OpenCogBot: Achieving Generally Intelligent Virtual Agent Control and Humanoid Robotics via Cognitive Synergy

Ben Goertzel[1,2] *and Hugo de Garis*[2] *and Cassio Pennachin*[1] *and Nil Geisweiller*[1] *and Samir Araujo*[1] *and Joel Pitt*[1] *and Shuo Chen*[2] *and Ruiting Lian*[2] *and Min Jiang* [2] *and Ye Yang*[2] *and Deheng Huang* [2]
[1] Novamente LLC, 1405 Bernerd Place, Rockville MD 20851, USA
[2] Artificial Brain Lab, Xiamen University, Xiamen, China
email: ben@goertzel.org

Abstract

We describe an integrative cognitive architecture for human-like, human-level, embodied general intelligence, founded on integrating the OpenCogPrime framework for cognition, language and high-level learning with a hierarchical temporal memory for lower-level perception and action. The core conceptual principle of the architecture is "cognitive synergy", wherein different components are specifically integrated in such a way as to compensate for each others scalability weaknesses. The current, preliminary implementation of the architecture to control a Nao robot is described, and contrasted with prior work using OpenCogPrime to control virtual agents in virtual worlds; and future plans for achieving greater intelligence via instructing the Nao in a "robot preschool" context are reviewed.

1 Introduction

Might it be possible, using a combination of current technologies, to create a software program enabling a robot to think, learn and converse roughly at the level of a young human child? While achievements like this have been elusive throughout the history of AI, we suspect that due to advances in AI, hardware, neuroscience and cognitive science, the answer is now: Yes, if the cognitive architecture of the program is right, and the combination of technologies is carried out according to the right theoretical principles. The goal of the research described here is to explore the hypothesis that **the cognitive synergy ensuing from integrating multiple symbolic and subsymbolic learning and memory components in an appropriate cognitive architecture and environment, can yield robust childlike intelligence**.

We describe here a cognitive architecture called **OpenCogBot**, aimed at enabling a humanoid robot to carry out a set of mentally challenging tasks in a robot lab outfitted as a "virtual preschool." The current robotic platform for the project is be the Nao humanoid robot, connected via wifi with a compute cluster running the OpenCogBot software, including standard multiprocessor servers and NVidia Tesla vector processing servers. In the OpenCogBot project, human child behavior is used not as a target to be closely imitated, but rather as a guide to formulating appropriate tasks and metrics for instructing and assessing the robot. The medium-term goal of the project is for the robot to learn, reason, create and converse at the qualitative level of a three year old human child. Longer-term, we aim at full human-adult-level intelligence, and ultimately beyond.

The OpenCogBot integrative cognitive architecture is a fusion of the OpenCog architecture for cognition and language [1] with a hierarchical temporal memory based system for perception and actuation (Itamar Arel's DeSTIN [2] is briefly discussed as a promising example of the latter).

Collectively, these systems involve sophisticated methods for visual and auditory pattern inference, language comprehension and generation, action planning, commonsense uncertain reasoning, and concept creation; and most of these capabilities have been evaluated in prior applications of narrower scope. In OpenCogBot these methods are integrated in a novel way, based on the principle of **cognitive synergy**, resulting in an overall cognitive architecture in which practical goals are achieved via multiple learning processes associated with multiple memory types cooperatively manipulating a common network-based knowledge store. While most of the components of OpenCogBot are moderately similar to those used in prior AI systems, our hypothesis is that the proper integration of these components, coupled with their usage in an appropriately stimulating context, will lead to overall system dynamics highly dissimilar to those seen in experiments with prior cognitive architectures. The architecture presented here is an evolution of those presented in the predecessor papers [3, 4], the difference being that here more specific and sophisticated assumptions are made about the perception/action system, allowing more to be said about the desired synergetic dynamics.

In this paper we describe the main ideas underlying OpenCogBot at a high level, sometimes referring to previous publications for details; and at the end we discuss our achievements in the project so far and our near-future development plans.

Approaching Embodied General Intelligence via Cognitive Synergy The human brain is an integration of an assemblage of diverse structures and dynamics, built using common components and arranged according to a sensible cognitive architecture. Its algorithms and structures have been honed by evolution to work closely together – they are very tightly inter-adapted, in the same way that the different organs of the body are adapted to work together. Due their close interoperation they give rise to the overall systemic behaviors that characterize human-like general intelligence. We believe that the main missing ingredient in AI so far – which has prevented the achievement of advanced Artificial General Intelligence (AGI) – is **cognitive synergy**: the fitting-together of different intelligent components into an appropriate cognitive architecture, in such a way that the components richly and dynamically support and assist each other, interrelating very closely in a similar manner to the components of the brain or body and thus giving rise to appropriate emergent structures and dynamics. Which leads us directly to an OpenCogBot type approach.

At the broadest level, there are four primary challenges in constructing a cognitive synergy based approach to AGI:

1. choosing an **overall cognitive architecture** that possesses adequate richness and flexibility for the task of achieving childlike cognition

2. choosing **appropriate AI algorithms and data structures** to fulfill each of the functions identified in the cognitive architecture (e.g. visual perception, audition, episodic memory, language generation, analogy,...)

3. ensuring that these algorithms and structures, within the chosen cognitive architecture, are able to cooperate in such a way as to provide appropriate **coordinated, synergetic intelligent behavior** (a critical aspect since childlike cognition is an integrated functional response to the world, rather than a loosely coupled collection of capabilities)

4. embedding one's system in an environment that provides **sufficiently rich stimuli and interactions** to enable the system to use this cooperation to ongoingly create an intelligent internal world-model and self-model

The use of OpenCogBot to control a Nao in a robot preschool provides a viable way to address these challenges.

2 OpenCogPrime: A Cognitive Synergy Based Architecture for Controlling Intelligent Virtual Agents

OpenCogPrime (OCP) is a comprehensive architecture for cognition, language, and virtual agent control, created by the PIs Goertzel and Pennachin and their collaborators during the period since 2001 (and building on their work from the 1990s). Conceptually founded on the systems theory of intelligence outlined in [5] and alluded to above, it is currently under development within the open-source OpenCog AI framework (see `http://opencog.org` and [1]). It combines multiple AI paradigms such as uncertain-logic, computational linguistics, evolutionary program learning and connectionist attention allocation in a unified cognitive-science-based architecture. Cognitive processes embodying these different paradigms interoperate together on a common neural-symbolic knowledge store called the Atomspace.

The high-level architecture of OCP – as shown in the broader OpenCogBot architecture diagram in Figure 2 – involves the use of multiple cognitive processes associated with multiple types of memory to enable an intelligent agent to execute the procedures that it believes have the best probability of working toward its goals in its current context. In a robot preschool context, for example, the top-level goals are simple things such as pleasing the teacher, learning new information and skills, and protecting the robot's body.

What OCP does not handle is low-level perception and action; OpenCogBot surmounts this problem via integrating OCP with a hierarchical temporal memory system.

Memory Types in OpenCogPrime OCP's memory types are the declarative, procedural, sensory, and episodic memory types that are widely discussed in cognitive neuroscience [6], plus attentional memory for allocating system resources generically, and intentional memory for allocating system resources in a goal-directed way. Table 1 overviews these memory types, giving key references and indicating the corresponding cognitive processes, and also indicating which of the generic patternist cognitive dynamics each cognitive process corresponds to (pattern creation, association, etc.).

In terms of the patternist cognitive theory underlying OCP [5], the multiple types of memory in OCP should be considered as specialized ways of storing particular types of pattern, optimized for spacetime efficiency. The cognitive processes associated with a certain type of memory deal with creating and recognizing patterns of the type for which the memory is specialized. While in principle all the different sorts of pattern could be handled in a unified memory and processing architecture, the sort of specialization used in OCP is necessary in order to achieve acceptable efficient general intelligence using currently available computational resources. And, efficiency is not a side-issue but rather the essence of real-world AGI (since as Hutter has shown [7], if one casts efficiency aside, arbitrary levels of general intelligence can be achieved via a trivially simple program).

The essence of the OCP design lies in the way the structures and processes associated with each type of memory are designed to work together in a closely coupled way, yielding cooperative intelligence going beyond what could be achieved by an architecture merely containing the same structures and processes in separate "black boxes."

The inter-cognitive-process interactions in OpenCog are designed so that

- conversion between different types of memory is possible, though sometimes computationally costly (e.g. an item of declarative knowledge may with some effort be interpreted procedurally or episodically, etc.)
- when a learning process concerned centrally with one type of memory encounters a situation where it learns very slowly, it can often resolve the issue by converting some of the relevant knowledge into a different type of memory: i.e. **cognitive synergy**

Goal-Oriented Dynamics in OpenCogPrime OCP's dynamics has both goal-oriented and "spontaneous" aspects; here for simplicity's sake we will focus on the goal-oriented ones. The basic

Memory Type	Specific Cognitive Processes	General Cognitive Functions
Declarative	Probabilistic Logic Networks (PLN) [8]; conceptual blending [9]	pattern creation
Procedural	MOSES (a novel probabilistic evolutionary program learning algorithm) [10]	pattern creation
Episodic	internal simulation engine [11]	association, pattern creation
Attentional	Economic Attention Networks (ECAN) [12]	association, credit assignment
Intentional	probabilistic goal hierarchy refined by PLN and ECAN, structured according to MicroPsi [13]	credit assignment, pattern creation
Sensory	In OpenCogBot, this will be supplied by the DeSTIN component	association, attention allocation, pattern creation, credit assignment

Table 1: Memory Types and Cognitive Processes in OpenCog Prime. The third column indicates the general cognitive function that each specific cognitive process carries out, according to the patternist theory of cognition.

goal-oriented dynamic of the OCP system, within which the various types of memory are utilized, is driven by implications known as "cognitive schematics", which take the form

$$Context \wedge Procedure \rightarrow Goal < p >$$

(summarized $C \wedge P \rightarrow G$). Semi-formally, this implication may 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." Cognitive synergy means that the learning processes corresponding to the different types of memory actively cooperate in figuring out what procedures will achieve the system's goals in the relevant contexts within its environment.

OCP's cognitive schematic is significantly similar to production rules in classical architectures like SOAR and ACT-R; however, there are significant differences which are important to OCP's functionality. Unlike with classical production rules systems, uncertainty is core to OCP's knowledge representation, and each OCP cognitive schematic is labeled with an uncertain truth value, which is critical to its utilization by OCP's cognitive processes. Also, in OCP, cognitive schematics may be incomplete, missing one or two of the terms, which may then be filled in by various cognitive processes (generally in an uncertain way). A stronger similarity is to MicroPsi's triplets; the differences in this case are more low-level and technical and are discussed in [14].

Finally, the biggest difference between OCPs cognitive schematics and production rules or other similar constructs, is that in OCP this level of knowledge representation is not the only important one. CLARION uses production rules for explicit knowledge representation and then uses a totally separate subsymbolic knowledge store for implicit knowledge. In OCP both explicit and implicit knowledge are stored in the same graph of nodes and links, with explicit knowledge stored in probabilistic logic based nodes and links such as cognitive schematics; and implicit knowledge stored in patterns of activity among these same nodes and links, defined via the activity of the "importance" values associated with nodes and links and propagated by the ECAN attention allocation process.

The meaning of a cognitive schematic in OCP is hence not entirely encapsulated in its explicit logical form, but resides largely in the activity patterns that ECAN causes its activation or exploration to give rise to. And this fact is important because the synergetic interactions of system components are in large part modulated by ECAN activity. Without the real-time combination of explicit and implicit knowledge in the system's knowledge graph, the synergetic interaction of

different cognitive processes would not work so smoothly, and the emergence of effective high-level hierarchical, heterarchical and self structures would be less likely.

Current and Prior Applications of OpenCog OpenCog has been used for commercial applications in the area of natural language processing and data mining; for instance, see [15] where OpenCog's PLN reasoning and RelEx language processing are combined to do automated biological hypothesis generation based on information gathered from PubMed abstracts. Most relevantly to the present proposal, has also been used to control virtual agents in virtual worlds [11], using an OpenCog variant called the OpenPetBrain (see Figure 1 for a screenshot of an OpenPetBrain-controlled virtual dog; and see `http://novamente.net/example` for some videos of these virtual dogs in action). The OpenCogBot project is a natural extension to humanoid robotics of this prior work in virtual worlds.

While the OpenCog virtual dogs do not display intelligence closely comparable to that of real dogs (or human children), they do demonstrate a variety of interesting and relevant functionalities including learning new behaviors based on imitation and reinforcement; responding to natural language commands and questions, with appropriate actions and natural language replies; and spontaneous exploration of their world, remembering their experiences and using them to bias future learning and linguistic interaction. These are simpler versions of capabilities we are working to demonstrate with the OpenCogBot system.

Figure 1: Screenshot of OpenCog-controlled virtual dog

Transitioning from Virtual Agents to a Physical Robot Via hybridizing OpenCogPrime with a dedicated (e.g. hierarchical temporal memory) system for perception and action, transitioning OCP's intelligent functionalities from the virtual-agents domain to the physical-robotics domain is expected to be a relatively straightforward process *initially*. We have already begun the task and have not yet encountered major difficulties. Achieving advanced intelligence via this integration will of course be a longer and more involved process.

A reasonable level of capability is achievable by simply interposing low-level software as a perception/action "black box" between OpenCog and a robot. The OpenCogBot experiments we have pursued so far follow this approach – connecting the OpenPetBrain to a Nao robot using

simple vision processing code and actuation scripts in the intermediary role. However, to meet the goals of the proposal we must go beyond this approach, and connect robot perception and actuation software with OpenCog in a "white box" manner that allows intimate dynamic feedback between perceptual, motoric, cognitive and linguistic functions. For instance, if the DeSTIN hierarchical temporal memory architecture is used for perception/action, then one may achieve this via the creation and real-time utilization of links between the nodes in OpenCog's and DeSTIN's internal networks. We return to this issue in Section 3 below.

Analysis and Synthesis Processes in OpenCogPrime The cognitive schematic $Context \wedge Procedure \rightarrow Goal$ leads to a conceptualization of the internal action of an intelligent system as involving two key categories of learning:

- **Analysis**: Estimating the probability p of a posited $C \wedge P \rightarrow G$ relationship
- **Synthesis**: Filling in one or two of the variables in the cognitive schematic, given assumptions regarding the remaining variables, and directed by the goal of maximizing the probability of the cognitive schematic

More specifically, where synthesis is concerned,

- The MOSES probabilistic evolutionary program learning algorithm is applied to find P, given fixed C and G. Internal simulation is also used, for the purpose of creating a simulation embodying C and seeing which P lead to the simulated achievement of G.
  - *Example: A virtual dog learns a procedure P to please its owner (the goal G) in the context C where there is a ball or stick present and the owner is saying "fetch".*
- PLN inference, acting on declarative knowledge, is used for choosing C, given fixed P and G (also incorporating sensory and episodic knowledge as appropriatel). Simulation may also be used for this purpose.
  - *Example: A virtual dog wants to achieve the goal G of getting food, and it knows that the procedure P of begging has been successful at this before, so it seeks a context C where begging can be expected to get it food. Probably this will be a context involving a friendly person.*
- PLN-based goal refinement is used to create new subgoals G to sit on the right hand side of instances of the cognitive schematic.
  - *Example: Given that a virtual dog has a goal of finding food, it may learn a subgoal of following other dogs, due to observing that other dogs are often heading toward their food.*
- Concept formation heuristics are used for choosing G and for fueling goal refinement, but especially for choosing C (via providing new candidates for C). They are also used for choosing P, via a process called "predicate schematization" that turns logical predicates (declarative knowledge) into procedures.
  - *Example: At first a virtual dog may have a hard time predicting which other dogs are going to be mean to it. But it may eventually observe common features among a number of mean dogs, and thus form its own concept of "pit bull," without anyone ever teaching it this concept explicitly.*

Where analysis is concerned:

- PLN inference, acting on declarative knowledge, is used for estimating the probability of the implication in the cognitive schematic, given fixed C, P and G. Episodic knowledge is also used this regard, via enabling estimation of the probability via simple similarity matching against past experience. Simulation is also used: multiple simulations may be run, and statistics may be captured therefrom.
  - *Example: To estimate the degree to which asking Bob for food (the procedure P is "asking for food", the context C is "being with Bob") will achieve the goal G of getting food, the virtual dog may study its memory to see what happened on previous occasions where it or other dogs asked Bob for food or other things, and then integrate the evidence from these occasions.*
- Procedural knowledge, mapped into declarative knowledge and then acted on by PLN inference, can be useful for estimating the probability of the implication $C \wedge P \rightarrow G$, in cases where the probability of $C \wedge P_1 \rightarrow G$ is known for some P_1 related to P.
  - *Example: knowledge of the internal similarity between the procedure of asking for food and the procedure of asking for toys, allows the virtual dog to reason that if asking Bob for toys has been successful, maybe asking Bob for food will be successful too.*
- Inference, acting on declarative or sensory knowledge, can be useful for estimating the probability of the implication $C \wedge P \rightarrow G$, in cases where the probability of $C_1 \wedge P \rightarrow G$ is known for some C_1 related to C.
  - *Example: if Bob and Jim have a lot of features in common, and Bob often responds positively when asked for food, then maybe Jim will too.*
- Inference can be used similarly for estimating the probability of the implication $C \wedge P \rightarrow G$, in cases where the probability of $C \wedge P \rightarrow G_1$ is known for some G_1 related to G. Concept creation can be useful indirectly in calculating these probability estimates, via providing new concepts that can be used to make useful inference trails more compact and hence easier to construct.
  - *Example: The dog may reason that because Jack likes to play, and Jack and Jill are both children, maybe Jill likes to play too. It can carry out this reasoning only if its concept creation process has invented the concept of "child" via analysis of observed data.*

In these examples we have focused on cases where two terms in the cognitive schematic are fixed and the third must be filled in; but just as often, the situation is that only one of the terms is fixed. For instance, if we fix G, sometimes the best approach will be to collectively learn C and P. This requires either a procedure learning method that works interactively with a declarative-knowledge-focused concept learning or reasoning method; or a declarative learning method that works interactively with a procedure learning method. That is, it requires the sort of cognitive synergy built into the OCP design.

3 Integrating OpenCogPrime with Hierarchical Temporal Memory

OpenCogPrime is designed to handle most of the types of knowledge important for human like intelligence, in a manner manifesting cognitive synergy, but in its current form it doesn't deal with low-level sensorimotor knowledge. It could be extended to handle such knowledge in various ways, but in the OpenCogBot architecture we have chosen a different approach: hybridizing OCP with another sort of cognitive architecture, hierarchical temporal memory, that is specifically oriented toward sensorimotor learning, and has already been extensively tested in the perception domain.

The best known hierarchical temporal memory architecture is Jeff Hawkins' HTM framework [16], but we have found Itamar Arel's DeSTIN system [2] more impressively functional and also

more robustly designed from an AGI perspective. For sake of concreteness we will speak mostly in terms of DeSTIN here, but the main points raised actually apply to any hierarchical temporal memory approach that handles perception, action and reinforcement.

The practical work with DeSTIN to date has focused on visual and auditory processing, but it has been developed with the intention of gradually extending it into a complete system for humanoid robot control. In the OpenCogBot context, however, the intention is to utilize DeSTIN not as a complete AGI architecture but rather as a collection of modules carrying out perception and actuation oriented processing, hybridizing it with OCP which will handle abstract cognition and language. Here we will discuss DeSTIN primarily in the perception context, only briefly mentioning the application to actuation which is conceptually similar.

The DeSTIN Hierarchical Temporal Memory Architecture The DeSTIN architecture comprises three interlinked hierarchies:

- a deep spatiotemporal inference network carrying out perception and world-modeling
- a similarly architected critic network that provides feedback on the inference network's performance
- an action network that controls actuators based on the activity in inference network.

The nodes in these networks perform probabilistic pattern recognition according to algorithms; and the nodes in each of the networks may receive states of nodes in the other networks as inputs, providing rich interconnectivity and synergetic dynamics.

For instance, the hierarchical architecture of DeSTIN's perception comprises an arrangement into multiple layers of "nodes" comprising multiple instantiations of an identical cortical circuit. Each node corresponds to a particular spatiotemporal region, and uses a statistical learning algorithm to characterize the sequences of patterns that are presented to it by nodes in the layer beneath it.

To understand the conceptual relationship between DeSTIN's three networks, consider the way people learn to distinguish between cups and bowls in part via hearing other people describe some objects as cups and others as bowls. To emulate this kind of learning, DeSTIN's critic network provides positive or negative reinforcement signals based on whether the action network has correctly identified a given object as a cup or a bowl, and this signal then impacts the nodes in the action network. The critic network takes a simple external "degree or success or failure" signal and turns it into multiple reinforcement signals to be fed into the multiple layers of the action network. The result is that the action network self-organizes so as to include an implicit "cup versus bowl" classifier, whose inputs are the outputs of some of the nodes in the higher levels of the perceptual network. This classifier belongs in the action network because it is part of the procedure by which the DeSTIN system carries out the action of identifying an object as a cup or a bowl.

We next describe how the two component systems of OpenCogBot may be integrated. The basic logic of the integration is depicted in Figure 2 but of course the essence of the integration lies in the dynamics, not the structure.

Pipeline Architectures for Perception and Action The simplest way to integrate DeSTIN and OCP, as noted above, is simply to have OCP treat DeSTIN in the same way as it treats a virtual agent, i.e.: first OCP sends DeSTIN high-level movement commands like "step forward", "grab the ball at coordinates $(10cm, 30cm, 120cm)$"; DeSTIN then translates these into coordinated sets of servomotor commands DeSTIN sends OCP high-level perceptual messages such as "dark blue object with label 1443 at coordinates $(15cm, 45cm, 200cm)$"; and finally OCP then creates semantic nodes in its own memory corresponding to these percepts. This amounts to OCP using DeSTIN's perception hierarchy as a pipeline from the robot, and its action hierarchy as a pipeline to the robot, and its critic hierarchy as a way of giving reinforcement signals to the other two hierarchies.

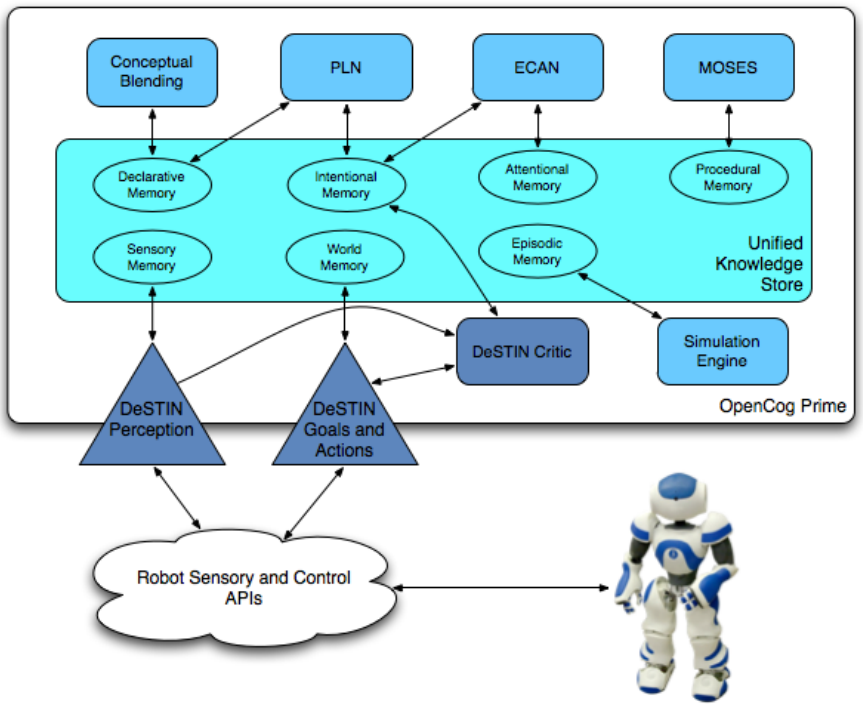

Figure 2: High-Level OpenCogBot Architecture Diagram

DeSTIN can initially be taught what servomotor commands to correlate with a high-level movement command via supervised learning: for instance, the robot can be forcibly moved according to a human's interpretation of the command "grab the A at coordinates (X, Y, Z)" while the command is fed into it from OCP, and once this is done for many values of A, X, Y, Z then DeSTIN's hierarchies will learn the command in a generalizable way.

DeSTIN can also be taught to recognize certain classes of objects and events by a similar supervised learning methodology; but the majority of DeSTIN's object and event recognition will be unsupervised, in which case recognized object or events will be identified in DeSTIN via using OCP's pattern-mining process to recognize simple attractor patterns in DeSTIN's upper layers. For instance, a specific cup will correspond to one such pattern; the general concept of "cup" will correspond to another such pattern. These mined patterns will be represented in OCP as PerceptNodes, and it will be OCP's job to supply semantic interpretations to these percepts as appropriate, based on correlating them with linguistic inputs or other percepts or concepts. All that's required of DeSTIN is that it consistently forms the same attractors in its upper layers, for stimuli that represent objects, events or other meaningful sensory regularities.

Finally, the reinforcement signals that guide DeSTIN's learning via its Critic hierarchy, will be linked to the Goal nodes in OCP's memory network, so that OCP's inferences and judgments can be used to provide DeSTIN's nodes with cognitively savvy feedback on their state inferences and generated actions.

Deep Integration of Cognition, Perception and Action The integration described above is not trivial, but it is simplistic in the sense of lacking feedback from cognition into perception and action. This sort of integration will suffice for many purposes, but we don't expect it will suffice for preschool level intelligence, especially not in the areas of visual attention allocation or manipulation of unfamiliar objects. For the latter problem areas, we believe, a tighter integration between perception and cognition will be required, manifested in OpenCogBot via a real-time dynamical relationship between the internal representations of DeSTIN and OCP. With a tighter

integration the "cognitive synergy" of the system can extend beyond OCP into DeSTIN, thus binding together the system as a whole.

Structurally this does not require anything new – it merely requires the same links between PerceptNodes and DeSTIN Nodes described in the previous paragraphs. However, the difference is that, to enable feedback from cognition to the sensorimotor layers, activation will be spread from OCP to DeSTIN and back again. Within OCP this activation will be interpreted as the ShortTermImportance values used by the ECAN attention allocation mechanism; within DeSTIN it will be interpreted as neural-net style activation. The passage of activation between the two subsystems will allow each subsystem to focus the attention of the other, thus allowing OCP's long-term memory and reasoning to influence DeSTIN's pattern recognition and action generation activities. This will allow the system to think about how it is manipulating a complex object while it's in the midst of doing it, and to use abstract inference and sensory pattern recognition together to recognize unfamiliar objects or complex events.

With this sort of deep integration the hierarchical and heterarchical pattern networks mentioned in the patternist theory of intelligence will span the DeSTIN and OCP components in a coherent way. And the system's emergent self-model will also span the two components, forming a unified internal image of OpenCogBot as a sensing, acting, thinking, remembering system.

4 Current Work and Future Plans

At the present time, the OpenCogBot project is an an early stage, but we do have a basic system in place allowing OCP to control a Nao robot. At the moment we are not using DeSTIN but have configured a simpler perceptual network, which processes data from overhead cameras looking down on the robot; this simplifies perception processing significantly, as we can simply assign each region of the robot lab perceived by the overhead cameras to a certain region of the perception network. We then use a pipeline architecture to map the output of the perception network into OCP's Atomspace. We handle speech data separately via text-to-speech software which then feeds text directly into the Atomspace. On the action side, we are not currently using a network architecture, but are using the high-level movement signals from OCP directly to trigger movement scripts programmed in Choregraphe, the Nao's action control language.

Our current pre-alpha OpenCogBot does not yet constitute a test of the integration of OCP with hierarchical temporal memory, but it does show the basic viability of using OCP in combination with a perceptual network and action scripts to control a humanoid robot. The robot is able to navigate through its environment (even if there are other moving entities around), and is able to understand and classify the objects in its environment and their relationships. It can handle simple questions and commands regarding observed relationships, of the same kind that the OCP controlled virtual agents answer, as discussed in [17]: things like "What is near the red table?", "Go get the ball that's between the chair and the wall," etc.

At time of writing our focus is on extending the imitation and reinforcement learning behavior demonstrated by OCP in the virtual world to the robot. One of the main challenges here is getting the perceptual network to do *event recognition* correctly, for the cases of known and novel events. This is the sort of thing that pushes one to require a more sophisticated perceptual network such as DeSTIN.

Next Steps: Sending OpenCogBot to Preschool Once the architecture is more fully implemented, our goal is not merely to emulate the behaviors achieved by OCP in the virtual world, but also to work toward the ideas described in our earlier paper on "AGI Preschool" [18], which explain the value and particulars of the "robot preschool" environment for creating and teaching AGI systems. The basic idea is to create a robot lab environment roughly similar to a preschool for young human children (but adapted to the robot's different sensors and actuators), and then focus on supervised and unsupervised learning roughly analogous to that carried out by preschool children. One of the advantages of this approach is that a variety of instruments already exist for evaluating the intelligence and capability of young children. Perhaps the most common is the

Wechsler Preschool and Primary Scale of Intelligence (WPPSI), an intelligence test designed for children ages 2 years 6 months to 7 years 3 months developed by David Wechsler in 1967 [19]. In [20] we present a detailed methodology for devising tasks and metrics for a humanoid robot in a robot preschool context, inspired by standard child intelligence tests and also broader cognitive science ideas.

Future Visions Finally, we take a few moments to reflect on the project from a broader perspective. As well as being extremely useful for applications like service robotics, a robot capable of carrying out preschool activities will arguably constitute a solution to the "common sense knowledge" problem, which according to many analyses has been the primary bottleneck holding back the AI field. It seems likely that the core ideas underlying the OpenCogBot project – as well as much of the software – will be useful in a wide variety of AI projects. OCP has already been used to aid with scientific data analysis and hypothesis discovery; and we are particularly interested in the prospect of hybridizing the commonsense capability of the robot preschool student with the specialized acumen of these science-oriented OCP applications, to yield novel forms of "artificial scientist." But this is only one among very many possibilities: in fact it would be hard to list important AI application areas where software embodying commonsense knowledge and the capability to flexibly acquire and utilize it would *not* be deeply valuable.

References

[1] B. Goertzel, "Opencog prime: A cognitive synergy based architecture for embodied artificial general intelligence," in *Proceedings of ICCI-09, Hong Kong*, 2009.

[2] I. Arel, D. Rose, and T. Karnowski, "A deep learning architecture comprising homogeneous cortical circuits for scalable spatiotemporal pattern inference." *NIPS 2009 Workshop on Deep Learning for Speech Recognition and Related Applications*, Dec 2009.

[3] B. Goertzel and H. de Garis, "Xia-man: An extensible, integrative architecture for intelligent humanoid robotics," pp. 86–90, 2008.

[4] B. Goertzel and D. Duong, "Opencog ns: An extensible, integrative architecture for intelligent humanoid robotics," 2009.

[5] B. Goertzel, *The Hidden Pattern*. Brown Walker, 2006.

[6] E. Tulving and R. Craik, *The Oxford Handbook of Memory*. Oxford University Press, 2005.

[7] M. Hutter, *Universal AI*. Springer, 2005.

[8] B. Goertzel, I. G. M. Iklé, and A. Heljakka, *Probabilistic Logic Networks*. Springer, 2008.

[9] G. Fauconnier and M. Turner, *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic, 2002.

[10] M. Looks, *Competent Program Evolution*. PhD Thesis, Computer Science Department, Washington University, 2006.

[11] B. Goertzel and C. P. et al, "An integrative methodology for teaching embodied non-linguistic agents, applied to virtual animals in second life," in *Proc. of AGI-08*, 2008.

[12] B. Goertzel, J. Pitt, M. Ikle, C. Pennachin, and R. Liu, "Glocal memory: a design principle for artificial brains and minds," *Neurocomputing, Special Issue of Artificial Brain*, to appear.

[13] J. Bach, *Principles of Synthetic Intelligence*. Oxford University Press, 2009.

[14] B. Goertzel, C. Pennachin, and N. Geisweiller, *Building Better Minds: Engineering Beneficial General Intelligence*. In preparation, 2010.

[15] B. Goertzel, H. Pinto, and C. Pennachin, "Using dependency parsing and probabilistic inference to extract relationships between genes, proteins and malignancies implicit among multiple biomedical research abstracts," in *Proceedings of the BioNLP Workshop/HLT-NAACL*, 2006.

[16] J. Hawkins and S. Blakeslee, *On Intelligence*. Brown Walker, 2006.

[17] B. Goertzel and et al, "A general intelligence oriented architecture for embodied natural language processing," in *Proc. of AGI-10*, 2010.

[18] B. Goertzel and S. V. Bugaj, "Agi preschool," in *Proc. of AGI-09*, 2009.

[19] D. Wechsler, *WPPSI-III Technical and Interpretive Manual*. The Psychological Corporation, 2002.

[20] S. Adams, B. Goertzel, and et al, "A roadmap toward human-level artificial general intelligence," *Submitted for publication*, 2010.