

OpenCog
OpenCog Wiki

Roadmap

- Page
- Discussion
- View source
- History

Search

Roadmap

From OpenCog

This is detailed roadmap for the next two years of OpenCog development (2011-2012).

- This roadmap coordinates loosely with the time-lines of several separate projects that utilise OpenCog code and that will be contributing to assist reaching these milestones.
- Most of the milestones given here should be achievable given current funding; but some are dependent on funds intended to be raised from the 2011 fundraising drive. The latter are explicitly explained below but broadly these milestones are related to computer vision and language processing. The immediate goal of our fundraising drive is to hire staff in these two areas.
- We've used version numbers for each major milestone, although these will not necessarily correspond to public code releases.

The main contributing projects mentioned above are:

- Hong Kong Polytechnic University, HKSAR ITF grant. This project, co-funded by the Hong Kong government and Novamente LLC, is building upon prior work done by Novamente LLC to develop intelligent agents within virtual worlds. The project will create emotive game characters that can speak as well as understand English. The characters will also learn from their environment and be able to socialise between themselves to share knowledge and convey their emotive state. The end product of the project will be an API that assists game developers in creating dynamic characters for their game worlds, powered by the OpenCog engine.
- Novamente LLC, which is pursuing consulting projects involving spatio-temporal reasoning, language processing and meta-learning, that will contribute to development of OpenCog code in these regards.
- Biomind LLC, which will integrate OpenBiomind with OpenCog with a goal of using the latter's probabilistic inference functionality to improve its genomic data analysis.
- Xiamen University's BLISS (Brain-Like Intelligent Systems) lab, which is collaborating with the Hong Kong project mentioned above, has sponsored students working on language generation for

OpenCog. Currently BLISS is also sponsoring work on DeSTIN-based vision processing with a view toward integration with OpenCog.

Contents

- 1 OpenCog v0.4
 - 1.1 Technical Foci
 - 1.2 Details
- 2 OpenCog v0.5
 - 2.1 Technical Foci
 - 2.2 Details
- 3 OpenCog v0.6
 - 3.1 Technical Foci
- 4 OpenCog v0.7
 - 4.1 Technical Foci
- 5 OpenCog v1.0
 - 5.1 Technical Foci
 - 5.2 Details

OpenCog v0.4

internal milestone 15th July 2011

Learning, Reasoning, Language and Emotion in a Game World

This milestone corresponds to demonstrations of OpenCog's virtual agent control system for the AGI-11 conference at the Mountain View GooglePlex.

This milestone will demonstrate a single OpenCog-controlled avatar learning from a human player about a simplistic block world. The world will contain blocks of several types and will require the avatar to use imitation learning, transfer learning, and action planning to solve several challenges to fulfil internal goals. As well as learning new behaviors, the avatar will be able to use and understand rudimentary English in order to:

- ask and answer questions about the world
- express its internal emotional state and motivations

The outcome of this milestone will be a 5-10 minute video demonstrating the above points. There will also be a live demo, but it's not expected to be particularly robust at this stage.

It's important to understand the differences between this project and "good old fashioned AI" style projects using blocks worlds and other similar toy scenarios. Our focus is on adaptive, interactive learning, not on execution of preprogrammed behaviors. Thus, we aim to use these simple scenarios to

teach our learning systems things that it can then generalize to other richer and broader scenarios. Whereas if one achieves certain intelligent-looking behaviors in a simple scenario using rigidly preprogrammed rules, one does not have a system that uses the simple scenario to acquire (declarative, procedural, episodic, attentional, etc.) knowledge that can be automatically extended to other more interesting domains.

As well as demo-focused improvements, this release will also include some advances not directly related to the demo, e.g.: a version of the DeSTIN vision system, configured to export to OpenCog the results of its analysis of images or videos; and the capability to produce descriptions of the 2D spatial relationships between entities observed in spatial scenes.

Technical Foci

The main technical foci during this period will be:

- Integration of OpenCog with the Unity3D game engine.
- Completion of the OpenPsi framework for emotion and motivation.
- Modification of the imitation/reinforcement learning process to support learning based on environmental reward rather than just explicit teacher-delivered reward.
- Integration of probabilistic inference with perceptual data, to allow reasoning about the world.
- Implementation of simple temporal planning functionality.
- DeSTIN code cleanup and interfacing and reimplementing of portions of DeSTIN on a GPU supercomputer.
- Extraction of a variety of spatial relationships from maps represented inside OpenCog.

Details

See a rough draft script of the demo video here. This is expected to evolve significantly as the work proceeds, but nevertheless serves as an indication of the level of intelligence we're after during this phase.

OpenCog v0.5

Public alpha release, 1st October 2011

Robust Game-World Intelligence + Binary Release

The primary thrust of effort between v0.4 and v0.5 will be to make the functionality demonstrated in the v0.4 demo more robust. Experimentation with a variety of behaviors similar to the ones demonstrated will be conducted, and adjustment will be made as needed to improve system intelligence and performance. Some algorithmic improvements will also be made, e.g. integration of MOSES alongside hill-climbing for behavior learning (MOSES is integrated with OpenCog but not currently used for learning virtual-world behaviors).

Effort will also be devoted, in this interval, to the creation of an OpenCog binary release and associated tutorial material. OpenCog was founded in 2008, and although the project has made steady progress

since its inception, it's important to make an official binary release for researchers who are curious about OpenCog and to generally make it easier for newcomers to become involved with the project. The result of this effort will be an influx of new contributors and a coherent set of tutorials to walk them through using the various modules within OpenCog.

The release will also include: inference of more complex spatial relationships between observed entities (and groups of entities), based on directly observable spatial relationships; and an improved DeSTIN system with greater accuracy and more scalable performance due to the underlying use of a GPU supercomputer.

Technical Foci

- Comprehensive test scenarios for the functionality developed for the initial demo.
- Full support for use of MOSES withing the embodiment LearningServer.
- Augmentation of hill-climbing with MOSES for behavior learning
- A binary and developer Ubuntu/Debian package with appropriate dependencies and Launchpad PPA for any non-default packages.
- A revised Python API.
- Better persistence scheme for the AtomSpace.
- Implementation and tuning of spatial inference rules.
- Aspects of natural language processing, specifically the semantic mapping provided by RelEx2Frame, will be migrated into core OpenCog system.
- Tuning of DeSTIN vision system for improved performance on GPUs.

Details

- Create a Debian package that users can download and install. Package external dependencies where necessary and place in a Launchpad PPA. Also place a "download" page on the main opencog.org website.
- Provide a Pythonic API to OpenCog that allows simple access to the core modules. This will open up OpenCog and the AtomSpace to a large section of the academic research community who tend to avoid C++. Allow MindAgents written in Python that can be dynamically loaded by the CogServer.
- Separate OpenCog into modular components that can be used by Python bindings. For example, each module should be able to connect to a specific AtomSpace instead of using the CogServer's global AtomSpace.
- Polish and improve the web and REST interface so that OpenCog is accessible to any language with a REST library, and that researchers can interact with OpenCog in their browser immediately after installing it. Focus on PLN and initiating inferences.
- Provide a taskboard within the CogServer where Agents can post requests to be fulfilled by other Agents in exchange for STI/LTI.
- Implement a simple disk store for the AtomSpace using SQLite. OpenCog will save/update the AtomSpace periodically and save the AtomSpace on exit. The CogServer will automatically reload the AtomSpace when it restarts.

OpenCog v0.6

1st March 2012

Hide and Seek: Mental Modeling of Self and Others + Robust Dialogue + Computer Vision

This release will feature a significantly richer and more exciting demo, centered on control of a virtual agent who plays hide-and-seek in a richly-featured blocks world. Hide-and-seek requires a fairly sophisticated level of mental modeling and we will allow agents to build structures with blocks in-between rounds of hide-and-seek. This building behaviour will enable them to create more interesting places to hide, and give ample options to demonstrate spatial inference and planning.

Achieving this will involve a number of technical improvements. Notably, the fine-tuning of the Economic Attention Networks (ECAN) component for regulating system attention, and the integration of ECAN with PLN to provide attention-guided inference control.

At the same time, the natural language dialogue system will be further fleshed out, adding additional conversational control mechanisms; PLN will be extended to handle the full spectrum of temporal inference rules and formulas; and the integration of DeSTIN with OpenCog will be completed in a more general and less primitive way, using machine learning to automatically create concepts in OpenCog corresponding to DeSTIN's internal patterns.

Technical Foci

- Tuning of OpenPsi and PLN to support rich modeling of self and others.
- Tuning of ECAN, and implementation/tuning of ECAN-based inference control.
- Completion of PLN temporal inference system.
- Spatiotemporal inference and planning for more complex physical construction operations (building structures).
- Flexible natural language dialogue covering a variety of rhetorical modes and situations.
- Moving more natural language parsing/generation code into the core OpenCog system, enabling greater cognitive flexibility and experiential language learning.
- Completion of machine-learning based layer for flexible interfacing of DeSTIN and OpenCog.

OpenCog v0.7

1st June 2012

Robust Mental Modeling of Self and Others + Robust Computer Vision

This release will offer largely similar functionality to v0.6, but with greater robustness and flexibility, due to testing on numerous examples and addressing of issues thus encountered.

Technical Foci

- Tuning and refinement of spatiotemporal inference rules.

- Extension of dialogue system to include richer conversation about mental states of self and others.
- Tuning of machine-learning based layer for flexible interfacing of DeSTIN and OpenCog.

OpenCog v1.0

31 December 2012

Version 1 OpenCog Engine for AI Development and Non-Player Character Creation

An appropriately marketed OpenCog 1.0 release, given the amount of time it's been in the making, should generate a fair amount of new interest from people who haven't heard of it before. The main goals are to:

- provide a robust engine for dynamic learning supporting parallel mind agents and fully supported automatic forgetting and loading of atoms from the backing store.
- provide an easy-to-use framework for creating and configuring OpenCog-controlled intelligent virtual agents living in a virtual world.

As well as improvements to the underlying OpenCog framework, some significant improvements to the AI code will be made during this period, including the integration of MOSES and PLN so as to allow each of them to learn from the others' results.

Technical Foci

- Robust forgetting and reloading of Atoms from the backing store with ECAN to guide caching.
- Integration of MOSES and PLN.
- Support for parallel MindAgents in their own threads or processes.
- AtomSpace API request prioritisation.

Details

- Support robust automatic forgetting and reloading from the backing store when accessed. Utilise attention values from ECAN to support intelligent caching of AtomSpace knowledge.
- Redesign CogServer so that MindAgents can choose to run continuously in their own thread/process or have their run() method called approximately every X seconds.
- Allow AtomSpace to prioritise requests. This will require callers of the AtomSpace API to register for a caller ID. Requests are signed by this ID. Also provide an interface to get statistical summaries of MindAgent access to the AtomSpace.
- MOSES can learn from the historical behaviour of backwards inference trees in PLN to guide the generation of knowledge by forward chaining using PLN rules.

Optional:

- Implement a distributed AtomSpace on top of a massively scalable key/value store like Riak, Cassandra, or other NoSQL scalable database. Allow multiple CogServers to share this distributed backing store.

Retrieved from "http://wiki.opencog.org/w/Roadmap"

- Navigation
 - Home
 - About
 - FAQ
 - Development
 - Project Status
 - Topic Categories
 - Recent Changes
- Community
 - The Team
 - Donate
 - Get Involved
 - Events
- Toolbox
 - What links here
 - Related changes
 - Special pages
- Personal tools
 - Log in / create account
- 
OpenCog 
- 141
-  11 Developers

This page was last modified on 5 March 2011, at 13:07. This page has been accessed 836 times.

- Content is available under GNU Free Documentation License 1.2.
 - About OpenCog
 - Disclaimers