# Opensim4OpenCog

User Documentation

rev 0.1

Tagline

Cogbot is a framework for developing intelligent robots in real and virtual worlds.

While Cogbot is suitable for serious applications in robotics and AI, beginning with Cogbot is fun and easy.

AI researchers looking for a platform that integrates many standard components including embodiment will find Cogbot the best developed offering available.

Software engineers creating presences in virtual worlds will find Cogbot the obvious choice.

Educators will find Cogbot a useful teaching tool.

Virtual world users will find Cogbot a cool way to make bots even with minimal or no programming experience. Educators will find Cogbot a useful teaching tool. Content creators will find Cogbot useful for automating tasks in shop and studio. Cogbot bots can be NPC's in RP Sims. If it needs a bot, it needs Cogbot.

# Summary

It is widely acknowledged that an AGI requires a rich environment with which to interact. However, real robots suffer from a number of infelicities, from sticky servos to lack of reproducibility. A virtual robot operating in a virtual world is more controllable. At the same time, the environment must be large, complex, and unpredictable. Fortunately, virtual worlds provide such an environment. Second Life, with its large size, user created content, and lack of a clear, unidirectional goal, serves well.

Second, we expect Opensim4OpenCog to serve a wide variety of users outside the immediate AGI community, as a robust, flexible, intelligent tool for constructing automatic 'bots' in virtual worlds.

Cogbot is a virtual robot for Second Life and OpenSim based virtual worlds. Our original area of application was for AGI research, but Cogbot is useful for many VW bot applications.

Capabilities:

* Encapsulates the complexity of dealing with libOMV and the streaming SL protocol. Maintains an object list. So an external program can, for example, ask for 'all red balls within 10 meters'.
* Provides a flexible mechanism for interacting with the bot via text commands over HTTP or sockets.
* Provides a bridge from C# to swi-Prolog
* Integrates Lucene, Cyc and AIML with the knowledge of objects in the world.
* Integrates with the Radegast viewer to manually control the bot. Many applications require some amount of manually 'driving' the bot.

Cogbot/Opensim4OpenCog is a server written in a mixture of C#, Prolog, and dot Lisp that connects to an avatar acting as "bot" in OpenSim/Second Life/RealXtend.

Opensim4OpenCog is likely to be useful to two audiences.

First, Opensim4OpenCog is being built to serve as a virtual robot for the OpenCog project.

It is widely acknowledged that an AGI requires a rich environment with which to interact. However, real robots suffer from a number of infelicities, from sticky servos to lack of reproducibility. A virtual robot operating in a virtual world is more controllable. At the same time, the environment must be large, complex, and unpredictable. Fortunately, virtual worlds provide such an environment. Second Life, with its large size, user created content, and lack of a clear, unidirectional goal, serves well.

Second, we expect Opensim4OpenCog to serve a wide variety of users outside the immediate AGI community, as a robust, flexible, intelligent tool for constructing automatic 'bots' in virtual worlds.

# Architecture

The Second Life Messaging Protocol[1](" \l "sdfootnote1sym), is a de facto standard protocol for operating a 'grid' of 'simulators', servers serving pieces of virtual land, along with support services. This protocol is supported by the Second Life(tm) service and by grids operating the OpenSim server.

The protocol is complex and involves connecting to a number of different actual server processes, but for our purposes we can simply think of a server connecting to a viewer.

Typically a viewer is a GUI client that displays the user's view of the world and allows them to control their avatar.

Opensim4OpenCog is a viewer. Unlike a normal viewer, Opensim4OpenCog is oriented towards connecting with another program and being run as a bot. Thus Opensim4OpenCog also acts as a server, and can be communicated with in a number of ways (see 'Interacting with Opensim4OpenCog') programmatically.

Opensim4OpenCog is a plugin extension to the Radegast[2](" \l "sdfootnote2sym) viewer, a viewer intended for interacting in situations where a 3rd person POV 'game' display is inappropriate. Radegast provides os4oc with the ability to control the bot manually.

# System Requirements

# Installation

<TODO>

Looks like we're going to do an installer or some packaging method

## Building from sources

## VS2008

### repository is requesting password

anonymous

no pw

## Configuration

Botconfig.xml

# Running Opensim4OpenCog

<TODO>

(you can add a startup command to botconfig.xml so you can consult a file)

Command line args?

# Interacting with the bot

## User Interface

At startup os4oc will open three windows.

os4oc is integrated into the SWI-Prolog[3](" \l "sdfootnote3sym) development environment. The normal SWI-Prolog integrated development environment is available during cogbot programming.

It is frequently necessary, during bot development, to operate the bot by hand. For this reason, cogbot operates as a plugin inside the Radegast viewer. Radegast displays its own window.

Other displays -

Optionally, the user can request images of objects in the radegast window. These are supplied via LookingGlass.

The pfdebug command opens a debug panel useful for understanding the bot's pathfinding behavior.

Douglas says its' too wonky for public consumption, don't advertise

/pfdebug

## Command line

! and $ in commands

$master

$selected (group of selected prims)

primid is local id

@ coerces prim position to 3d location @primid 15

@: Restrict the following commands to one or all avatars.

max is maximum prims in filterset

dist is maximum dist to avatar

/priminfo $selected maxdist 2

the selected prims (that is, the prims in my 'selected' grasper) tha are within 2 meters of my AV

See all this stuff starting at line 445 of WorldCommandParsing

/priminfo $selected maxdist 2.1 mindist 1.9 max 4 nth2 matches Window !matches -like ownedby $master

it must have been previously selected, 1.9 to 2.1 meters from the bot, no more than 4 returns, only the second object, that has the word Window in it's description or name\*, and does not match "-like"

family - linkset from prim

\* the name and description are in the form

name | descript

follow\*

/swip X is 1 + 1, botClientCmd(say(X)).

see Cogbot.listeners.WorldCommandParsing for list of prepositions and variables

location formats

sim

[13:19] BinaBot Daxeline: location formats simname/x/y/z

[13:19] BinaBot Daxeline: location formats x/y/z

[13:20] BinaBot Daxeline: location formats "Annie Obscure"

[13:20] BinaBot Daxeline: location formats "Annie"

[13:20] BinaBot Daxeline: location formats <x,y,z>

[13:20] bungiecord Burnstein: with a space

[13:21] bungiecord Burnstein: so <4.0, 5.0, 18.0>

[13:21] BinaBot Daxeline: location formats "ObjectNAme spec" 1

[13:21] bungiecord Burnstein: if it matches more than one

[13:21] BinaBot Daxeline: location formats nth 1 "ObjectNAme spec"

[13:23] bungiecord Burnstein: at symbol

[13:23] bungiecord Burnstein: angle and distance

[13:33] BinaBot Daxeline: ok /moveto @0\*1

[13:33] BinaBot Daxeline: ok /moveto @90\*2

## Programmatically

### TCP to 5550(check #)

##### TCP Level controlsbye - hangs up

##### hideon - no low level events

##### hideoff - no hideon

##### filters - lists the filters

##### always - makes sure you always get the event + add - remove from always

##### never - makes sure you never get the event + add - remove

### HTTP to 5580(check #)

TODO - document later when http is more stable

Two purposes,

Get info

Issue cmds

http doesn't have the garbage

### Prolog

#### Acting on the world

:-use\_module(library(clipl)).

?-runSL.

At swipl top levelquery

?- Say('hi there').

Or

botClientCmd('say hi there', X).

:-use\_module(library('simulator/cogrobot')).

Annie - look in botcmdlist for the commands

#### Receiving information from the world

botClientCmd('priminfo nearby 10',X).

simAvatar(X),cliGet(X,'name',Name).

#### API

##### runSL/0,

Makes sure the bot is running. Logs bot on at thebotconfig.

##### worldSystem/1, worldSystem/2,

##### botClient/1, botClient/2,

##### botClientCall/1, botClientCall/2,

##### botClientCmd/1, botClientCmd/2, botClientCmd/3,

##### (create-botclient "Douglas" "Miles" GlobalPasswd "http://107.7.21.240:8002/" "last")

##### simObject/1,

##### simAvatar/1,

##### simAvDistance/3,

##### simAsset/1,

##### simAccount/1,

##### gridClient/1,

##### resolveObjectByName/2,

##### vectorAdd/3,

##### distanceTo/2,

##### toGlobalVect/2,

##### toLocalVect/2,

##### onSimEvent/3,wasSimEvent/3,

##### obj2Npl/2,

##### npl2Obj/2,

##### chat/1,

##### chat/2,

##### chat/3,

##### createWritelnDelegate/2,

##### createWritelnDelegate/1,

##### textureIDToImage/2,

##### textureIDToImageParts/2,

##### requestTexture/1,

##### simObjectColor/2

##### cliCall/3,

##### cliGet/3,

##### cliSet/3,

##### cliToData/2,

##### cliWriteln/1,

[2:40:38 PM | Edited 2:40:50 PM] Douglas R. Miles: simObject(X), toLocalVect(X,P).

#### cliinterface

cli\_debug/1,

cli\_Eval/3,

cli\_GetSymbol/3,

cli\_Intern/3,

cli\_IsDefined/2,

cliAddEventHandler/3,

cliAddLayout/2,

cliArrayToTerm/2,

cliArrayToTermList/2,

cliCall/3,

cliCall/4,

cliCol/2,

cliCollection/2,

cliFindClass/2,

cliFindMethod/3,

cliFindType/2,

cliGet/3,

cliGetRaw/3,

cliGetType/2,

cliIsNull/1,

cliIsObject/1,

cliIsType/2,

cliLoadAssembly/1,

cliMemb/2,

cliMemb/3,

cliMembers/2,

cliNew/3,

cliNew/4,

cliPropsForType/2,

cliSet/3,

cliSetRaw/3,

cliSubProperty/2,

cliShortType/2,

cliSubclass/2,

cliToData/2,

cliToData/3,

cliToFromLayout/3,

cliToString/2,

cliToStringRaw/2,

cliToTagged/2,

cliTypeSpec/2,

cliWrite/1,

cliWriteln/1,

cliUnify/2,

cliWithLock/2,

cliEnterLock/1,

cliExitLock/1,

cliNewDelegate/3,

link\_swiplcs/1,

to\_string/2,

cliToFromRecomposer/4,

cliWriteFormat/3,

cliWriteFormat/2

### C#

ask doug for rules for $ and ! In his cmd interpreter

# Code Structure

AIMLbot -

not loaded (commented out iin botconfig), makes the bot listen on chat and respond with aiml interpreter

CSProlog – C# based Prolog interpreter, is only currently loaded by AIMLBot, which loads it but doesn't do anything useful with it.

DotLisp – always loaded used for the event filtering and interpreting botconfig.xml

IKVM - module that allows .net to access jar files

Lucene – natural language database used by AIMLbot

MushDLR223 -

shared utility lib shared between cogbot and it's modules

general purpose utilities

supposed to be a scripting framework

PathSystem3D -

pathfinder, not dependent on cogbot. Cogbot depends on it

SharpWordNet – C# lib for WordNet, a part of speech database. LAIR accesses it.

LAIR – bunch of C# utilities that do language parsing (tells what are nouns, verbs, adjectives) , used by AIMLBOT

Cogbot.LAIR – abstraction layer on LAIR so LAIR can be removed from cogbot.

LibOMV – library to deal with SL Messaging Protocol

LookingGlassClient – visualization library for Radegast

RadegastClient – this is Radegast

SwiProlog – SWI prolog interpreter

SwiPlCs – allows swi-prolog to access .net libs

PrologScriptEngine – unified architecture for accessing Prolog from C#

swicli – the swi prolog module that hosts .net runtime

cogbot32 – allows cogbot to act like it's on a 32 bit machine

cogbot.library

TheOpenSims – all the objects that cogbot uses (name is misleading) – meshes, objects, avatars

Tutorials- a set of classes to give the bot some behavior that teaches one how to use it

[1](" \l "sdfootnote1anc)http://wiki.secondlife.com/wiki/Protocol

[2](" \l "sdfootnote2anc)http://radegast.org/wp/

[3](" \l "sdfootnote3anc)http://www.swi-prolog.org/

# ==================

# This stuff is from the thisweeknotes.txt document. I'm trying to get everything in one document

Not everything below here is in the docs yet

not clear if prolog install should be in

"C:\development\opensim4opencog\pl"

needs context!

some prolog readme's in os4oc dir

in INSTALL.txt this is borked

set PATH=C:\development\opensim4opencog\bin;C:\Program Files (x86)\TortoiseSVN\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem

"C:\Program Files (x86)\Microsoft Visual Studio 9.0\Common7\IDE\devenv.exe"

(not to mention that all this is massively invasive)

in compile.bat

call runprebuild2008.bat

the file is called runprebuild.bat

tcp commands are passed to bot, which doesn't understand them

what's IKVM again?

.net interpreter for java classes

Need the commands (there were 2) to control debug info at the telnet level

need the enclosing format for messages to the bot

is it /help or hey you, do help, or

tcp level

bye - hangs up

hideon - no low level events

hideoff - no hideon

filters - lists the filters

always - makes sure you always get the event + add - remove from always

never - makes sure you never get the event + add - remove

this is in BotTcpServer::EvaluateCommand

repository is requesting password

anonymous

no pw

im the bot

cmd help

simAvatar(X),cliGet(X,hasprim,@(true)),cliToString(X,S).

[16:39] Nephrael Rajesh: botClientCmd(shout("hi there!")).

[16:39] bungiecord Burnstein shouts: hi there!

[16:40] Annie Obscure: apropos(botClientCmd).

[16:41] Nephrael Rajesh: simAvatar(X),cliGet(X,hasprim,@(true)),cliToString(X,S).

[16:43] Nephrael Rajesh: botClientCall(executeCommand("jump"),X),cliWriteln(X).

gridClientEvents(E),writeq(E),nl,fail.

pfdebug

! and $ in commands

$master

$selected (group of selected prims)

primid is local id

@ coerces prim position to 3d location @primid 15

max is maximum prims in filterset

dist is maximum dist to avatar

/priminfo $selected maxdist 2

the selected prims (that is, the prims in my 'selected' grasper) tha are within 2 meters of my AV

See all this stuff starting at line 445 of WorldCommandParsing

/priminfo $selected maxdist 2.1 mindist 1.9 max 4 nth2 matches Window !matches -like ownedby $master

it must have been previously selected, 1.9 to 2.1 meters from the bot, no more than 4 returns, only the second object, that has the word Window in it's description or name\*, and does not match "-like"

family - linkset from prim

\* the name and description are in the form

name | descript

see Cogbot.listeners.WorldCommandParsing for list of prepositions and variables

follow\*

/swip X is 1 + 1, botClientCmd(say(X)).

?- set\_prolog\_flag(double\_quotes,string). <-- cogbots usually this

set\_prolog\_flag(double\_quotes,chars). <-- atoms

set\_prolog\_flag(double\_quotes,atom). <-- "foo" == foo

periscope - used to replicate everything in an opensim

simian server

periscope

@'C#663540224'"TextureIDs"

how to get the members

simObject(O),cliMembers(O,M).

[11:58:42 AM] Anne Ogborn: whats with the !! and ! versions of commands?

[11:59:54 AM] Douglas R. Miles: if it tries to load a command with teh same name more than omnce.. it puts the secondary command with a ! in front

[12:00:19 PM] Anne Ogborn: kk. Why's it loading commands more than once?

[12:00:37 PM] Douglas R. Miles: oh jus thte same by name

[12:00:51 PM] Douglas R. Miles: depending on which plugin systme loaded it

[12:00:56 PM] Anne Ogborn: ah

[12:01:24 PM] Douglas R. Miles: TextSL, Radegast, LibOMV, Cogbot all did a serioes of comnands

[12:01:46 PM] Douglas R. Miles: stop-flying came from textSL for example

[12:01:57 PM] Douglas R. Miles: where libomv never made one

[12:03:00 PM] Douglas R. Miles: ok soimething about invoentry commnands that cogbot implments.. is it uses a command shell approach to folders

[12:03:20 PM] Douglas R. Miles: you can type /ls

[12:03:37 PM | Edited 12:03:54 PM] Douglas R. Miles: or /cd somefolder

to get the debug window for the path finding

[1:20:48 PM] Douglas R. Miles: as /pfdebug

To register for an event

gridClient(Obj),cliGet(Obj,'Self',S),cliAddEventHandler(S, 'ChatFromSimulator',onChat(\_,\_,\_)).

Figuredout 'ChatFromSimulator' by running listMembs (look at it's source in runcogbot.pl) one line at a time.

figured out arity of onChat by guessing apparently they're all arity 3

onChat(Event originator, Event Sender , event itself)

what you get

onChat(@'C#534101280',@'C#534101280',event('ChatEventArgs',@'C#625254248',"flaming numnums",enum('ChatAudibleLevel','Fully'),enum('ChatType','Normal'),enum('ChatSourceType','Agent'),"Anne Ogborn",uuid("f09015a4-41ca-4717-a98b-0ac31f58ddad"),uuid("f09015a4-41ca-4717-a98b-0ac31f58ddad"),v3(148.46278381347656,156.40774536132812,1000.9852294921875)))

read\_term (prolog lib pred) with catch around it

test the case where the goal is just beyond a wall

file\_search\_path to add a directory to searhc path

09:32] Douglas Miles: 1 ?- botClient([name],X).

X = "Dogbert Miles".

[09:32] Douglas Miles: (is one way to get the bot's name)

[09:32] Anne Ogborn: ah, sweet, lovely, ok

[09:33] Douglas Miles: though remmber its a string and the prim emote a atom

[09:34] Douglas Miles: :- style\_check(-discontiguous).

[09:35] Douglas Miles: ah.. i must have started my email to you and never pressed 'send' the otehr day

[09:35] Anne Ogborn: yes, I didn't get anything

[09:35] Douglas Miles: :- style\_check(+singleton).

:- style\_check(-discontiguous).

:- style\_check(-atom).

:- style\_check(-string).

================

This stuff is from openoffice Cogbot User Documentation.odt

Commands syntax, from

<http://code.google.com/p/opensim4opencog/source/browse/trunk/sources/main/LibCogbot/Listeners/WorldCommandParsing.cs?r=2086#433>

and Douglas

prims = GetNearByObjects(TheSimAvatar.GlobalPosition, TheSimAvatar, dist, false);

public List<SimObject> GetNearByObjects(Vector3d here, object except, double maxDistance, bool rootOnly);

verbs

(very incomplete)

i also have /select and /unselect cmds

[5:47:57 PM] Douglas R. Miles: /unselect $selected - clears all selections

[5:48:10 PM] Douglas R. Miles: /select $region... all object is region

The nouns

(probably incomplete)

you can add nouns by

provideGroup(\_,\_) had soemting defined like provideGroup(anniesgroup,X):..

$master = group of master

$region = region you arte in

$all = ALL

$selected

a name

a UUID

the conjunctions

and or comma - union the results

The adjectives

adjectives can be combined, a la

/priminfo maxdist 10 mindist 5

/priminfo dist 10

/priminfo maxdist 10

limit to within 10 meters from the bot

/priminfo mindist 10

limit to more than 10 away

/select family $selected

if you had two objects selected it gets them and all their children

parentof - the parent of the argument

max - limit to max number of objects

nth – return only the nth match

matches -

matches is sort of the "default'

thats the mode the command parser thinks it in at onset

i have this matchString that all objects present

name | desc ShapType ID (localid x### )

childsof

ownedby

bydist sorts it's argument in ascending order

anything else is the

bydistance will "sort" the prims:

an example of "getting the two closest prims owned by someone "

/select max 2 bydistance ownedby 34-3434--343434-344

[6:46:17 PM] Anne Ogborn: yah, got that

[6:46:35 PM] Anne Ogborn: oh, good to say it can be name or UUId

------------------- radegast

Few things that are on master or friends list - autoaccept objects and tp's

Botcmds in im session

All cogbots I'm master in -

/say local

=========================

Starting with a prolog top level

[2:11:45 PM] Douglas R. Miles: cd c:\pathtoCogBot\

[2:12:09 PM] Douglas R. Miles: swipl-win.exe -f prolog/cogbot.pl

[2:12:14 PM] Anne Ogborn: cool

[2:12:48 PM] Douglas R. Miles: there is a c:\pathtoCogBot\runcogbot.pl i was going to try to be sure worked with one click

[2:13:04 PM] Douglas R. Miles: that would start SWI-prolog

How, if possible