

Jeffrey Epstein IV Foundation
AGI Initiative
R&D Funding Proposal

Ben Goertzel

December 2, 2013

Executive Summary

This proposal outlines a research initiative aimed at creating **human-level, generally-intelligent thinking machines** within an 8 year period.

During the last few decades, the AI field has wandered from its initial focus on human-like general intelligence, and has devoted nearly all its attention to the creation of highly specialized, task-specific AI. While this focus on “narrow AI” has borne impressive fruit, the time has come to redirect attention toward the field’s original goals. Given recent advances in computing hardware and algorithms, cognitive science and neuroscience, the goal of building powerful Artificial General Intelligence (AGI) is far more achievable now than in the 1950s when the AI field was founded.

In the proposed research, we will create a general-purpose cognitive engine, and demonstrate and test it on tasks from a number of domains:

- Intelligently controlling animated characters in a 3D video game style world
- Controlling humanoid and wheeled robots in an indoor physical environment
- Engaging in natural language dialogue utilizing information on the Web
- Analyzing genomics datasets relating to the longevity of various organisms
- Automated program learning and theorem proving

From a narrow AI perspective, these endeavors might seem to have little in common – but from an AGI perspective they are fairly similar, as they all all require general intellectual abilities such as those found in the human brain.

Of course, like all revolutionary scientific advances, once advanced AGI has been achieved it will have a wealth of practical applications. The five areas mentioned above are only examples, which we have chosen largely due to our own prior experience applying AI in those domains.

Creating something as complex as a human-level mind requires a comprehensive, coherent design based on sound science and engineering principles. This is something that did not exist in the early decades of the AI field, when scientists greatly underestimated the problem. But we have learned that lesson, and we have completed a full design, which we call the CogPrime AGI Design.

CogPrime is a high level design for a human-level AGI system. It leaves many medium and lower level algorithmic and implementation problems open; but it gives a clear, coherent conceptual and software framework for AGI engineering, detailed design and experimentation. For a detailed description and an explanation of why we believe the CogPrime AGI Design has the capability to create true general intelligence, we refer the reader to the overview available at http://wiki.opencog.org/w/CogPrime_Overview. For a more complete and fully technical description, we refer to the books *Engineering General Intelligence Vol. 1 and 2*, by Ben Goertzel, Cassio Pennachin and Nil Geisweiller, to be released in December 2013.

We project that this project can be completed in an eight year period, by a team of 80 scientists and engineers, with a budget of US\$80 million. This cost could be reduced by a factor of 3-4 via offshoring some of the R&D.

In the first five years of the project, we will develop a thinking machine with the general intelligence of a 3 to 5 year old child, and highly powerful, practically useful capabilities in the areas of natural language dialogue, game AI, toy robotics, computer algorithm design, theorem-proving and genomic data analysis. In the following three years, we will give the thinking machine the ability to modify its own AI algorithms and to act independently as a research scientist. It will also be able to control robots moving commonsensically through everyday human environments, including attending and passing college classes.

Contents

1	Introduction	4
1.1	Why Now?	4
1.2	The CogPrime Design	5
1.3	Application Foci	6
1.4	Modular Design & Development	7
1.5	Open Source Development	8
1.6	Potential for Commercial Spin-offs	8
2	Incremental Development Milestones	9
2.1	Phase 1	9
2.2	Phase 2	14
3	Staffing and Costs	18
3.1	Advisors and Technical Leads	18
3.1.1	Technical Leads	18
3.1.2	Advisors	20
3.2	Estimated Costs	20
3.2.1	Lower-Cost Alternatives	21
4	Scientific and Technical Development	21
4.1	Integrated Cognition	21
4.1.1	Unified Rule Engine	22
4.1.2	Probabilistic Reasoning	22
4.1.3	Motivation and Emotion	22
4.1.4	Procedure Learning	22
4.1.5	Procedure Execution	23
4.1.6	Pattern Mining	23
4.1.7	Planning	23
4.1.8	Language Processing	23
4.1.9	Attention Allocation	23
4.1.10	Concept Formation	24
4.1.11	Perception	24
4.1.12	Action	24
4.2	Distributed and Multicore Processing Infrastructure	24
4.3	Sensation (Vision, Audition & Haptics)	25
4.4	Robot Movement Control	25
4.5	Game World Development	26
4.6	Genomics Data Analysis	26
4.7	Text, Image and Video Mining	27
4.8	Automated Programming	27
4.9	Automated Theorem Proving	28
4.10	Teaching and Intelligence Testing	29
4.11	Software Integration and Testing	29

1 Introduction

A best-selling book a few years ago claimed that “All I Really Need to Know I Learned in Kindergarten.” This maxim has some truth in an AI context. Arguably, what current AI programs are lacking is the kind of commonsense that every normal human toddler has. If one could create an AI with toddler-like common sense, alongside the specialized calculating and problem-solving ability that today’s intelligent software already displays, one would be well on the way to creating software systems with general intelligence at the adult human level and beyond.

This is precisely the thrust of the present proposal. Leveraging a new Artificial General Intelligence (AGI) design called CogPrime and an allied open source software system, OpenCog, we plan to **begin** by creating an AI software system that

1. displays the commonsense knowledge and everyday creativity of a young human child, in the context of controlling an animated character in a game world, and controlling a humanoid robot in an indoor environment
2. displays impressive practical prowess at a variety of intellectual tasks: algorithm design, geometric theorem-proving, and genomic data analysis

This combination, in itself, will not constitute the “end game” of our AGI work. However, we believe this “Phase 1” achievement will encapsulate solutions to the hardest problems in AGI design and engineering, and leave us poised to take the next step – toward a “Phase 2” AGI system that possesses the general intelligence of a human adult, enhanced by nonhuman calculational and problem-solving capabilities that digital computers bring. The Phase 2 system will be a genuine “AGI Scientist”, which can self-modify and improve its own intelligence, alongside other science and engineering capabilities

We propose that a “Phase 1” early-stage AGI system, displaying the dual capabilities described above as aspects of its unified cognitive functionality, can be produced within 5 years at a rough cost of US\$9 million per year. Further, we propose that once Phase 1 has been achieved, Phase 2 can be accomplished within the same design and software system, and will actually be a smaller leap: 3 further years of effort at the same rate.

1.1 Why Now?

Human-level general intelligence was, of course, the original focus of the founders of the AI field in the 1950s and 60s. But the hardware, software and conceptual frameworks of that time were not adequate to the task, and in subsequent decades the AI field shifted focus to narrower problems. Today, however, our hardware, software and understanding have advanced considerably, so that human-level Artificial General Intelligence (AGI) is an initiative whose time has finally come.

Today, task-specific narrow AI currently pervades nearly every area of industry, within various forms of back-end software. It is also achieving an increasing public profile with achievements like self-driving cars, IBM Watson, online recommendation systems and chatbots like Siri and Google Now (to name just a few). Complementarily, neuroscience and cognitive science are providing us an ever-deeper understanding of the human brain and mind each year.

While pursuit of human-level AGI was marginalized within AI academia during the 1980s and 90s, now it is increasingly becoming accepted as a valuable R&D direction once again. We now have annual conferences on AGI, Cognitive Systems and Biologically-Inspired Cognitive Architectures (BICA), and also an increasing number of sessions related to human-level AGI and AI within AAAI, IJCAI, IEEE and other generic AI-oriented conferences. AI visionary Ray Kurzweil has recently taken a position as a Director of Engineering at Google; and industry luminaries like Intel CEO Justin Rattner now forecast the arrival of human-level AGI within decades.

The time is ripe for a serious frontal assault on the AGI problem. We have the tools and the knowledge; all that’s needed now is the courage and persistence to confront the problem head-on. AGI is not a trivial problem by any means, combining as it does multidisciplinary R&D with large-scale software engineering. But given the technology and science of 2013, human-level AGI is an eminently reasonable near-term development goal.

1.2 The CogPrime Design

To create something as complex as a human-level mind requires a comprehensive, coherent design based on sound science and engineering principles. The foundation of the proposed research is the CogPrime AGI design, described in the books *Engineering General Intelligence Vol. 1 and 2*, by Ben Goertzel, Cassio Pennachin and Nil Geisweiller, published in December 2013 [10, 11].

CogPrime provides a high level design for a human-level AGI system. It leaves many medium and lower level algorithmic and implementation problems open; but it gives a clear, coherent conceptual and software framework for AGI engineering, detailed design and experimentation. Exposition of CogPrime and why we believe it has the capability to yield human-level general intelligence would extend this proposal excessively, so we refer the reader to the online CogPrime overview available at http://wiki.opencog.org/w/CogPrime_Overview. There have also been several academic conference papers published on CogPrime and its potential, most recently [8] presented at the 2013 *IEEE Symposium on Human-Level AI*, but the online article gives the clearest concise exposition of the design at this stage.

The conceptual foundation of CogPrime is the “patternist” theory of mind developed in Ben Goertzel’s work during the 1990s and elaborated at length in [5], which views an intelligent system as concerned with recognizing patterns in itself and the world, focused substantially toward patterns regarding what actions will achieve its goals in observed contexts. On a practical level, the high-level architecture of CogPrime 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 would be simple things such as pleasing the teacher, learning new information and skills, and protecting the robot’s body. Each of CogPrime’s cognitive processes is biased to recognize particular sorts of patterns, and the particular assemblage of cognitive processes is chosen based on a careful analysis of human cognition, with input from neuroscience, linguistics, philosophy of mind, computer science and other disciplines as well.

CogPrime’s memory types are the declarative, procedural, sensory, and episodic memory types that are widely discussed in cognitive neuroscience [18], 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 crudely indicating which fundamental cognitive dynamics each cognitive process corresponds to (pattern creation, association, etc.).

The essence of the CogPrime 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.” All OpenCog memory types are implemented using a common weighted, labeled hypergraph knowledge store called the Atomspace; and all OpenCog cognitive processes are implemented as software objects called MindAgents, which interact with the Atomspace.

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**

Obviously this sort of high-level sketch merely serves to evoke the rough nature of the CogPrime system, and the curious reader should peruse the above references to get a fuller picture.

The open source OpenCog software project (see opencog.org) provides a foundation designed explicitly for the implementation of the CogPrime design, and currently contains partial implementations of many of the algorithms constituting CogPrime. OpenCog has been used for commercial applications in the area of natural language processing and data mining. It has also been used for research involving controlling virtual agents in virtual worlds, and humanoid robots.

Memory Type	Specific Cognitive Processes	General Cognitive Functions
Declarative	Probabilistic Logic Networks (PLN) [4]; conceptual blending [3]	pattern creation
Procedural	MOSES (a novel probabilistic evolutionary program learning algorithm) [16]	pattern creation
Episodic	internal simulation engine [9]	association, pattern creation
Attentional	Economic Attention Networks (ECAN) [14]	association, credit assignment
Intentional	probabilistic goal hierarchy refined by PLN and ECAN, structured according to MicroPsi [2]	credit assignment, pattern creation
Sensory	Supplied by the DeSTIN component	association, attention allocation, pattern creation, credit assignment [1]

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

The present proposal is aimed at completing the detailed design and implementation of the CogPrime AGI architecture within the OpenCog framework.

1.3 Application Foci

It is important that an AGI system must do *something* even as its development proceeds; intelligence is centrally about engagement in a variety of complex tasks in complex environments. But choice of any one specific application runs the risk of overfitting the work to that application and ending up with a more specialized system than intended. Hence we propose six different application foci, to be pursued concurrently using the same integrated intelligent system:

1. Control of intelligent animated characters in a 3D “video game” style world
2. Control of (humanoid and wheeled) mobile robots in an indoor environment
3. Natural language dialogue in the context of information available on the Web
4. Analysis of genomics datasets related to longevity in various organisms
5. Automated program learning
6. Automated theorem proving

Each of these applications stresses different aspects of CogPrime and relates preferentially to different aspects of human intelligence. Pursuing them concurrently with the same developing AGI system guarantees generality of focus on the application level, alongside the generality of capability existing on the software level due to the nature of the CogPrime design.

These applications have also been chosen because, with the exception of automated theorem proving, they are all areas that have already been explored using the OpenCog system, either in commercial or prototype applications; to wit:

1. **Animated character control:** In the period 2008-2013, a variety of research prototype systems have been built using OpenCog to control virtual characters in virtual worlds [9]. Currently this initiative has research funding from the Hong Kong government, aimed at creating a toolkit enabling OpenCog-controlled non-player characters to be used in commercial games.
2. **Mobile robot control:** In 2009, at Xiamen University, a research project was conducted involving the use of OpenCog to control a humanoid Nao robot [15] [13]. This work was documented in the award-winning film *Singularity or Bust*, see <http://singularityorbust.com>. Currently the Hong Kong government is providing research funding aimed at extending this work via using OpenCog to control David Hanson's RoboKind humanoid robots, during 2014 and 2015.
3. **Natural language dialogue:** OpenCog's language processing tools have been used on the back end of several practical applications, such as an online language-teaching site, and a US government information system aimed at intelligence analysts. While these have not focused on dialogue, they have stressed most of the same NLP components that will be used in dialogue. These language processing tools have also been used in research together with the NIH on information extraction from PubMed abstracts [12]. A Hong Kong government grant has been obtained to support extension of this work toward the creation of an OpenCog dialogue system, in the specific context of a smartphone-based dialogue agent focused on media consumption.
4. **Genomics data analysis:** OpenCog tools, primarily MOSES but also PLN and clustering tools, have been extensively used to analyze genomics data for commercial, government and academic customers. This has led to various successes such as discovering the first genetic basis for Chronic Fatigue Syndrome [7], learning highly accurate diagnostics for Alzheimers and Parkinsons Disease [17], and understanding the means via which calorie restriction impacts longevity [6].
5. **Automated program learning:** OpenCog's MOSES component, which performs automated program learning, has been used for numerous custom commercial data mining jobs (via the consulting firm Novamente LLC), to learn small programs constituting patterns in data ranging from biology to finance to market research and power transformer performance

1.4 Modular Design & Development

The CogPrime architecture is structurally modular but dynamically unified. This means that, from an engineering perspective, it subdivides human-level AGI into a set of discrete modules. However, the intended intelligent operation of the whole system is dependent on synergetic interactions between the modules. Each module is intended to display meaningful intelligent behaviors on its own but it's expected that these behaviors will be less scalable and more narrowly-scoped, than the behaviors the same modules will display in the context of the overall integrated CogPrime system.

The modular architecture of CogPrime naturally supports development by a distributed team of teams, with teams focused on particular modules, teams focused on infrastructure tools useful across multiple modules, and then a central integrative team focused on putting all the pieces together to achieve overall generally intelligent behavior. While the module-specific teams may be purely research oriented, and the infrastructure teams may be purely engineering oriented, the central integration team must combine research and engineering capabilities.

We propose that development be broken down according to the following teams, some of which correspond specifically to application areas as outlined above, and some of which correspond to important support tasks:

1. Integrated Cognition
2. Distributed and Multicore Processing Infrastructure

3. Sensation (Vision, Audition & Haptics)
4. Robot Movement Control
5. Game World Development
6. Genomics Data Analysis
7. Text, Image and Video Mining
8. Automated Programming
9. Automated Theorem Proving
10. Teaching and Intelligence Testing
11. Software Integration and Testing

In Section 4 below we summarize the basic approaches and tasks we propose each team to undertake. While it would be viable to colocate all the teams, it would also be viable to spread the teams among different locations based on the existence of relevant expertise. Some of the larger teams could potentially be split among more than one location, in themselves.

1.5 Open Source Development

The OpenCog system has been developed as an open source software platform since 2008, and we propose to continue development of the CogPrime design in this vein. The advantage the OSS methodology presents is the capability of leveraging dramatic additional intellectual and software development, debugging and testing resources, via leveraging of the academic and OSS software development communities. On the software level, the advantages of OSS for ensuring software robustness, scalability and stability are well known. On the intellectual level, there are obviously very considerable benefits to be achieved via open involvement of the academic research world in the ongoing improvement of various aspects of a human-level AGI system as it matures.

While a traditional business perspective would suggest that open sourcing the proposed software development is a negative from the perspective of ultimate monetization of the results, this is not necessarily the case. There are many viable, and potentially highly lucrative business models that the funders and developers of the proposed AGI software could pursue, that would not be negatively impacted by the open source nature of the underlying software.

1.6 Potential for Commercial Spin-offs

The R&D project described here is proposed primarily for the transformative effect it would have upon science, technology, society and the evolution of intelligence. However, as a side-effect, numerous possibilities will arise along the way for leveraging the technology developed in various business domains. As AGI has the potential to transform every single area of commerce, giving a comprehensive list here would not be viable. However, a few of the possibilities closest to the specific AI applications to be pursued in the proposed work are:

- AGI non-player characters for video games, or (depending on the game design) AGI for controlling whole game worlds. This could be provided as game-AGI middleware, or on a custom per-game basis in partnership with game companies
- AGI toy robots, home service robots or elder care robots
- Combining game characters, robots and other possibilities, it would be viable to develop a cloud-based facility for serving OpenCog based intelligence to various online software applications – e.g. games, robots, consumer electronic devices, specialized information systems. The developers of the first generally intelligent OpenCog system would obviously have a substantial “first-mover advantage” in setting up such a facility. And once

such a facility were operational, the cloud-resident AGI would gain dramatic knowledge from its customers in the course of its operation, setting up an “increasing returns” dynamic that would make it very difficult for competitors to catch up (similar to but quite likely more substantial than the increasing-returns based advantages enjoyed by current firms such as Google and Facebook).

- A host of genomics-related biomedical opportunities, including
 - Discovery of targets for pharmaceutical, nutraceutical and gene therapy interventions for age-associated and other diseases.
 - Integration of AGI software with rational drug design software to enable creation of novel molecules targeting combinations of genes highlighted by AGI genomic/proteomic analysis.
 - Predictive toxicology, to identify the human body’s reactions to substances prior to their synthesis
 - Integration of AGI with systems biology simulation software to enable simulation of organismic response to therapies
- Conversational personal assistants – like Siri, but with genuine understanding of what they’re talking about
- Conversational agents for helpdesk and customer support

By its very nature a general intelligence can be applied in multiple domains, for great benefit and also potentially substantial financial profit. But first we must meet the challenge of creating a core of generally intelligent software, which is the focus of the proposed R&D work.

2 Incremental Development Milestones

2.1 Phase 1

Here we list high-level development milestones in each of the identified application areas, year by year for Phase 1 of the proposed project. Much more detailed milestones will be established year by year as the work proceeds.

Year 1

Capability	Milestone
Animated Agent	Conception and execution of plans to achieve complex movement and building tasks in blocks-focused game world
Mobile Robot	<ul style="list-style-type: none"> • Effective recognition of a closed class of objects and events • Navigation in dynamic indoor environments
Dialogue System	Simple dialogue about objects , events and goals in the game world
Genomic Analysis	Construction of an integrated Atomspace-based knowledge base containing gene expression, SNP and protein-protein interaction data, pathway and Gene Ontology data, and information extracted from PubMed abstracts
Automated Programming	MOSES-based learning of sorting and searching algorithms
Automated Theorem Proving	Effective importation of Mizar formalized math database into OpenCog knowledge representation

Year 2

Capability	Milestone
Animated Agent	<ul style="list-style-type: none"> • Linguistic communication about needs and desires in game world • Event recognition • Recognition of never-before-seen objects • Social reasoning in game world; theory of mind; basics of empathy, and social manipulation and deception
Mobile Robot	<ul style="list-style-type: none"> • Recognition of unfamiliar objects and events • Simple, goal-directed reaching and grasping
Dialogue System	Dialogue about objects , events and goals in the game world, involving complex sentences with multiple clauses
Genomic Analysis	<ul style="list-style-type: none"> • Supervised (MOSES) and unsupervised (MOSES, clustering, pattern mining) learning based analysis of multiple genomic datasets utilizing integrated information • Extrapolation of consequences for drug/nutraceutical target discovery and diagnostics for age-associated diseases
Automated Programming	MOSES-based learning of simple AI heuristics for solving puzzles and narrow-AI problems
Automated Theorem Proving	Simple set theory and geometry theorem proving within OpenCog

Year 3

Capability	Milestone
Integrated Cognition	Ability to pass “3 year old child” variant of AIQ test in the game world
Animated Agent	<ul style="list-style-type: none"> • Construction of complex objects • Group creativity among several AGI agents in game world • Following and giving of multi-step instructions
Mobile Robot	<ul style="list-style-type: none"> • Perception-guided object manipulation with robot hands • World-understanding based on integrating acoustic and visual data
Dialogue System	<ul style="list-style-type: none"> • Game-world conversations involving roughly human child like understanding of context and intention • Dialogue about objects, events and goals in the physical world, in the robotics context
Genomic Analysis	<ul style="list-style-type: none"> • integration of information extracted from research article bodies • conception of novel hypotheses regarding relationships between biological entities and processes, via PLN inference
Automated Programming	Learning of modular programs, combining other learned programs in judicious ways
Automated Theorem Proving	Use of PLN probabilistic reasoning to guide set theory and geometry theorem proving

Year 4

Capability	Milestone
Integrated Cognition	Ability to pass “4 year old child” variant of AIQ test in the game world
Animated Agent	Carrying out of simple “scientific” experimentation in game world
Mobile Robot	<ul style="list-style-type: none"> • Building structures from blocks and other simple objects • Supplementation of third-party speech-to-text with deep learning based speech-to-text
Dialogue System	<ul style="list-style-type: none"> • Ability to understand and produce sequences of sentences embodying a coherent, contextually relevant thought (in the game world and physical world) • Dialogue about information extracted from texts (not only about directly experienced events)
Genomic Analysis	“Artificial bioinformatic scientist” functionality involving iterated automated generation of hypotheses and testing of hypotheses against datasets
Automated Programming	Learning of simple programs that involve interaction with the Atomspace
Automated Theorem Proving	Use of intuitions gained from the game world to guide geometric theorem proving

Year 5

Capability	Milestone
Integrated Cognition	<ul style="list-style-type: none"> • Ability to pass “5 year old child” variant of AIQ test in the game world • Ability to pass “3 year old child” variant of AIQ test in the robotic embodiment
Animated Agent	Solution of complex game-world puzzles based on a combination of formal and intuitive reasoning
Mobile Robot	Creative, child-like play with physical objects
Dialogue System	Ability to understand and produce sequences of sentences embodying a coherent, contextually relevant thought, regarding information extracted from text
Genomic Analysis	<ul style="list-style-type: none"> • Generation/testing of more complex hypotheses • Integration of more complex information from research article bodies
Automated Programming	Learning of programs involving AI-based heuristics that interact with the Atomspace
Automated Theorem Proving	<ul style="list-style-type: none"> • Use of intuitions gained from robotics to guide geometric theorem proving • Extrapolation from experientially grounded (game-world-related) to ungrounded theorem-proving via PLN analogical inference; e.g. from geometry to indirectly geometry-related set theory algebra

2.2 Phase 2

Phase 1 development will be focused on implementation and testing of new CogPrime functionalities, and exploration of the implications of these functionalities in the chosen application domains. Phase 2, on the other hand, is intended to encompass 3 years of **teaching** the AGI system, and watching it learn and explore, and making modifications to the system as merited by observing its progress.

It is difficult, at this stage, to project the learning progress of a CogPrime system of this level of sophistication. However, based on the nature of the CogPrime design, we can conjecture with reasonable solidity as to what kind of functionality the system may acquire after roughly 3 years of learning from its environment and its human teachers:

Year 6

Capability	Milestone
Integrated Cognition	<ul style="list-style-type: none"> • Ability to pass “8 year old child” variant of AIQ test in the game world • Ability to pass “5 year old child” variant of AIQ test in the robotic embodiment
Animated Agent	Ability to robustly take knowledge gained in the game world and port it to the physical world, and vice versa
Mobile Robot	Leaving the lab and learning to navigate and socially interact in the city streets (with human assistance at first)
Dialogue System	<ul style="list-style-type: none"> • Ability to hold intelligent conversations at the rough level of a 7 year old human child • Ability to robustly learn new words and linguistic expression patterns from experience • Ability to read and understand general written information aimed at children aged 7 or younger • Ability to effectively correlate words with images and videos, as required for understanding e.g. children’s books or educational videos
Genomic Analysis	<ul style="list-style-type: none"> • Formalized understanding of experimental designs and how they relate to datasets • Robust inferential connection between biological domain knowledge and general knowledge of the everyday world
Automated Programming	Ability to automatically create simple MindAgents to perform aspects of OpenCog reasoning, based on formal descriptions of MindAgent requirements
Automated Theorem Proving	Ability to prove theorems in more abstract areas of (still elementary) geometry or set theory, beyond what the game world provides grounding for

Year 7

Capability	Milestone
Integrated Cognition	<ul style="list-style-type: none"> • Ability to pass adult-level variant of AIQ test in the game world • Ability to take ordinary human IQ test and correctly understand and answer a majority of questions
Animated Agent	Ability to enter a variety of game worlds, understand the properties of these worlds, and figure out how to achieve the relevant goals there
Mobile Robot	More robust interaction in a variety of social and physical situations outside the lab
Dialogue System	<ul style="list-style-type: none"> • Ability to hold intelligent conversations at the rough level of a 10 year old child – about science, and also about human relations and the system’s own mind-state • Ability to read and understand most Web pages, except those with highly specialized or informal content • Ability to learn aspects of new languages based on experience and teaching
Genomic Analysis	Creation of hypotheses, analysis of data and design of experiments at the level of a human biology undergraduate student
Automated Programming	<ul style="list-style-type: none"> • Ability to write simple scripts to carry out functions in the Linux operating system • Ability to create MindAgents performing aspects of OpenCog reasoning, based on informal description of MindAgent requirements
Automated Theorem Proving	Ability to prove theorems in more advanced undergraduate geometry, set theory, topology and calculus

Year 8

Capability	Milestone
Integrated Cognition	Ability to pass an ordinary human adult IQ test with a strong score
Animated Agent	Ability to enter an essentially arbitrary new game world, understand the properties of the world, and figure out how to achieve the relevant goals in the world
Mobile Robot	<ul style="list-style-type: none"> • Ability to navigate and manipulate objects in unfamiliar sorts of environments (e.g. outdoors, in a basement, etc.) • Ability to automatically adapt to new robotic body parts
Dialogue System	<ul style="list-style-type: none"> • Ability to hold intelligent, though not necessarily precisely human-like, adult-level conversations – about science, and also about human relations and the system’s own mind-state • Ability to adapt all aspects of language comprehension and generation based on linguistic experience • Ability to read and understand most Web pages • Ability to learn new languages based on experience and teaching, in the manner of human language learners
Genomic Analysis	Creation of hypotheses, analysis of data and design of experiments at the level of a human biology graduate student
Automated Programming	<ul style="list-style-type: none"> • Ability to write simple scripts to carry out functions in the Linux operating system • Ability to modify its own MindAgents for superior functionality
Automated Theorem Proving	Ability to prove theorems in all areas of undergraduate mathematics, guided when needed by analogy to its grounded experience with geometry, set theory and arithmetic

3 Staffing and Costs

3.1 Advisors and Technical Leads

3.1.1 Technical Leads

Key to the success of the proposed work will be the involvement of individuals already expert in OpenCog software and its application, e.g.

Name	Role
Dr. Ben Goertzel	founder of the OpenCog project
Dr. Linas Vepstas	principal engineer of the OpenCog software system for the last several years
Dr. Nil Geisweiller	<ul style="list-style-type: none"> • current main developer of OpenCog's MOSES subsystem • prior developer of OpenCog's PLN subsystem and OpenCog's connection to the Nao robot • coauthor of <i>Engineering General Intelligence</i>
Dr. Eddie Monroe	OpenCog machine learning engineer at Novamente LLC
Dr. Matthew Ikle'	co-developer of the mathematics underlying OpenCog's attention allocation and probabilistic logic modules
Dr. Joel Pitt	former OpenCog Hong Kong team lead, developer of OpenCog's attention allocation module
Ruiting Lian	lead OpenCog natural language developer
David Hart	OpenCog IT/infrastructure guru since the project's start
Shujing Ke	lead OpenCog planning, pattern mining & game-AI developer
Lake Watkins	current developer of the game world used for testing OpenCog
Scott Jones	OpenCog core system developer, current team lead of OpenCog Hong Kong project
Cosmo Harrigan	OpenCog AI developer
Alex van der Peet	OpenCog game world and AI developer
Jade O'Neill	OpenCog AI developer (principal developer of OpenCog's current Probabilistic Logic Networks implementation)
Ted Sanders	DeSTIN expert
Michel Drenthe	DeSTIN expert
Misgana Bayetta	OpenCog (MOSES) developer
Teddy Habtegabriel	DeSTIN developer
Rodas Solomon and Amen Belayneh	OpenCog language processing specialists
Keyvan Sadeghi	author of OpenCog temporal/spatial reasoning
Mike Duncan	Biomind LLC bioinformaticist, currently applying MOSES to genomics datasets
Angus Griffiths	lead developer of Mathics, key tool for automated theorem proving

The availability of a relatively large team of experienced AI software developers who are “ready to go” on OpenCog applications is a valuable asset.

3.1.2 Advisors

The proposed work will also benefit substantially from the part-time participation of a set of AI-expert advisors, including many who have collaborated on OpenCog work in one way or another in the past, and some who have developed their own related ideas. Most of these individuals were mentioned above in the sections specific to their expertise areas, but are listed here in summary form, along with their relevant specialty.

Name	Affiliation	Relevant Specialty
Cassio Pennachin	Aidyia Limited	original software architect of OpenCog
Itamar Arel	U. Tennessee Knoxville	vision and audition
Peter Stone	U. Texas Austin	robotics
Bertram Shi	HKUST	robotics
Mark Tilden	consultant	robotics
David Hanson	Hanson Robotics	robotics
Tom Mitchell	Carnegie Mellon	information extraction
Dan Miller	consultant	virtual world infrastructure and robotics
Michael Rose	U.C. Irvine	genomics
Joao Pedro de Magalhaes	U. Liverpool	bioinformatics, genomics
Predrag Janicic	University of Belgrade	automated theorem proving
Moshe Looks	Google	automated program learning
Paul Rosenbloom	USC	probabilistic reasoning
Noah Goodman	Stanford	probabilistic reasoning
Joscha Bach	MIT/Harvard	motivation & emotion
Eray Ozkural	Bilkent University	automated program learning
Juergen Schmidhuber	IDSIA	automated program learning, vision, robotics

Note: Some but not all of the above-mentioned individuals have been explicitly asked about potential involvement in this specific project, at this stage. All have previously been in discussions regarding collaboration and advisement on OpenCog related AGI projects.

3.2 Estimated Costs

This section presents a crude and preliminary cost estimate for the proposed work. A detailed budget can be produced upon request, but will depend on specific assumptions such as the particular locations where the different aspects of the work will be carried out.

Assuming the Integrated Cognition team has 25 staff, and each of the other 11 teams outlined above each have 5 staff, we would then have a total of 80 scientific and technical staff on the project. As some staff will be senior and some junior, cost per staff member may vary, but if we assume a cost of US\$100K per year per staff member on average (including overheads), the result is a staff cost of US\$8 million per year. We will assume a cost of \$6M for the first year as staff will likely be brought on gradually.

We also assume that the team’s expert advisors will occasionally be paid consulting fees to give detailed reviews of system design and performance issues, and paid expenses to visit the project from time to time. Crudely we

assume this amounts to US \$200K/year.

Computing hardware will be a significant expense as well. If we assume 1000 cloud-based servers at a cost of US\$25 per month per server, the cost of hardware would be US \$300K per year. In the first years the need for hardware would be less than this, perhaps US \$150K. This many machines will also require significant dedicated IT support, for which we may budget roughly US \$300K per year, which adds to the above staff cost. In Phase 2 we will potentially require a greater amount of hardware, but the details are difficult to foresee in part because the costs of hardware are rapidly declining in complex ways.

Thus, exclusive of administrative costs, we arrive at a rough annual cost estimate as follows (all figures in millions of USD):

Year	Staff (fully loaded)	Advisors	Hardware	Total Cost
1	6.3	.2	.15	9.
2	8.3	.2	.3	10.
3	8.3	.2	.3	11.
4	8.3	.2	.3	11
5	8.3	.2	.3	11.
6	8.3	.2	.5	11.
7	8.3	.2	.5	11.
8	8.3	.2	.5	11.
Total	64.4	3.2	1.35	69

Adding reasonable administrative, travel, legal and miscellaneous costs we obtain a rough cost estimate of \$80M over 8 years.

3.2.1 Lower-Cost Alternatives

The above cost estimate assumes doing all, or nearly all, development in the US or other locations of comparable cost. An alternative strategy would be to have a team of perhaps 15 expert staff in the US or other high-cost locations, supplemented by a team in a low-cost location. A number of OpenCog developers are currently working from the office of iCog Labs, an AI outsourcing and R&D firm in Addis Ababa, Ethiopia. Extensive use of iCog Labs staff could bring the total annual cost down to US\$2.5-3M from \$11M, a substantial reduction. Intermediate scenarios are also a possibility, with less extreme utilization of offshoring resulting in significant but less extensive cost reduction.

4 Scientific and Technical Development

In this section we give rough indications of the work to be done to achieve the above-listed incremental milestones.

4.1 Integrated Cognition

The CogPrime design for advanced Artificial General Intelligence has, at the present time, only been partially implemented within the OpenCog open-source AGI software framework. Considerable work remains to complete the implementation and testing of the CogPrime design within the OpenCog system. Here we will describe only the most major points; however, numerous related similar tasks also need doing, many of which are documented on the OpenCog wiki site.

Implementation and testing of the CogPrime design is intended to be completed within the scope of Phase 1 of this proposal. The achievement of childlike commonsense and domain-specific advanced intelligence can be worked toward gradually, as more and more of CogPrime is implemented and made workable. During Phase 2, the focus will be on teaching the system and enabling it to improve its intelligence via spontaneous learning as well as human instruction. Improvements to the underlying AI code will be made during this phase as seems appropriate based on observations of the system's learning progress; but these improvements cannot be foreseen in detail at this stage.

4.1.1 Unified Rule Engine

At present the OpenCog system has a number of different “rule engines” implemented within it, which apply different sorts of “rewrite rules” to different sorts of data, in the context of different cognitive processes. This is workable but ultimately not the best approach; a design has been created for a unified OpenCog rule engine, but it needs to be fleshed out and implemented.

4.1.2 Probabilistic Reasoning

OpenCog currently contains an implementation of CogPrime’s Probabilistic Logic Networks reasoning system, integrated with the Economic Attention Allocation module to achieve scalable inference control. However, the current PLN implementation has a number of limitations in need of remedying, e.g.

- it should be modified to utilize the Unified Rule Engine mentioned above, when the latter is ready
- it currently uses only a simple version of probabilistic truth values, and should be extended to use “indefinite” and “distributional” truth values
- it has recently been extended to deal efficiently with temporal reasoning. Similar extensions need to be done for spatial and quantitative reasoning
- inductive inference control needs to be implemented, wherein the choice of next inference step is made based on analysis of a historical database of prior inferences

While PLN implements its own unique brand of AGI-oriented probabilistic logic, it also has numerous relationships with other probabilistic logic systems. As PLN work progresses, ongoing feedback from other probabilistic reasoning oriented AGI researchers such as Paul Rosenbloom (USC) and Noah Goodman (Stanford) will be valuable.

4.1.3 Motivation and Emotion

OpenCog’s OpenPsi module, modeled loosely on Joscha Bach’s MicroPsi architecture, currently causes an OpenCog agent’s actions to be chosen based on its goals. This functionality works reasonably but is currently relatively simplistic. Key improvements needed are:

- Implement the CogPrime system of “Request for Services” based economic goal fulfillment, wherein a goal offers Atoms “virtual funds” for helping it get achieved, in a way that relies on the Economic Attention Allocation subsystem
- Implement novelty, amount-of-learning and aesthetics as goals, based on information-theoretic definitions

4.1.4 Procedure Learning

The MOSES automated program learning algorithm is perhaps the most mature portion of the OpenCog codebase, yet still requires significant augmentation to be fully useful for AGI purposes, i.e.

- Implement effective modeling of multiple interdependent continuous-valued program inputs, to allow effective MOSES learning of programs depending on multiple continuous values (this is important e.g. for using MOSES to learn motor control procedures, where the continuous values represent states of a body’s motor control system)
- Extend MOSES’s program simplification aspect (Reduct) to handle higher-order functions and local variables, via using techniques from the functional programming literature such as director strings. This will allow MOSES to learn more complex programs
- Integrate MOSES with PLN inference to enable each MOSES learning run to benefit from prior MOSES learning runs via probabilistic analogical reasoning

4.1.5 Procedure Execution

The current OpenCog system lacks the facility to execute multiple procedures at the same time, and mediate potential conflicts between the procedures. That is, in terms of its interactions with an external world, the system cannot effectively “multitask.” Refinement and implementation of the CogPrime design for “execution management,” which models and acts on the dependences between concurrently executing procedures, is needed.

4.1.6 Pattern Mining

OpenCog’s current pattern mining subsystem, Fishgram (Frequent and Interesting SubHypergraph Mining), is prototype code and not sufficiently scalable. A detailed design has been outlined for reimplementing it in a scalable manner, but this needs to be executed on. Further, pattern mining needs to be integrated with MOSES so it can guide, and be guided by, MOSES learning.

4.1.7 Planning

OpenCog currently possesses a planner that integrates hierarchical plan learning with logical inference in a unique way, customized for learning of plans combining navigating and building in a 3D game world, but extensible much more generally. This planner needs to be refactored to utilize the Unified Rule Engine, and integrated with PLN so as to leverage probabilistic inference.

4.1.8 Language Processing

Currently OpenCog has a fairly robust language comprehension component, and a prototype language generation system, and very little facility for controlling the course of a natural language dialogue. Further, the comprehension and generation systems rely heavily on hand-coded rules and specialized rule engines. What needs to be done is to upgrade the current NLP system to be more fully learning-based and Atomspace-centered, i.e.

- Port the specialized hand-coded NLP rules currently used into Atomspace format, so that NLP can be done as a consequence of generic cognitive processing
- Replace invocation of specialized NLP rule engines with invocation of the Unified Rule Engine
- Transform the existing NL comprehension rules into a reversible format, so that they can be reversed and used for NL generation
- Deploy clustering and pattern mining to learn new language processing rules from corpus analysis, beginning with (reversible versions of) the existing NLP rules as initial conditions
- Implement initial dialogue control heuristics as Atoms in the Atomspace, so that speech acts can be chosen by OpenPsi along with other sorts of acts, based on the system’s motivations

4.1.9 Attention Allocation

OpenCog’s Economic Attention Allocation (ECAN) module works reasonably effectively on modest-sized Atomspaces at the moment, but needs

- extension to handle large Atomspaces efficiently
- fuller integration with PLN, to enable more sophisticated reasoning about what may become important and deserve attention in the future
- more sophisticated forgetting mechanisms
- mechanisms enabling it to pull knowledge into RAM from disk when merited

4.1.10 Concept Formation

Mechanisms for forming new concepts based on the action of clustering and pattern mining algorithms need to be implemented. The existing code enabling creation of new concepts via “conceptual blending” of existing concepts, needs to be extended via integrating information-theoretic measures of what constitutes a high-quality blend.

4.1.11 Perception

Currently the DeSTIN deep perceptual pattern recognition system exports cognitive information into the Atomspace via mediation of a “frequent pattern mining” component which recognizes patterns in DeSTIN’s state and exports these patterns into the Atomspace. However,

- This specialized pattern mining component is prototype-level and needs to be tuned and completed.
- Concept formation and PLN inference need to be tested on, and customized for, the drawing of inferences from patterns input to the Atomspace from DeSTIN.
- Information derived from cognitive inferences needs to be fed back into DeSTIN to bias its pattern formation. This feedback will require significant experimentation and tuning.

4.1.12 Action

The DeSTIN hierarchy needs to be extended to handle actions as well as perceptions. This is somewhat subtle since instead of the 3D space used for vision or the 1D space used for audition, for action DeSTIN must represent the higher-dimensional “Configuration Space” of a motoric dynamical system. The basic concepts and mechanisms of DeSTIN will carry over to this case, but numerous implementation changes will be required.

4.2 Distributed and Multicore Processing Infrastructure

To achieve human-level AGI, given current computing technology, will require a significant cloud computing infrastructure (a large network of multiprocessor Linux machines), and software customized to make use of said infrastructure. A rough estimate is on the order of 500-1000 quad-processor servers, including perhaps 200 with GPU as well as CPU. The OpenCog framework has been designed with extensibility to networks of this size in mind, but requires significant enhancement (within its current architecture) to be effective in this kind of deployment, e.g.

- refactoring of the core Atomspace knowledge repository to encompass a single Atomspace spanning a large number of machines (likely utilizing existing third-party graph DB technology, with specialized extensions)
- design and implementation of subsystems enabling automated management of OpenCog systems running on large networks
- implementation of code for automatically shifting knowledge among different machines in a large Atomspace, to optimize intelligence and efficient use of resources
- re-implementation of specific aspects of OpenCog to utilize GPUs effectively (e.g.: the DeSTIN perception/action hierarchy, truth value estimation in Probabilistic Logic Networks, importance spreading in Economic Attention Allocation)

This is engineering rather than AI ██████ work, but nevertheless involves many novel aspects and must be done with care. The OpenCog project currently involves contributors with a combination of OpenCog expertise and decades of experience in this area, e.g. Dr. Linas Vepstas.

4.3 Sensation (Vision, Audition & Haptics)

“Deep learning” is rapidly becoming the pre-eminent approach to the understanding of visual and auditory data; it is applicable to haptic (touch) data as well, though this has been less explored. In visual object and event recognition, deep learning systems are now outperforming narrower systems based on hand-coded feature extractors, for a variety of tasks. In speech-to-text, deep learning systems are now outperforming the Hidden Markov Model type systems used by major industry players.

OpenCog has been preliminarily integrated with the DeSTIN deep learning based computer vision system [1], originally developed by Dr. Itamar Arel at the University of Tennessee, Knoxville and now open sourced as part of the OpenCog framework. DeSTIN is currently being applied to robot vision utilizing input from stereo cameras and the Microsoft Kinect.

What is proposed here, during Phase 1, is further development of the DeSTIN system for vision, and extension to audition and haptics as well. Also, to achieve scalable performance, the current version of DeSTIN should be ported to CUDA for operation on GPUs (an earlier version of DeSTIN was ported to CUDA and dramatic speedup was achieved).

4.4 Robot Movement Control

Effective movement control is critical to robotics applications, but also valuable from a broader AGI perspective, because it provides a paradigm case of coordinated activity and dynamic planning. A great deal of human cognitive, social and linguistic activity is orchestrated via analogy to physical movement. In the human brain, the cerebellum is responsible for many additional sorts of planning and sequencing operations alongside motor control, all done using the same representations and mechanisms.

Standard humanoid or wheeled robotic architectures, with independently controlled motors associated with individual joints, are not especially well-suited for integration with AGI systems, as they lack the complex internal dynamics of biological bodies. An alternative is the “biomorphic” architecture, exemplified by the nervous-network approach pioneered by Mark Tilden, in which the motors in a robot body are controlled by a network of neurons or other comparable processing elements that pulse signals to each other in coordination with motor movements. In the biomorphic approach, the body is a complex dynamical system which may be coupled with the complex dynamics of a cognitive system – resulting in a “deep learning” approach in which the lower layer of learning occurs within the body via reinforcement learning in the network of pulsing elements; and the upper layers occurs in a cognitive system coupled with the body (such as OpenCog).

We do not propose novel hardware research, however, only novel research in interfacing cognitive control systems with biomorphic robotics. This work is best pursued in partnership with experienced robot control researchers. We are fortunate to have two leaders in this field as consultants and collaborators:

- Mark Tilden, pioneer of biomorphic robotics, former NASA roboticist, and creator of the RoboSapien line of toy biomorphic robots (more than 23 million sold)
- David Hanson, leader in creation of visually, emotionally and behaviorally human-like humanoid robots

The robot movement control aspect of this project could be carried out via a combination of in-house efforts and collaboration with appropriate academic robot labs such as (two of many possible examples):

- Peter [REDACTED] lab at the University of Texas, Austin
- Bertram Shi’s lab at Hong Kong University of Science and Technology, which has special expertise in neuro-morphic systems (and is located near Mark Tilden)

The overall goal during Phase 1 will be to create a robot capable of robust navigation, object and event recognition and object manipulation in an indoor environment. Generalization to a broader class of environments (e.g. city streets rather than the interior of a robot lab) will be carried out during Phase 2, alongside appropriate improvements to the robot hardware infrastructure (and bearing in mind that robotic hardware for sensing, moving and grasping improves each year).

4.5 Game World Development

While it is viable and interesting to experiment with early-stage AGI systems in commercial video game worlds, ultimately, to get the most from 3D simulated worlds for AGI learning, teaching and interaction, one needs to build customized virtual environments with the requirements of AGIs in mind. The main extensions to current game world technology needed are:

- More accurate modeling of the physics of non-rigid bodies, such as fabrics, pastes, fluids, etc.
- Integration of robot-simulator functionality with game-world functionality, to enable a “multiplayer robot simulator” with a complex simulated world

The latter functionality may be achieved by integrating an existing open source robot simulator (e.g. Gazebo is one option) with an existing open source game engine. The former functionality can be achieved by integrating known equations from the physics engine research literature with existing open source physics engines such as ODE.

No breakthrough research is required here, but this is a substantial piece of software engineering work that is not getting done because it doesn't provide an immediate commercial upside. Once available this technology will likely have a broad impact beyond the domain of AGI testing/training/teaching.

We have discussed this work extensively with Dan Miller, who integrated the ODE physics engine into the OpenSim virtual world, and has also done extensive humanoid robotics work with Anybots and Hanson Robotics; and he is interested in leading or participating in this aspect of the proposed work.

The capability of our Phase 1 AGI system to operate intelligently in a complex 3D game world will have wide commercial applicability in the gaming industry, in addition to its scientific value. In Phase 2 focus will be laid on the capability of the system to adapt to novel game worlds and learn spontaneously how the world operates and how to survive and flourish within it.

4.6 Genomics Data Analysis

One of the most exciting and impactful application areas for AGI technology will be the creation of artificial scientists. Analysis of scientific data and synthesis of novel data-driven scientific hypotheses, is a skill that AGIs can be expected to eventually master beyond the human level, since the human brain evolved to be specialized for quite different tasks than those involved in science. Furthermore, scientific analysis and hypothesis are somewhat different in nature from the other applications considered in this proposal, which have more to do with the control of embodied agents or the manipulation of formal structures like programs and mathematics. Thus it seems important to include an example of AI-based science as one of the initial applications pursued here.

The value of AI for genomics data analysis is well demonstrated in the literature, and OpenCog's MOSES component has been used for a number of successful genomics data applications, especially related to the genomics of longevity. What is proposed here is to take the next step, and push from the narrow AI style use of machine learning tools to analyze individual genomic datasets or small collections of genomic dataset, to the more general analysis of large volumes of genomic data en masse. This ultimately should be done across the scope of all genomics datasets available online, but we propose to begin specifically with aging and longevity related datasets, across multiple organisms. Specifically this requires:

1. Human effort to standardize and normalize a large set of aging-relevant SNP, gene expression and other genomic datasets into a common graph-database format. This has already been done for a significant number of datasets, in the GenAge database. Proteomic, metabolomic and other such data may also be integrated as appropriate.
2. Integration of OpenCog's natural language comprehension tools with existing open source software for extracting information from biological texts (e.g. specialized biological entity and relationship extraction code)
3. Human curation of an OpenCog Atomspace combining information from experimental datasets, information extracted from natural language, and information imported from structured biological knowledge bases such as the Gene Ontology and pathway databases

4. Tuning of OpenCog cognitive processes for effective pattern recognition, reasoning and learning on this biological data

Toward this end, we will benefit from advisement by informatics-savvy biologists who have collaborated with the Biomind LLC team on the application of OpenCog software to genomics data in the past, e.g.:

- Michael Rose, University of California, Irvine
- Joao Pedro de Magalhaes, University of Liverpool

Both of these scientists are recognized leaders in the biology of aging.

The Phase 1 AGI system will be an extremely powerful “biologist’s assistant”, with some capability to pose original hypotheses and test them via analysis of datasets. In Phase 2 the focus will be on teaching the system to display more autonomy in posing its own research questions and designing its own experiments.

4.7 Text, Image and Video Mining

While a human-level AGI requires its own experience of the world and needs to learn aspects of language via its own dialogic, social, embodied experience; nevertheless, there is a lot an early-stage AGI can learn from “reading” and analyzing the copious amount of text available online. This can be approached via the same tools used for language comprehension in a dialogue context, but requires significantly different “tuning” of the tools, as the usage of language in spoken conversation or text chat is significantly different from that in most text documents. Also, a scalable infrastructure for text mining has somewhat different requirements from an infrastructure for control of an AGI agent.

Alongside text, the Internet contains a massive amount of information in the form of images and videos, of potential interest to an AGI system. Information regarding what entities and events are depicted in images and videos can be used to provide knowledge to an AGI system, to aid its understanding of the world.

Carnegie-Mellon University has been conducting very interesting experiments in these directions, called NELL (Never-Ending Language Learning) and NEIL (Never-Ending Image Learning). These systems conduct information extraction from texts and images via a combination of machine learning algorithms. A very promising approach would be to integrate standard machine learning algorithms with OpenCog’s proto-AGI algorithms within this sort of overall information extraction framework.

The NELL/NEIL code is not currently open source, but the CMU team is potentially amenable to open-sourcing it if supplied with resources to “clean up” and professionalize their code (which is currently research-grade rather than production-grade). Alternately, if this hits a snag, the same functionality could be reimplemented using the designs described in their research papers.

As Phase 1 and then Phase 2 progresses, the system’s capability to understand text, images and video will improve, enabling it to extract more information from the Web. And of course, the additional information extracted will help the system to improve its intelligence yet further, in a virtuous cycle. What begins as relatively simple “information extraction” will gradually segue into genuine human-level understanding.

4.8 Automated Programming

Expressed in a minimalistic programming language, a typical cognitive control process could be implemented in a program containing less than 500 terms. This means that if an AGI program could create reasonably complex programs of this size, it would be able to systematically improve its own cognitive capabilities, and potentially embark upon a pathway of steadily self-improving intelligence. To get to this point however, we must first create proto-AGI systems that can learn to create simpler programs. OpenCog contains a powerful automated program learning subsystem, MOSES, which learns programs via probabilistic evolutionary learning. Currently MOSES can only learn relatively simple programs; however, the mathematical and software mechanisms needed to extend MOSES more generally have been clearly articulated.

What is proposed in this regard, technically, is:

1. Extend MOSES's internal representation language to encompass the full set of Mathematica/Mathics structures (Mathics is an open source analogue of Mathematica, enabling symbolic theorem-proving and numerical mathematics capabilities within a functional programming framework).
2. Extend MOSES's internal program analysis functionality to encompass higher-order functional programs (needed for Mathematica/Mathics), as outlined in *Engineering General Intelligence*
3. Connect MOSES's program reduction (simplification) module to PLN (which in turn will be connected to Mathics) to enable logic-based simplification of candidate programs during the course of program learning
4. Enable MOSES to observe the execution traces of running programs, so as to include this information in its modeling of programs (along with the data it currently models: program source code and input/output behavior)

As this particular application is closely tied to OpenCog's internal cognition processes, it is best carried out by a team closely integrated with the core OpenCog team. However, this work should also be carried out in close collaboration with Angus Griffith (current lead developer of Mathics), and the original developer of the MOSES algorithm, Moshe Looks, currently a Google employee.

The first priority for automated programming will be to get learning of simple algorithms like sorting, searching and graph traversal to work robustly, without problem-specific engineering as is generally done in the research literature on such topics. Following this, the next step will be automated learning of simple cognitive algorithms such as heuristic search and pattern mining.

By the time Phase 2 is reached, the assumption is that a fairly general program learning capability will have been achieved, and it will be possible to gradually lead the system through learning the programming exercises in a standard functional programming textbook. Rather than running program-learning experiments, one will be teaching the system to program.

4.9 Automated Theorem Proving

Automated theorem proving has the potential to play a critical role in the future of AI, because AI systems are formally describable using mathematics. Hence an AI system capable of robust theorem proving would be able to formally analyze its own behavior, and derive conclusions regarding its own functionality and how to optimize itself. Computer scientists use theorem proving to understand aspects of new algorithms they create; and potentially AGIs can do this as well, combining automated programming with automated theorem proving.

Specialized theorem proving software has already proved invaluable in mathematics, e.g. assisting with the proof of the characterization of finite simple groups, and resolving unsolved problems in logic. However, existing theorem-proving software is not yet capable of proving complex theorems on its own, without significant human guidance, nor of identifying interesting new theorems in need of proof. In order to overcome these weaknesses, two innovative steps are needed:

1. To enable large-scale analogical reasoning in the context of mathematical proofs. One needs to feed a large number of mathematical theorems and proofs into an AGI system's knowledge base, so that it can reason about each new theorem by analogy to others it has studied.
2. To ground a substantial subset of mathematical theorems in domains where an AGI system has direct experience. A simple example of this would be to ground basic geometric theorem proving in an AGI system's observations in a video game world – so it could prove theorems about objects and movements in the world, directly related to objects and movements it had observed itself

Toward the first point, preliminary work has been done regarding loading the Mizar corpus of mathematics theorems and proof (which covers all mathematics up to the Masters degree level, plus more) into OpenCog's AtomSpace.

Given these steps, one can then carry out automated theorem proving in OpenCog using an integration of OpenCog's current PLN (Probabilistic Logic Networks) system with an external theorem proving engine such as Mathics.

Predrag Janicic, a world authority on automated theorem proving, has worked with OpenCog in the past along with his graduate students, and has expressed interest in assisting with this aspect of the proposed work.

As Phase 1 segues into Phase 2, the system will be asked to prove more and more complex theorems, generalizing further and further beyond the game world domain that it uses to ground its knowledge of set-theoretic and geometric mathematics. By the end of Phase 2, the system should be acting as an innovative mathematician on its own, though perhaps with different strengths and weaknesses than human mathematicians.

4.10 Teaching and Intelligence Testing

How can one measure incremental progress toward adult human level AGI? IQ assessment instruments as presently defined are overfit to the specific of the human mind and body and not applicable to AGI systems with slightly different capabilities. However, it is possible to design "AIQ" tests capturing the essential concepts of human IQ tests but oriented toward AGI systems that control virtual or robotic agents rather toward humans. Such tests have been outlined already, but fleshing out the details and implementing them in software will require collaboration between a psychologist trained in child intelligence assessment and a small group of programmers.

Well-designed, solidly implemented AIQ tests will be valuable beyond the scope of this particular project, and are likely to play a valuable role in moving the AI and AGI fields forward more broadly.

4.11 Software Integration and Testing

Any project involving multiple teams developing complex software for multiple overlapping purposes, presents subtle software integration and testing issues. A small dedicated team devoted to integrating, maintaining and testing the collective codebase of the project will be critical. This team can also mediate the contributions of volunteer open source developers and participating external academic researchers, which may become substantial as the project evolves.

References

- [1] 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*, 2009.
- [2] Joscha Bach. *Principles of Synthetic Intelligence*. Oxford University Press, 2009.
- [3] G. Fauconnier and M. Turner. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic, 2002.
- [4] B. Goertzel, I. Goertzel M. Iklé, and A. Heljakka. *Probabilistic Logic Networks*. Springer, 2008.
- [5] Ben Goertzel. *The Hidden Pattern*. Brown Walker, 2006.
- [6] Ben Goertzel, Lucio Coelho, Mauricio Mudado, and Cassio Pennachin. Identifying the genes and genetic interrelationships underlying the impact of calorie restriction on maximum lifespan: An artificial intelligence based approach. *Rejuvenation Research* 11(4): 735-748, 2008.
- [7] Ben Goertzel and et al. Combinations of single nucleotide polymorphisms in neuroendocrine effector and receptor genes predict chronic fatigue syndrome. *Pharmacogenomics*, 2005.
- [8] Ben Goertzel and et al. The cogprime architecture for embodied artificial general intelligence. In *Proceedings of IEEE Symposium on Human-Level AI, Singapore*, 2013.

- [9] Ben Goertzel and Cassio Pennachin Et Al. An integrative methodology for teaching embodied non-linguistic agents, applied to virtual animals in second life. In *Proc.of the First Conf. on AGI*. IOS Press, 2008.
- [10] Ben Goertzel, Cassio Pennachin, and Nil Geisweiller. *Engineering General Intelligence, Part 1: A Path to Advanced AGI via Embodied Learning and Cognitive Synergy*. Springer: Atlantis Thinking Machines, 2013.
- [11] Ben Goertzel, Cassio Pennachin, and Nil Geisweiller. *Engineering General Intelligence, Part 2: The CogPrime Architecture for Integrative, Embodied AGI*. Springer: Atlantis Thinking Machines, 2013.
- [12] Ben Goertzel, Hugo Pinto, Cassio Pennachin, and Izabela Freire Goertzel. Using dependency parsing and probabilistic inference to extract relationships between genes, proteins and malignancies implicit among multiple biomedical research abstracts. In *Proc. of Bio-NLP 2006*, 2006.
- [13] Ben Goertzel, Joel Pitt, Zhenhua Cai, Jared Wigmore, Deheng Huang, Nil Geisweiller, Ruiting Lian, and Gino Yu. Integrative general intelligence for controlling game ai in a minecraft-like environment. In *Proc. of BICA 2011*, 2011.
- [14] Ben Goertzel, Joel Pitt, Matthew Ikle, Cassio Pennachin, and Rui Liu. Glocal memory: a design principle for artificial brains and minds. *Neurocomputing*, April 2010.
- [15] Ben et al Goertzel. Opencogbot: An integrative architecture for embodied agi. *Proc. of ICAI-10, Beijing*, 2010.
- [16] Moshe Looks. *Competent Program Evolution*. PhD Thesis, Computer Science Department, Washington University, 2006.
- [17] Rafal Smigrodzki, Ben Goertzel, Cassio Pennachin, Lucio Coelho, Francisco Prosdoscimi, and W Davis Parker. Genetic algorithm for analysis of mutations in parkinsons disease. *Artif Intell Med.*, 2005.
- [18] Endel Tulving and R. Craik. *The Oxford Handbook of Memory*. Oxford U. Press, 2005.