

OPENCOG PRIME: A COGNITIVE SYNERGY BASED ARCHITECTURE FOR ARTIFICIAL GENERAL INTELLIGENCE

Ben Goertzel
Novamente LLC
ben@goertzel.org

Abstract

OpenCogPrime (OCP), a comprehensive architecture for artificial general intelligence (AGI) is briefly overviewed. Aimed in the long term at AGI at the human level and beyond, the current partial implementation of OCP is being used for applications such as controlling virtual pets in virtual worlds and inferring novel conclusions from sets of semantic relations extracted from natural language. The key aspects of OCP are described here in the context of the theoretical foundation of “cognitive synergy theory”; and the current implementation status is briefly reviewed.

Keywords: Artificial general intelligence, human-level AI, cognitive synergy, systems theory

1. INTRODUCTION

OpenCog Prime (OCP) is an architecture for advanced artificial general intelligence, based on Cognitive Synergy Theory (CST) [1] and related ideas regarding the systemic organization and dynamics of mind [2]. It is described in a 300-page online wikibook [3] that is currently being transformed into an ordinary book for publication in traditional paper form; our goal here is merely to briefly summarize the architecture and its core underlying ideas.

OCP is closely related to the Novamente Cognition Engine (NCE) which has been reviewed previously [2,4,5], and has inherited some code as well as ideas from the latter architecture. The purpose of this paper is not to discuss the differences between OCP and NCE (which are mainly low-level and technical), but rather to articulate the conceptual foundation of this family of AGI designs in the systems theory of mind (and particularly in the notion of cognitive synergy), more clearly than has been done before.

Our current plans for developing and teaching OCP based systems largely center around embodying them in online virtual worlds; however, that aspect of the project has been discussed extensively elsewhere [6,7] and will not be the focus here.

2. COGNITIVE SYNERGY THEORY

Cognitive Synergy Theory adopts a working definition of intelligence as “the ability to achieve goals in environments”, where the issue of how to weight different goals and environments is admitted as a subtle one. [8] argues that if one weights more highly the goals and environments relevant to existence in a community of embodied communicative agents, then one obtains nontrivial constraints on the sorts of systems that are likely to be highly generally intelligent given feasible computational resource restrictions.

Specifically, it is argued that this weighting leads naturally to cognitive architectures that contain multiple distinct but interacting types of memory, corresponding to the following types of knowledge: declarative, procedural, sensory, episodic, attentional and intentional.

The first four of these memory types are standard in cognitive science [9,10]. Attentional knowledge is knowledge pertaining to which entities within the system should get space and time resources at a given moment; this is closely related to the notion of consciousness [11]. Intentional knowledge refers to the system’s overall goals and derived subgoals (which may be continually revised by the system’s activity).

The essence of Cognitive Synergy Theory (CST) is the hypothesis that, in order to achieve a high level of general intelligence in the context of a community of embodied, communicative agents, a system needs to:

- Contain cognitive processes (i.e. knowledge creation mechanisms) specialized for each of the above knowledge types
- Contain methods for synergy between these processes, so that the processes specialized for each knowledge type can appeal to processes specialized for other knowledge types for aid as needed, often achieving dramatic efficiency increases as a result

CST divides cognitive processes into the two categories of analysis versus synthesis, and posits a “cognitive schematic” that models the overall goal-achieving activity of an intelligent system; these aspects will be reviewed a little later.

3. OPENCOG PRIME

Tables 1 and 2 present the key structures and processes involved in OCP; and are ideally studied together with Figure 1 from [1]. Table 1

shows the structures used in OCP for handling the key memory types discussed in CST:

Memory Type	OpenCogPrime data structure
Declarative	The AtomTable, which is a special form of weighted, labeled hypergraph -- i.e. a table of nodes and links (collectively referred to as Atoms) with different types, and each weighted with a multi-dimensional truth value (embodying an “indefinite probability” value that give both probability and confidence information [13]). See [12] for a review of the system of Atom types.
Attentional	Atoms in the AtomTable are weighted with AttentionValue objects, which contain both ShortTermImportance values (governing processor time allocation) and LongTerm Importance values (governing memory usage).
Procedural	This is handled using special “Combo” tree structures embodying LISP-like programs, in a special program dialect intended to manage behaviors in a virtual world and actions in the AtomTable
Sensory	Handled via a collection of specialized sense-modality-specific data structures
Episodic	Handled via an internal simulation world that allows the system to run “mind’s eye movies” of situations it remembers, has heard about, or hypothetically envisions.
Intentional	Goals are represented by Atoms stored in the AtomTable; there is a separate table indicating which Atoms are top-level goals, which is used to guide attention allocation and goal refinement processes

Table 1. The OpenCogPrime data structures used to represent the key knowledge types involved in Cognitive Synergy Theory

Table 2 shows the key cognitive processes considered in Cognitive Synergy Theory, and then identifies the specific OpenCogPrime algorithms that embody these

Cognitive Process	OpenCogPrime algorithm
Uncertain inference	Probabilistic Logic Networks (PLN) , a logical inference framework capable of uncertain reasoning about abstract knowledge, everyday commonsense knowledge, and low-level perceptual and motor knowledge [13]
Supervised procedure learning	MOSES , a probabilistic evolutionary learning algorithm, which learns procedures (represented as LISP-like program trees) based on specifications [14]
Attention allocation	Economic Attention Networks (ECAN) , a framework for allocating (memory and processor) attention among items of knowledge and cognitive processes, utilizing a synthesis of ideas from neural networks and artificial economics. ECAN also comes with a forgetting agent that either saves to disk or deletes knowledge that is estimated not sufficiently valuable to keep in memory. [15]
Map formation	Use of frequent subgraph mining, MOSES and other algorithms to scan the knowledge base of the system for patterns and then embodying these patterns explicitly as new knowledge items
Concept creation	A collection of heuristics for forming new concepts via combining existing ones, including conceptual blending, mutation and extensional and intensional logical operators
Simulation	The running of simulations of (remembered or imagined) external-world scenarios in an internal world-simulation engine
Goal refinement	Transformation of given goals into sets of subgoals, using concept creation, inference and procedure learning

Table 2. Key cognitive processes in Cognitive Synergy Theory, and the algorithms that play their roles in OpenCogPrime.

processes. Each of these cognitive processes deals with one or more types of memory --

declarative, procedural, sensory, episodic or attentional.

Next, CST defines an implication called the “cognitive schematic” which serves as a general formulation of an intelligent system’s basic cognitive activity:

$$\text{Context \& Procedure} \rightarrow \text{Goal } \langle p \rangle$$

This formula may be interpreted to mean “If the context C appears to hold currently, then if I enact the procedure P, I can expect to achieve the goal G with certainty p.” The system is initially supplied with a set of high-level goals such as “get rewarded by my teacher”, “learn new things” and so forth; and it then uses inference (guided by other cognitive mechanisms) to refine these initial goals into more specialized subgoals.

In the OCP context, a procedure in this schematic is a Combo tree stored in the system’s procedural knowledge base; and a context is a (fuzzy, probabilistic) logical predicate stored in the AtomTable, that holds, to a certain extent, during each interval of time. A goal is a fuzzy logical predicate that has a certain value at each interval of time, as well.

The cognitive schematic leads to a partitioning of cognitive processes into analysis versus synthesis processes. Synthesis processes aim to create new implications of the given form. Analysis processes aim to assess the truth values of implications of the given form. CST suggests that social, communicative embodied intelligence requires both analysis and synthesis processes corresponding to all the key knowledge types; and that each of these processes must appropriately synergize with processes corresponding to other knowledge types. Table 3 sums up the role of the different OCP cognitive mechanisms in terms of the CST categories of analysis and synthesis.

As mentioned earlier, attentional knowledge is handled in OCP by the ECAN artificial economics mechanism, that continually updates ShortTermImportance and LongTerm

Importance values associated with each item in the system’s memory, which control the amount of attention other cognitive mechanisms pay to the item, and how much motive the system has to keep the item in memory. HebbianLinks are then created between knowledge items that often possess ShortTermImportance at the same time; this is OCP’s version of traditional Hebbian learning.

	Synthesis	Analysis
Declarative	•Evolutionary, blending-based and logical concept creation	
PLN (Declarative & Procedural)	•PLN forward inference	•PLN backward inference
MOSES (Declarative & Procedural)	•MOSES and hillclimbing procedure learning (combining portions and features of prior procedures)	•Probabilistic modeling to identify patterns among programs fulfilling a certain goal in a certain context (part of MOSES)
Sensory/Episodic	•Imagination of hypothetical episodes based on specified criteria, via combination of aspects of known episodes	•Filling in gaps in remembered or hypothesized episodes
Attentional	•Hebbian learning •Importance spreading •Map formation	•Assignment of credit
Intentional	•Goal synthesis	•Goal refinement

Table 3. The key OpenCogPrime cognitive processes categorized according to knowledge type and process type

ECAN has deep interactions with other cognitive mechanisms as well, which are essential to its efficient operation; for instance, PLN inference may be used to help ECAN extrapolate conclusions about what is worth paying attention to, and MOSES may be used to recognize subtle attentional patterns. ECAN also handles “assignment of credit”, the figuring-out of the causes of an instance of successful goal-achievement, drawing on PLN

and MOSES as needed when the causal inference involved here becomes difficult.

The synergies between OCP’s cognitive processes are well summarized in Table 3 from [1], which is a 16x16 matrix summarizing a host of interprocess interactions generic to CST. Rather than replicating that analysis here, we present some comments more specialized to OpenCog Prime.

One key aspect of how OCP implements cognitive synergy is PLN’s sophisticated management of the confidence of judgments. This ties in with the way OpenCog’s PLN inference framework represents truth values in terms of multiple components (as opposed to the single probability values used in many probabilistic inference systems and formalisms): each item in OpenCog’s declarative memory has a confidence value associated with it, which tells how much weight the system places on its knowledge about that memory item. This assists with cognitive synergy as follows: A learning mechanism may consider itself “stuck”, generally speaking, when it has no high-confidence estimates about the next step it should take. Without reasonably accurate confidence assessment to guide it, inter-component interaction could easily lead to increased rather than decreased combinatorial explosion. And of course there is an added recursion here, in that confidence assessment is carried out partly via PLN inference, which in itself relies upon these same synergies for its effective operation.

To illustrate this point further, consider one of the synergetic aspects described in Table 3 in [1]: the role cognitive synergy plays in deductive inference. Deductive inference is a hard problem in general – but what is hard about it is not carrying out inference steps, but rather “inference control” (i.e., choosing which inference steps to carry out). Specifically, what must happen for deduction to succeed in OCP is:

1. the system must recognize when its deductive inference process is “stuck”, i.e. when the PLN inference control mechanism carrying out deduction has no clear idea regarding which inference step(s) to take next, even after considering all the domain knowledge at its disposal
2. in this case, the system must defer to another learning mechanism to gather more information about the different choices available – and the other learning mechanism chosen must, a reasonable percentage of the time, actually provide useful information that helps PLN to get “unstuck” and continue the deductive process

For instance, deduction might defer to the “attentional knowledge” subsystem, and make a judgment as to which of the many possible next deductive steps are most associated with the goal of inference and the inference steps taken so far, according to the HebbianLinks constructed by the attention allocation subsystem, based on observed associations. Or, if this fails, deduction might ask MOSES (running in supervised categorization mode) to learn predicates characterizing some of the terms involving the possible next inference steps. Once MOSES provides these new predicates, deduction can then attempt to incorporate these into its inference process, hopefully (though not necessarily) arriving at a higher-confidence next step..

4. CURRENTLY IMPLEMENTED COMPONENTS

OpenCogPrime is currently partially implemented within a broader C++ software framework called OpenCog [16], which was created during 2007-2008 with a view toward broadly fostering integrative open-source AI software development, as well as providing a software foundation for OCP. OpenCog is now

supported by a small but active developer community.

In addition to the OpenCog framework itself, some of the key OCP components already substantially implemented are:

- A version of the PLN probabilistic logical inference engine (which however has some lacunae, e.g it does not include the full PLN rules for spatiotemporal or intensional inference, and its adaptive inference control mechanisms are incomplete)
- A version of the MOSES probabilistic evolutionary program learning algorithm (which however currently handles only a subset of the space of possible programmatic constructs: it deals with loops and conditionals and Boolean and real inputs, but not yet internal variables or higher order functions)
- A version of ECAN, which is restricted to the detection and use of binary attentional patterns between Atoms; and also, a simple forgetting mechanism that removes Atoms with low long-term importance
- Proxies between OpenCog and the RealXTend and Multiverse virtual worlds, used for controlling virtual dogs and humanoids¹
- An initial goal system and procedure execution framework, used for executing learned procedures in the above virtual worlds

¹ A highly technical comment here: at the moment the software mentioned in these last two bullet points is contained in separate versions of OpenCog, called OpenPetBrain; these two versions will be integrated during the first half of 2008.

5. CONCLUSION

To thoroughly describe a comprehensive, integrative AGI architecture in a brief conference paper would be an impossible task; all we have attempted here is a brief overview, drawing heavily on the formulation of Cognitive Synergy Theory from [1] to structure the discussion. We do not expect this brief summary to be enough to convince the skeptical reader that the approach described here has a high odds of success at achieving its stated goals; but we will rate our summary successful if the reader is motivated to look more deeply by exploring [1,2,3,13] and other related references that provide more depth.

References

[1] Goertzel, Ben (2009). Cognitive Synergy: A Universal Principle of Feasible General Intelligence?, *Dynamical Psychology*

[2] Goertzel, Ben (2006). *The Hidden Pattern*. BrownWalker.

[3] Goertzel, Ben (1998). *OpenCog Prime: Design for a Human-Level AGI*. Online at OpenCog.org/OpenCogPrime

[4] Goertzel, Ben (2006). *Patterns, Hypergraphs and General Intelligence*. Proceedings of WCCI 2006, Vancouver.

[5] Goertzel, Ben, Cassio Pennachin, Andre Senna, Thiago Maia and Guilherme Lamacie (2003). *Novamente: An Integrative Architecture for Artificial General Intelligence*. Proceedings of IJCAI-03 Workshop on Agents and Cognitive Modeling, Acapulco, August 2003

[6] Goertzel, Ben . *A Pragmatic Path Toward Endowing Virtually-Embodied AIs*

with Human-Level Linguistic Capability, Special Session on Human-Level Intelligence, IEEE World Congress on Computational Intelligence (WCCI) Hong Kong, 2008

[7] Goertzel, Ben (2009). *What Must a World Be That a Humanlike AGI Might Develop In It?*, *Journal of AGI*

[8] Goertzel, Ben (2009). *The Embodied Communication Prior*. Submitted for publication.

[9] Tulving and Craik (2005). *The Oxford Handbook of Memory*. Oxford University Press.

[10] Eichenbaum, Henry (2002). *The Cognitive Neuroscience of Memory*. Oxford University Press.

[11] Metzinger, Thomas (Editor) (2005). *Neuronal Correlates of Consciousness*. MIT Press.

[12] Goertzel, Ben and Cassio Pennachin (2006). *The Novamente AI Engine*. In Goertzel and Pennachin, *Artificial General Intelligence*, Springer.

[13] Goertzel, Ben, Matt Ikle, Izabela Freire Goertzel and Ari Heljakka (2008). *Probabilistic Logic Networks*. Springer.

[14] Looks, Moshe (2006). *Competent Program Evolution*. PhD thesis, Computer Science Dept. Washington University St. Louis

[15] Goertzel, Ben, Matt Ikle and Joel Pitt (2009). *Economic Attention Networks*. Proceedings of AGI-09, Atlantis Press.

[16] Hart, David and Ben Goertzel (2008). *OpenCog: A Software Framework for Integrative Artificial General Intelligence*, in *Proceedings of the First AGI Conference*, Ed. Wang et al, IOS Press