

# The Novamente Artificial Intelligence Engine (2007)

Tom Rochette <tom.rochette@coreteks.org>

August 30, 2025 — [861fb9d0](#)

## 0.1 Context

## 0.2 Learned in this study

## 0.3 Things to explore

# 1 Overview

## 1.1 1.2 Novamente for Knowledge Management and Data Analysis

- Novamente may be used in two complementary ways to work with unstructured text:
  - “Information retrieval” oriented, wherein a text is taken as a series of characters or a series of words, and analyzed statistically
  - Natural language processing (NLP) oriented, wherein an attempt is made to parse the sentences in the texts and extract their meanings into semantic-relationship form
- To interact with human users, the Novamente design supports a variety of different modalities, including conventional search-engine and NLP queries, web forms queries, dynamic visualization and automated report generation

## 1.2 2 Enabling Software Technologies

- Make use of an hybrid of NLP and formal languages (such as first-order logic) where one would specify the relations and constraints in his given queries
- Two categories of enabling technologies
  - Distributed software architecture
  - Database integration

## 1.3 2.2 Database Integration and Knowledge Integration

- 4 strategies to deal with databases
  - Federation: Create a common GUI for separate DBs
  - Amalgamation: Create formal mappings between the schema of different DBs
  - Schema translation: Create a new RDB combining information from multiple DBs
  - Knowledge integration: Create a translator mapping DB contents into a “universal formal knowledge representation”

## 1.4 3.1 What Is General Intelligence?

- Novamente’s goal is to create a nonhuman digital intelligent system - one that will complement human intelligence by carrying out data analysis and management tasks far beyond the capability of the human mind; and one that will cooperate with humans in a way that brings out the best aspects of both the human and the digital flavors of general intelligence
- General Intelligence is the ability to achieve complex goals in complex environments

- A system is fully generally intelligent for complexity N if it can achieve any goal of complexity N in any environment of complexity N

## 1.5 3.2 The Integrative Approach to AGI

- 3 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
- Roughly 2/3 of the Novamente design is based on existing narrow AI approaches

## 1.6 3.3 Experiential Interactive Learning and Adaptive Self-modification

- Software and mathematics alone, no matter how advanced, cannot create an AGI
- Intelligence most naturally emerges through situated and social experience
- Human brains learn to think through being taught, and through diverse social interactions
- For a software system to demonstrate AGI, it must demonstrate:
  - A coherent autonomy as an independent, self-perceiving, self-controlling system
  - The ability to modify and improve itself based on its own observations and analyses of its own performance
  - The ability to richly interact with, and learn from, other minds (such as human minds)

## 1.7 4 The Psynet Model of Mind

- The psynet model is based on what Ray Kurzweil calls a “patternist” philosophy
- A mind is neither a physical system, nor completely separate from the physical
- A mind is something associated with the set of patterns in a physical system
- A pattern in an entity is considered as an abstract computer program that is smaller than the entity, and can rapidly compute the entity
- One particular general goal is posited as critical to the goal system of any intelligent system: the creation and recognition of new patterns
- Patterns dynamics
  - Association: 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: 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: 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: Habitual patterns in the system that are found valuable for goal achievement, are explicitly reinforced and made more habitual
- Pattern networks
  - Hierarchical network: Patterns are habitually in relations of control over other patterns that represent more specialized aspects of themselves
  - Heterarchical network: The system retains a memory of which patterns have previously been associated with each other in any way
  - Dual network: Hierarchical and heterarchical structures are combined, the dynamics of the two structures working together harmoniously
  - “Self” structure: A portion of the network of patterns forms into an approximate (fractal) image of the overall network of patterns
- 3 systems built on the psynet model
  - 1994: Antimagicians: An experimental psy-net inspired program in the pure self-organizing-systems vein, with very few built-in structures and an intention for the structures and dynamics

of mind to emerge via experience. The anticipated emergence was not observed, and a more engineering-oriented approach was attempted.

- 1996-2001: The Webmind AI Engine: A large-scale Java software system that derived its software design from the psynet model in a very direct way. Parts of Webmind were successfully used in the domain of financial prediction and information retrieval. A great amount of useful prototyping was done. Directly mapping the psynet model's construct to object-oriented software structures leads to serious problems with computational efficiency.
- 2001-2009/now: Novamente: An entirely different approach, embodying a highly flexible, computationally efficient AGI framework, which could be used to implement a variety of AI systems. The framework includes 3 main aspects: the DINI architecture, the philosophy of tightly-coupled integrative AI, and the Novamente “Mind OS” architecture.

## 1.8 5 The Novamente AGI Design

- The Novamente AI Engine is a large-scale, object-oriented, multithreaded software system
- Written in C++ with a few externally-facing components written in Java
- Development primarily on Linux
- A Novamente system is a collection of analytical clusters, most of them tightly-integrated, some of them more simple and specialized
- Integrate a number of narrow AI approaches
- Here are some majors aspects of the Novamente design:
  - Nodes: May symbolize entities in the external world, embody simple executable processes, symbolize abstract concepts or serve as components in relationship-webs signifying complex concepts or procedures
  - Links: May point to nodes or links. Embody various types of relationships between concepts, percepts or actions. The network of links is a web of relationships.
  - MindAgents: A software object embodying a dynamical process such as importance updating, concept creation, or first-order logical inference. Acts directly on individual Atoms, but is intended to induce and guide system-wide dynamical patterns.
  - Mind OS: Enables diverse MindAgents to act efficiently on large populations of Nodes and Links distributed across multiple machines.
  - Maps: Declarative or procedural knowledge, as a pattern distributed across many Nodes and Links.
  - Units: A collection of Nodes, Links and MindAgents, living on a cluster of machines, collectively devoted to carrying out a particular function.

## 1.9 5.1 An Integrative Knowledge Representation

- Knowledge is represented in Novemente on two levels:
  - Atoms: Software objects that come into two species: Nodes or Links.
  - Maps: A set of Atoms that tend to be activated together, or tend to be activated according to a certain pattern.

## 1.10 5.2 The Mind OS

- Similar to an operating system:
  - Determines the task that should be executed (MindAgents)
  - Manage the duration of the MindAgent execution
- Communication with a Mind OS can be done through:
  - A shell interface
  - XML
  - A Java/J2EE middleware layer
  - (designed but not implemented) A functional-logical programming language called Sasha

- (designed but not implemented) A knowledge encoding language called NQL (Novamente Query Language)
- Mind OS does
  - Multi-threading
  - Plugging/Scheduling of heterogeneous agents
  - Distributed knowledge with local proxies and caches
  - Transaction control
  - Communication with external software agents through XML and scripts
  - Task and query processing through ticketing system
  - Adaptive parameter control
  - Dynamic, adaptive load balancing

### 1.11 5.3 Atom Types

- Data structures and dynamics have been chosen based mainly on the following criteria:
  - Demonstrated power in narrow AI applications
  - Mutual coherence as an integrative AGI framework
  - Propensity for embodying the dynamics and structures posited by the Psynet Model of Mind
- Different types of nodes
  - Perceptual: Perceived items (word, character, number, pixel)
  - Procedure: Small programs called “schema”
  - Concept: Categories of perceptual or action or conceptual nodes, or portions of maps representing such categories
  - Psyche: Goal and feeling nodes which play a special role in overall system control, in terms of monitoring system health, and orienting overall system behavior
- Different types of links
  - Logical: Symmetric or asymmetric logical relationships among nodes or among links
  - Member: Denotes fuzzy set membership
  - Associative: Denotes generic relatedness, including Hebbian learning and relationships derived from natural language or databases
  - Action-Concept: Conceptual record of the actions taken
  - List and ConcatList: Internally created or externally observed lists

### 1.12 5.4 Novamente Maps

- Some Atoms gain meaning only via their coordinated activity involving other Atoms, i.e. their involvement in “maps”
- MaximizeSatisfaction GoalNode is the center of Novamente’s motivational system
- Different types of maps found in Novamente
  - Concept map
  - Percept map
  - Schema map
  - Predicate map
  - Memory map
  - Concept-percept map
  - Concept-schema map
  - Percept-concept-schema map
  - Event map
  - Feeling map
  - Goal map

### 1.13 5.5 Mind Agents

- Responsible of updating the Atoms in the system

- There are “system maintenance” MindAgents which do things like collecting system statistics, cache Atoms to disk periodically, update caches related to distributed processing, handle queues of queries from users and other machines in the same analytic cluster or other Novamente analytic clusters
- Here is a list of different types of MindAgents
  - First-order inference: Acts on first-order logical links, producing new logical links from old links 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
  - Importance updating: Updates Atom importance variables and other related quantities using probabilistic inference
  - Concept formation: Creates speculative, potentially interesting new ConceptNodes
  - Evolutionary optimization: Used for schema and predicate learning and overall optimization of system parameters
- Here is a list of planned Novamente MindAgents
  - High-order inference
  - Logical unification
  - Predicate/Schema formation
  - Hebbian association formation
  - Evolutionary schema learning
  - Schema execution
  - Map encapsulation
  - Map expansion
  - Homeostatic parameter adaptation

## 1.14 5.6 Map Dynamics

- A map is a set of Atoms that act as a whole
- Maps may relate to other maps, just like Atoms may relate to each other
- Relationships between maps do not take the form of individual links; they take the form of bundles of links joining Atoms inside on map to the Atoms inside another (no link between maps but links between Atoms inside of each maps)

## 2 See also

## 3 References

- DOI: [10.1007/978-3-540-68677-4\\_3](https://doi.org/10.1007/978-3-540-68677-4_3)