents, which create hierarchical entities like categories or clusters, and also learn new schema for perception, action, and cognitive control; control agents which focus the mind's attention, forget old, useless Atoms, and adaptively control the mind's parameters; agents that bridge the Atom and Map levels, encapsulating Maps or expanding Atoms; and others.

Tables 4 and 5 briefly mention a few existing and designed but not implemented/tuned MindAgents.

MindAgent	Description
First-Order Inference	Acts on first-order logical links, producing new logical links from old using the formulas of Probabilistic Term Logic.
Logical Link Mining	Creates logical links out of nonlogical links.
Evolutionary Predicate Learning	Creates PredicateNodes containing predicates that predict membership in ConceptNodes.
Clustering	Creates ConceptNodes representing clusters of existing ConceptNodes (thus enabling the cluster to be acted on, as a unified whole, by precise inference methods, as opposed to the less-accurate map-level dynamics).
Activation Spreading	Spreads activation among Atoms in the manner of a neural network.
Importance Updating	Updates Atom "importance" variables and other related quantities.
Concept Formation	Creates speculative, potentially interesting new ConceptNodes.
Evolutionary Optimization	A "service" MindAgent, used for schema and predicate learning, and overall optimization of system. parameters

Table 4: Existing Novamente MindAgents

5.5 Units and Functional Specialization

The Novamente MindAgents are designed to be tightly integrated, so that a large collection of MindAgents acts on a large population of Atoms in an interleaved way. But, according to the Novamente design, there is also another layer required, a layer of loose integration on top of the tightly integrated layer. A Novamente system consists of a loosely-integrated collection of *analytic clusters* or *units*, each one embodying a tightly-connected collection of AI processes, involving many different Atom types and MindAgents, and dedicated to a particular cognitive processing in a certain particular domain, or with a specific overall character.

The different analytic clusters interact via DINI; they all draw data from, and place data in, the same system-wide data warehouse. In some cases they may also query one another. The parameters of the MindAgents inside the various analytic clusters may be adapted and optimized globally.

In a complex Novamente configuration, there are many different functionally specialized analytical clusters, corresponding to functions such as: generic cognition, slower background thinking, highly focused attention, processing of particular types of sensory data, linguistic processing, or schema learning.

The simplest multi-cluster Novamente has three units, namely:

1. A *primary cognitive unit* .
2. A *background thinking unit*, containing many more nodes with only very important relationships among them, existing only to supply the primary cognitive unit with things it judges to be relevant.
3. An *AttentionalFocus unit*, containing a small number of atoms and doing very resource-intensive processing on them.

Here the specialization has to do with the intensity of processing rather than with the contents of processing.

For a Novamente to interact intensively with the outside world, it should have two dedicated clusters for each "interaction channel":

- One to contain the schemata controlling the interaction.
- One to store the "short-term-memory" relating to the interaction.

An "interaction channel" is a collection of sensory organs of some form, all perceiving roughly the same segment of external reality. Each human has only one interaction channel. But Novamente does not closely emulate either the human body or brain, and so it can easily be in this situation, interacting separately with people in different places around the world.

MindAgent	Description
Higher-order Inference	Carries out inference operations on logical links that point to links and/or PredicateNodes.
Logical Unification	Searches for Atoms that mutually satisfy a pair of PredicateNodes.
Predicate or Schema Formation	Creates speculative, potentially interesting new SchemaNodes. Also, evolutionary schema learning builds new Schema or PredicateNodes that are expected to fulfill some criteria, e.g., satisfy a given GoalNode.
Hebbian Association Formation	Builds and modifies HebbianLinks between Atoms, based on a Hebbian reinforcement learning rule.
Schema Execution	Enacts active SchemaInstanceNodes, allowing the system to carry out coordinated trains of action.
Map Encapsulation	Scans the AtomTable for patterns and creates new Atoms embodying these patterns.
Map Expansion	Takes schemata and predicates embodied in nodes, and expands them into multiple nodes and links in the AtomTable (thus transforming complex Atoms into maps of simple Atoms).
Homeostatic Parameter Adaptation	Applies evolutionary programming to adaptively tune the parameters of the system.

Table 5: Additional Novamente MindAgents

Perceptual processing like image or sound processing will best be done in specially dedicated units, with highly modality-tuned parameter values. Language processing also requires specialized units, dealing specifically with aspects of language processing such as parsing, semantic mapping, and disambiguation.

6 Applications

This section describes the current application of Novamente to bioinformatics, in a data analysis and knowledge discovery context, as well as the conversational interaction which is requireed, according to the EIL philosophy, for robust AGI development.

6.1 Novamente and Bioinformatic Pattern Mining

Now we present an example of what the current version of Novamente can do in the context of bioinformatic data analysis. The work reported here was done in the context of testing an early version of the Biomind Toolkit product for gene expression data analysis [Baldi and Hatfield, 2002]. It is described in a forthcoming journal article [Goertzel *et al.*, 2003].

What we will discuss here is one among several applications of Novamente to gene expression data analysis – "regulatory network inference", which involves studying gene expression data and recognizing the patterns that interrelate the expression levels of genes. This is a problem of the general nature of "inferring the dynamical rule of a system from samples of its trajectory."

The data presented to Novamente in this case consists of:

- A collection of "gene expression" datasets. Some of these are time series data sets, reporting the expression levels of all the genes in a genome (e.g. human, yeast) at a series of time points during the cell cycle of a single cell. Some are categorical datasets, giving the expression levels of genes in various individuals, along with data about category membership of the individuals (e.g. cancerous versus non-cancerous).
- A collection of biological background knowledge, derived from biological databases such as SGD, MIPS, BLAST, the Gene Ontology, and so forth [Letovsky, 1999].

An example rule inferred from this sort of data is a temporal pattern in the expression levels of the five specific genes that are familiar to molecular biologists and known by the labels *SIC1*, *PCL2*, *CLN3*, *ACE2*, and *SWI5*:

```
C = (LOW(SIC1) OR MOD_LOW(SIC1))
  AND (LOW(PCL2))
  AND (LOW(CLN3)) OR (MOD_LOW(CLN3))

C AND EXTRA_HIGH(SWI5)
==>
DECREASE(SWI5) AND INCREASE(ACE2)

C AND (MOD_HIGH(SWI5) OR HIGH(SWI5))
==>
INCREASE(SWI5) AND INCREASE(ACE2)
```

Table 6: Example Regulatory Network Patterns

The arrows ==> represent probabilistic logical implications (ImplicationLinks). In this case, all relations involved in a given implication refer to the same time point. The predicates *LOW*, *MOD_LOW* (moderately low), *DECREASE*, etc. are quantitatively grounded using probabilistic logic, with parameters adaptively tuned to the dataset under analysis.

In this pattern, the inferred proposition *C* gives a context, in which the dependence of the correlation between *SWI5*'s movements and *ACE2*'s movements upon the level of *SWI5* can be observed. The system is not only detecting a contextual relationship here, it is detecting a context in which a certain gene's value can serve as context for a dynamic relationship between genes. This is exactly the kind of complex interrelationship that makes genetic dynamics so subtle, and that standard data mining approaches are not capable of detecting.

The above example does not use background knowledge; it is strictly a pattern in gene expression values. The following is a simple example of a pattern involving background knowledge.

```
ConceptNode C
EquivalenceLink
 (MemberLink X C)
 (AssociativeLink
  X
  (transcriptional_regulation(CUP1)))

MOD_LOW(FKH2, t) AND LOW(MCM1, t)
AND (MOD_LOW(C) OR LOW(C))
==>
INCREASE(FKH2, t+1)
AND STABLE(MCM1, t+1)
AND INCREASE(C)
```

Table 7: Example Regulatory Network Patterns

Here the ConceptNode *C* is the category of genes that are associated with transcriptional regulation of the gene *CUP1*. The knowledge of which genes are associated with transcriptional regulation of *CUP1* comes from biological databases – not from any single database, but from integration of information from multiple databases using Novamente inference. The decision to incorporate this particular category in the rule was made by Novamente as part of its unsupervised pattern mining process.

In this particular application – like many others – the Novamente approach is significantly more effective than traditional statistics, decision trees, straightforward genetic programming based rule induction, or other traditional machine learning methods. It can find subtler patterns – both in gene expression data alone, and via judiciously incorporating knowledge from biological databases.

6.2 Conversational Interaction

According to the EIL philosophy, it is only through conversing with humans about items of mutual nonlinguistic understanding, that Novamente will be able to develop robust AGI.

Conversation implies natural language processing, which has been a traditional weak point for AI systems, mainly due to the difficulty of integrating semantic and pragmatic understanding into syntactic language analysis [Manning and Schutze, 1999]. In the Novamente framework, however, fluent NL functionality is intended to emerge along with increasing general intelligence. Novamente is designed to carry out syntactic analysis via logical unification, a process that automatically incorporates available semantic and pragmatic knowledge into its behavior. We have not yet implemented NLP in the Novamente system, but a similar approach was prototyped in the Webmind AI Engine.

To complement Novamente's internal cognitive approach to language processing, we have designed an innovative method for experiential language learning, based on interactive conversation using a combination of natural language and formal language. For example, a dialogue with an advanced Novamente system using this methodology might look something like the excerpt in table 8.

```
USER: I want information on
 Chinese-speaking residents of
 Arab nations.
NM: Mandarin, Cantonese, or both?
USER: Mandarin
NM: See the file ChineseArab.txt
USER: Thanks
NM: Would you like information
 on Arab-speaking residents of
 Chinese nations?
USER: Not now.  What I want is
 information on Chinese-speaking
 residents of Arab nations who
 send their children to schools
 in the US.
NM: Could you rephrase please?
USER:
  Find Y:
  Inheritance(X , 'Arab nation')
  resident(Y, X)
  speaks(Y, Chinese)
  child_of(Z, Y)
  attends (Z, W)
  Inheritance (W, 'school')
  part_of(X: location(W, X), 'US')
NM: By child_of(Z, Y) do you mean
 that Z is a child of Y?
USER: Yes
NM: Thanks.  See the file
 ChineseArab_1.txt
```

Table 8: Example Novamente Conversation

Here we have used a traditional predicate logic syntax for the formal-language component of the communication; we have experimented with other, less traditional syntaxes as well.

Through interactions of this nature, a Novamente system may be explicitly taught to map natural language structures into formal language structures corresponding closely to its internal nodes and links. This process may be a valuable complement to the more traditional process of grounding linguistic structures in observed nonlinguistic structures (images, database entries, etc.).

7 Conclusion

In this brief article, we have done little more than sketch the broad contours of a large and ambitious AI project.

While Novamente is currently partially engineered, the results we have obtained with the system so far are more along the lines of data mining than ambitious AGI. Given the dismal history of grand AI visions, we realize the need for caution in making grand claims for the as-yet undemonstrated abilities of our system. But we do not consider history a good reason for conservatism as to the future. We believe that by drawing on the best insights of existing AI paradigms and integrating these within a novel synthetic framework based on self-organization and experiential interactive learning, we have a serious chance of taking AI to a new level, and approaching genuine Artificial General Intelligence. And we have a precise design and plan of action for testing our hypotheses in this regard.

References

[Amit, 1992] Daniel Amit. *Modeling Brain Function*. Cambridge University Press, 1992.

[Baldi and Hatfield, 2002] Pierre Baldi and G. Wesley Hatfield. *DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling*. Cambridge University Press, 2002.

[Bateson, 2002] Gregory Bateson. *Mind and Nature: A Necessary Unity*. Hampton Books, 2002.

[Chaitin, 1988] Gregory Chaitin. *Algoritmic Information Theory*. Addison-Wesley, 1988.

[Curry and Feys, 1958] Haskell Curry and Robert Feys. *Combinatory Logic*. North-Holland, 1958.

[Dennett, 1998] Daniel Dennett. *Brainchildren: Essays on Designing Minds*. MIT Press, 1998.

[Devaney, 1989] Robert Devaney. *An Introduction to Chaotic Dynamical Systems*. Westview, 1989.

[Ferreira, 2001] Candida Ferreira. Gene Expression Programming: a New Adaptive Algorithm for Solving Problems. *Complex Systems*, 13, no 2:87–129, 2001.

[Goertzel and Pennachin, 2003a] Ben Goertzel and Cassio Pennachin. *Novamente: Design for an Artificial General Intelligence*. 2003. In preparation.

[Goertzel and Pennachin, 2003b] Ben Goertzel and Cassio Pennachin. *The Novamente AI Engine*, chapter 4. 2003. To appear.

[Goertzel *et al.*, 2000] Ben Goertzel, Ken Silverman, Cate Hartley, Stephan Bugaj, and Mike Ross. The Baby Webmind Project. In *Proceedings of AISB 00*, 2000.

[Goertzel *et al.*, 2003] Ben Goertzel, Cassio Pennachin, Andre Senna, Thiago Maia, Guilherme Lamacie, and Takuo Henmi. Inferring Genetic Regulatory Networks with Integrative AI. *Journal of Computational Biology*, 2003. Submitted for publication.

[Goertzel, 1993a] Ben Goertzel. *The Evolving Mind*. Gordon and Breach, 1993.

[Goertzel, 1993b] Ben Goertzel. *The Structure of Intelligence*. Springer-Verlag, 1993.

[Goertzel, 1994] Ben Goertzel. *Chaotic Logic*. Plenum Press, 1994.

[Goertzel, 1997] Ben Goertzel. *From Complexity to Creativity*. Plenum Press, 1997.

[Goertzel, 2001] Ben Goertzel. *Creating Internet Intelligence*. Plenum Press, 2001.

[Goldberg, 1989] David Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[Hofstadter, 1996] Douglas Hofstadter. *Mathematical Themes: Questing for the Essence of Mind and Pattern*. Basic Books, 1996.

[Holland, 1975] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[Hutter, 2001] M. Hutter. Towards a universal theory of artificial intelligence based on algorithmic probability and sequential decisions. *Proceedings of the 12^{th} Eurpean Conference on Machine Learning (ECML-2001)*, pages 226–238, 2001.

[Koza, 1992] John Koza. *Genetic Programming*. MIT Press, 1992.

[Kurzweil, 2000] Ray Kurzweil. *The Age of Spiritual Machines*. Penguin Press, 2000.

[Letovsky, 1999] Stanley Letovsky. *Bioinformatics: Databases and Systems*. Kluwer Academic, 1999.

[Manning and Schutze, 1999] Christophet Manning and Heinrich Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[Pelikan, 2002] Martin Pelikan. *Bayesian Optimization Algorithm: From Single Level to Hierarchy*. PhD thesis, University of Illinois at Urbana-Champaign, October 2002.

[Wang, 1995] Pei Wang. *Non-Axiomatic Reasoning System: Exploring the Essence of Intelligence*. PhD thesis, University of Indiana, 1995.