A Guide to FRDCSA and POSI

If you are interested in obtaining the latest FRDCSA Virtual Machine distribution, helping with the release, or collaborating on projects, please email me at andrewdo@frdcsa.org

FRDCSA

is a project under continual development since 2001. The primary goal is to develop a free software artificial intelligence system which can help to improve the quality of life. The method undertaken is to collect and package such software, and to write it ourselves where not already available.

The *Panoply GNU+Linux VM distribution* is an experimental release of the FRDCSA. It is not intended for public redistribution at this time (it has to be cleaned up for the public release). It is a 20 GB VirtualBox image.

http://frdcsa.org

http://frdcsa.org/frdcsa

IRC: #frdcsa@freenode.net

POSI

is a platform under development for group collaboration. It uses semantic web tools to map out information on members' goals, interest and abilities. From this and additional information about which goals require which skills and what topics the goals pertain to, we can estimate which goals/projects would be of interest to different people. POSI was initially presented at Flourish 2009, at which time we did not have a shared goaling system completed. Fortunately, in 2011, we have the SPSE2 Shared Priority System Editor v2. However, SPSE2 does not at this time allow distributed editing of goal contexts, further work is needed. It does however allow for editing personal goal systems. SPSE2 is available as part of the Panoply GNU+Linux distribution.

The latest POSI web-based interface is accessible at:

http://intranet.posi.frdcsa.org

http://posi.frdcsa.org

IRC: #posi@freenode.net

About the Author

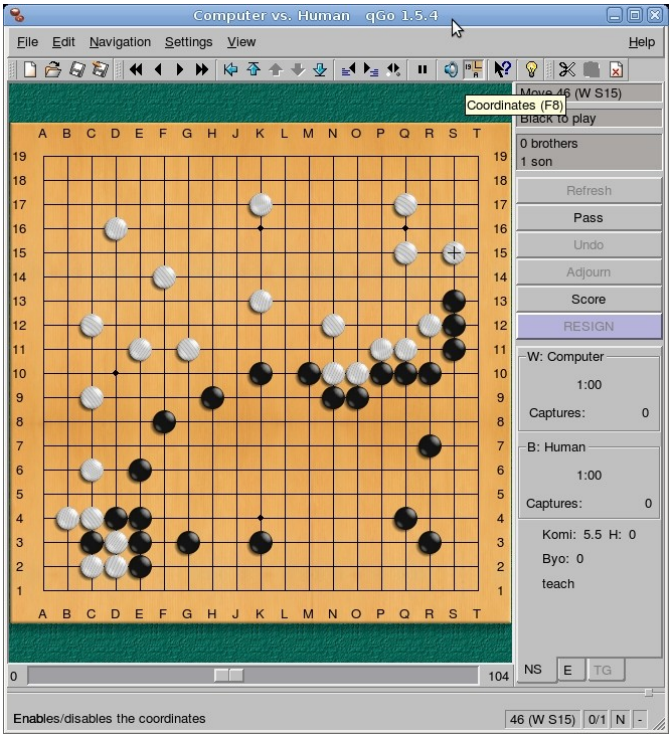
Andrew Dougherty
See http://frdcsa.org/~andrewdo

An *Expert System* that Plays the Game *...OF YOUR LIFE*

Like Deep Blue *or* GNUGo, *but analyzes the facts of your situation and helps prioritize and recommends good goals*

Pre-Alpha version available on FRDCSA *DVD or at:*
`http://frdcsa.org/~andrewdo/flourish-2011-frdcsa.iso`

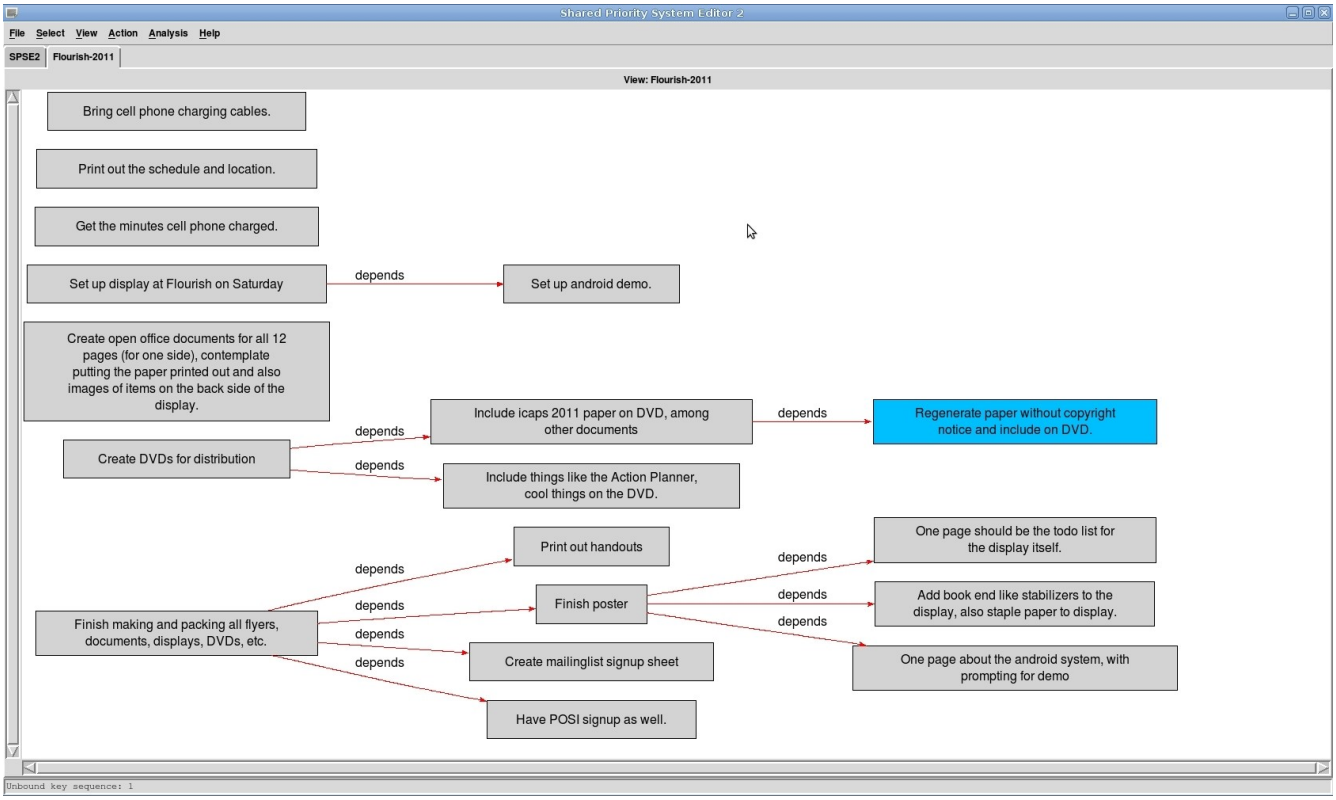
Sample GNUGo Situation



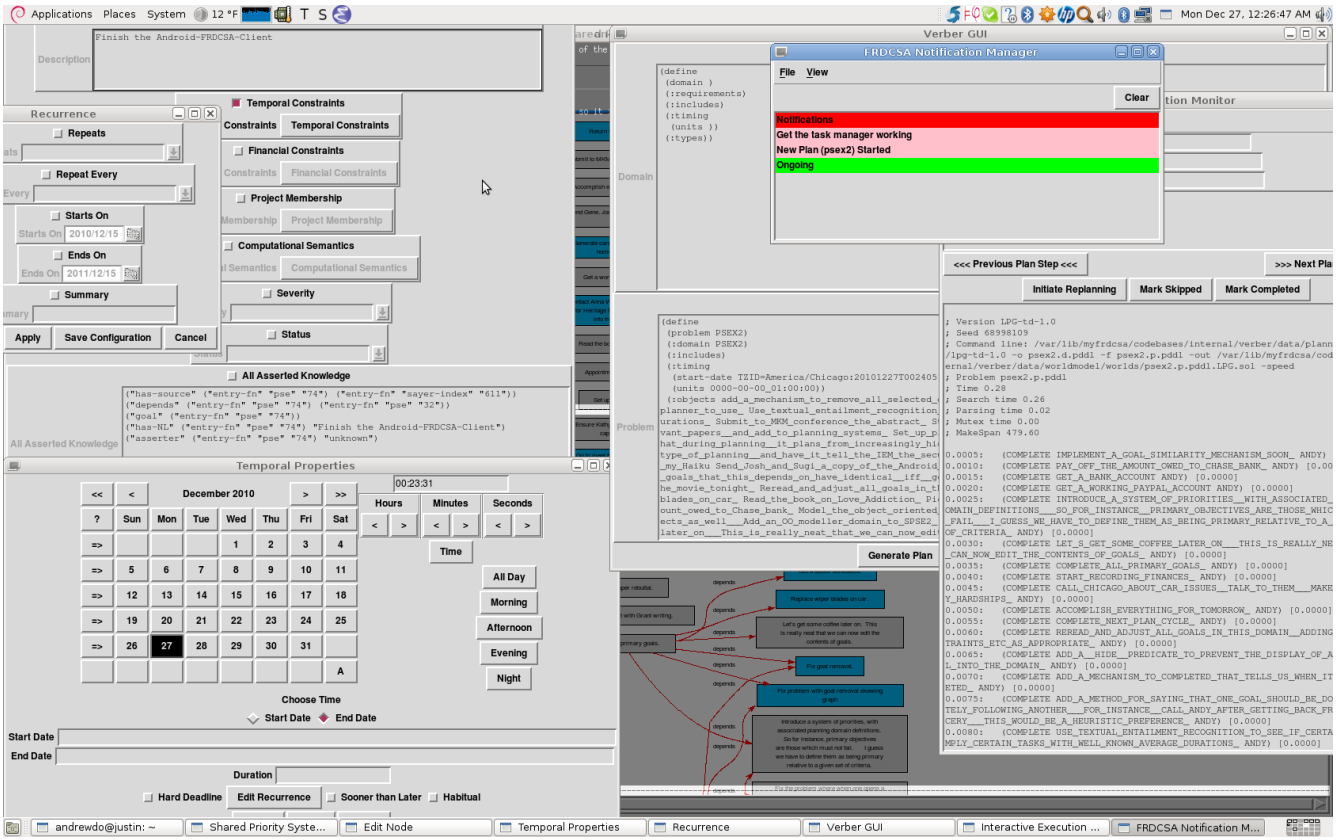
Sample Life Situation

```
(deffacts init
 (has-job John-Doe Teacher)
 ;; (not (disqualified-from-receiving-foodstamps John-Doe))
 (address
  (name John-Doe)
  (firstname John)
  (lastname Doe)
  (address1 13-Berry-Street)
  (city Sheridan-Lake)
  (state Colorado)
  (zipcode 81071)
  (country USA)
  )
 (has-medical-condition John-Doe Diabetes)
 (has-health-insurance John-Doe Blue-Cross)
 (has-bank-account John-Doe PNC-Bank)
 (father-of John-Doe Steven-Doe)
 (mother-of John-Doe Margarette-Doe)
 (brother-of John-Doe Mark-Doe)
 (relationship-strength John-Doe Steven-Doe okay)
 (relationship-strength John-Doe Mark-Doe good)
 (has-friend John-Doe Jeremy-Pierce)
 (relationship-strength John-Doe Jeremy-Pierce fair)
 (has-friend John-Doe Dina-M-James)
 (relationship-strength John-Doe Dina-M-James good)
 (has-automobile John-Doe Chevy-Camero)
 (has-laptop John-Doe dell-laptop)
 (has-computer John-Doe office-computer)
 (debt (debtor John-Doe) (amount 3000) (creditor Fannie-Mae) (type Student-
Loan))
 (debt (debtor John-Doe) (amount 500) (creditor Flaherty-Foot-Clinic) (type
Medical))
 (has-goal John-Doe "(is-self-reliant John-Doe)")
 ;; (not (is-emotionally-secure John-Doe))
 (has-fitness-plan John-Doe John-Doe-s-fitness-plan)
 ;; (not (has-financial-plan John-Doe ?plan))
 ;; (not (has-budget John-Doe ?budget)))
```

)



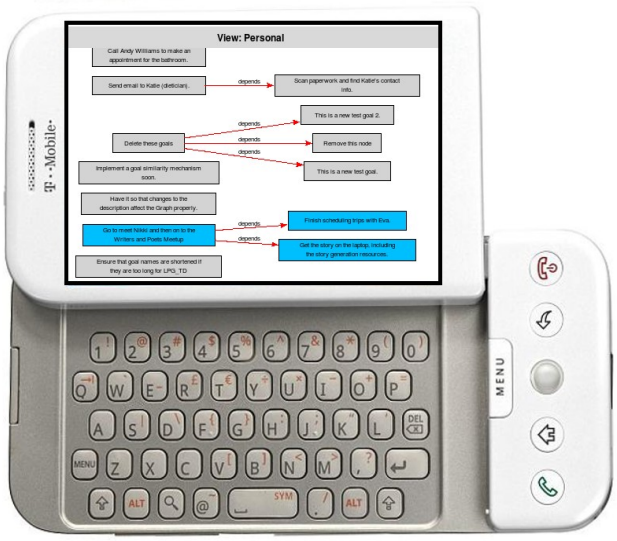
Task Planning and Monitoring



Ask to See a Demo of

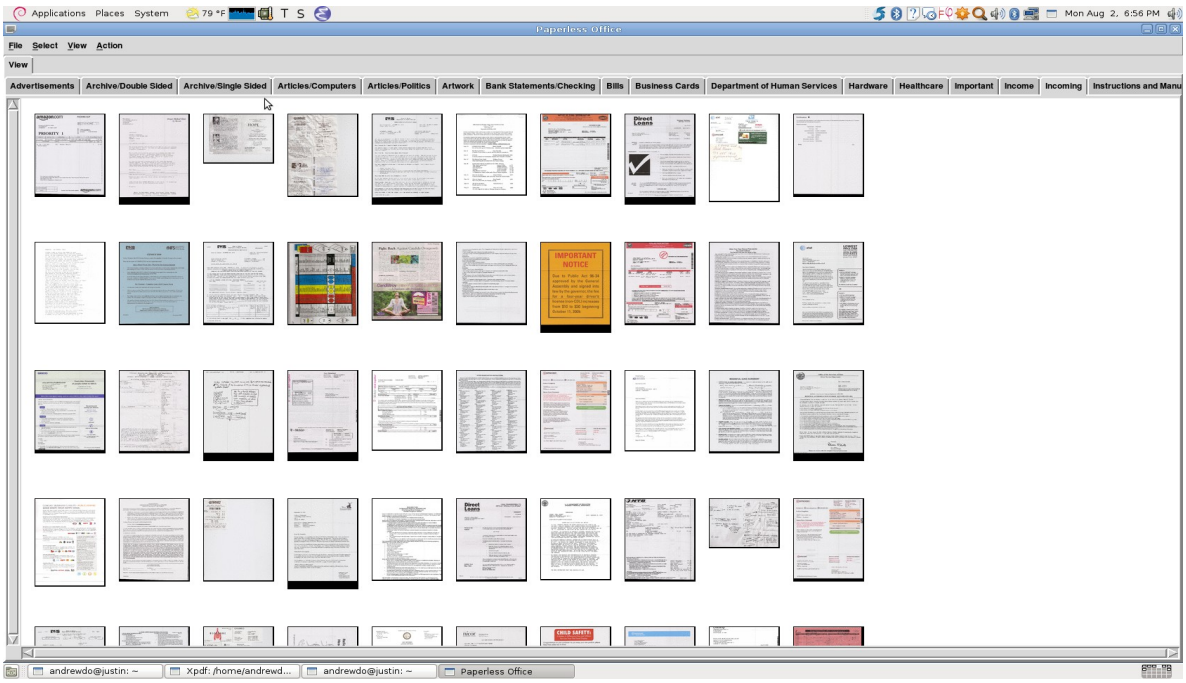
Android Hands-Free Voice Control

Seamlessly interact with your personal free software A.I. on your home computer through a cell-phone based speech interface, to record goals, ask questions and generally interact with software agents.



HTC Dream / T-Mobile G1 Skin by Olaf Encke http://www.android.encke.net

Document Management and Electronic Filing System



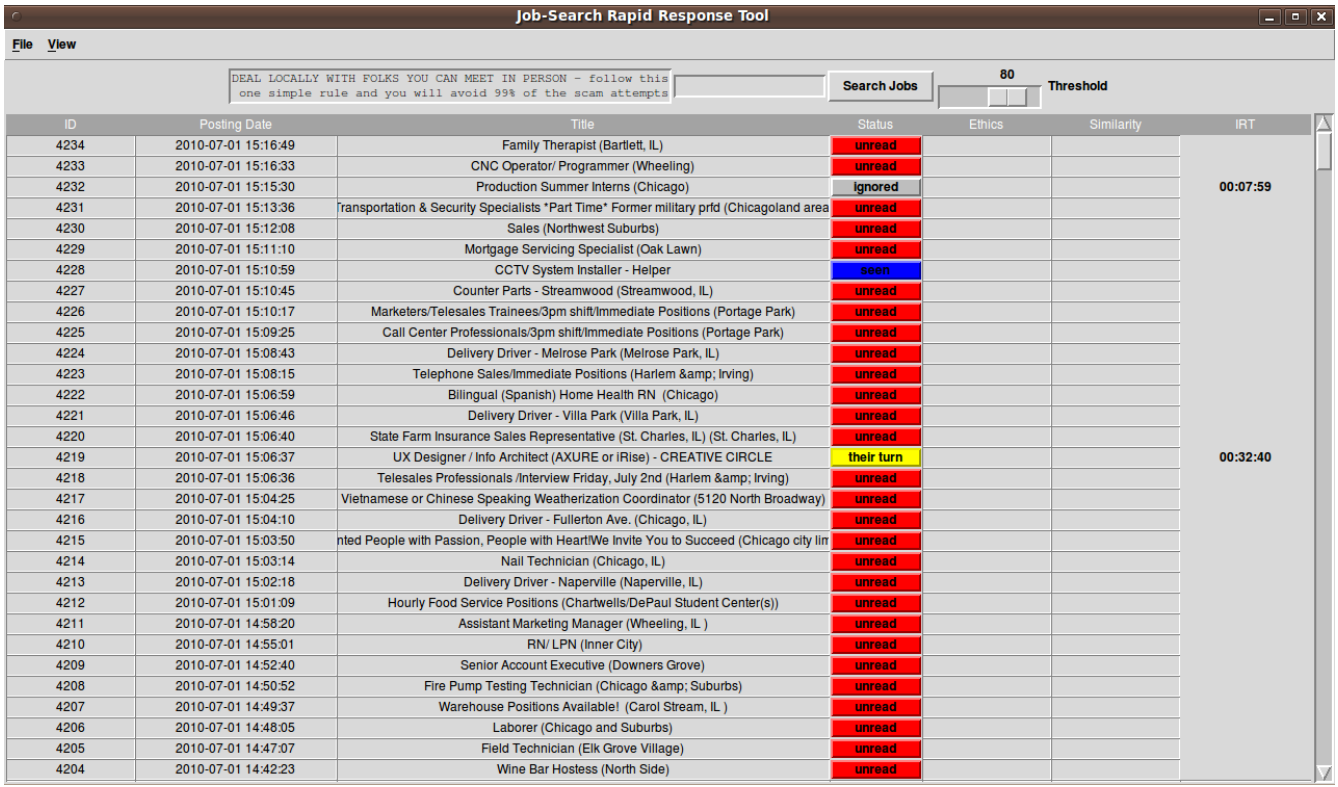
POSI

Record your Goals, Interests and Abilities using SPSE2 and form dynamic teams to accomplish shared goals. (See slides)

http://intranet.posi.frdcsa.org

Job Search Tool

For resume building, relevance ranking, rapidly responding to job postings, tracking the results, and invoicing.



ID	Posting Date	Title	Status	Ethics	Similarity	IRT
4234	2010-07-01 15:16:49	Family Therapist (Bartlett, IL)	unread			
4233	2010-07-01 15:16:33	CNC Operator/ Programmer (Wheeling)	unread			
4232	2010-07-01 15:15:30	Production Summer Interns (Chicago)	ignored			00:07:59
4231	2010-07-01 15:13:36	Transportation & Security Specialists *Part Time* Former military prfd (Chicagoland area	unread			
4230	2010-07-01 15:12:08	Sales (Northwest Suburbs)	unread			
4229	2010-07-01 15:11:10	Mortgage Servicing Specialist (Oak Lawn)	unread			
4228	2010-07-01 15:10:59	CCTV System Installer - Helper	seen			
4227	2010-07-01 15:10:45	Counter Parts - Streamwood (Streamwood, IL)	unread			
4226	2010-07-01 15:10:17	Marketers/Telesales Trainees/3pm shift/Immediate Positions (Portage Park)	unread			
4225	2010-07-01 15:09:25	Call Center Professionals/3pm shift/Immediate Positions (Portage Park)	unread			
4224	2010-07-01 15:08:43	Delivery Driver - Melrose Park (Melrose Park, IL)	unread			
4223	2010-07-01 15:08:15	Telephone Sales/Immediate Positions (Harlem & Irving)	unread			
4222	2010-07-01 15:06:59	Bilingual (Spanish) Home Health RN (Chicago)	unread			
4221	2010-07-01 15:06:46	Delivery Driver - Villa Park (Villa Park, IL)	unread			
4220	2010-07-01 15:06:40	State Farm Insurance Sales Representative (St. Charles, IL) (St. Charles, IL)	unread			
4219	2010-07-01 15:06:37	UX Designer / Info Architect (AXURE or iRise) - CREATIVE CIRCLE	their turn			00:32:40
4218	2010-07-01 15:06:36	Telesales Professionals /Interview Friday, July 2nd (Harlem & Irving)	unread			
4217	2010-07-01 15:04:25	Vietnamese or Chinese Speaking Weatherization Coordinator (5120 North Broadway)	unread			
4216	2010-07-01 15:04:10	Delivery Driver - Fullerton Ave. (Chicago, IL)	unread			
4215	2010-07-01 15:03:50	nted People with Passion, People with Heart!We Invite You to Succeed (Chicago city lin	unread			
4214	2010-07-01 15:03:14	Nail Technician (Chicago, IL)	unread			
4213	2010-07-01 15:02:18	Delivery Driver - Naperville (Naperville, IL)	unread			
4212	2010-07-01 15:01:09	Hourly Food Service Positions (Chartwells/DePaul Student Center(s))	unread			
4211	2010-07-01 14:58:20	Assistant Marketing Manager (Wheeling, IL)	unread			
4210	2010-07-01 14:55:01	RN/ LPN (Inner City)	unread			
4209	2010-07-01 14:52:40	Senior Account Executive (Downers Grove)	unread			
4208	2010-07-01 14:50:52	Fire Pump Testing Technician (Chicago & Suburbs)	unread			
4207	2010-07-01 14:49:37	Warehouse Positions Available! (Carol Stream, IL)	unread			
4206	2010-07-01 14:48:05	Laborer (Chicago and Suburbs)	unread			
4205	2010-07-01 14:47:07	Field Technician (Elk Grove Village)	unread			
4204	2010-07-01 14:42:23	Wine Bar Hostess (North Side)	unread			

Semi-Automatically Create Packages Quickly For Debian and Ubuntu Linux

FRD

Formalized Research Database

is intended as a collection of all mathematical knowledge. It implements a transfinite solution to Hilbert's Program. The goal is to build a very capable theorem proving system, capable of resolving as much of mathematical truth as is possible and necessary efficiently.

CSA

Cluster Study and Apply

is a software toolchain for locating, packaging and using existing free and open source software. CSA has an emphasis on practice (not mathematical, but real-world problems, such as software conglomeration, document management, etc).

EXISTING CAPABILITIES:

(Non-exhaustive) List of the Capabilities of the FRDCSA via internal and external packages

KEY:
(#system) -> not fully implemented
(%system) -> partially performs capability
(!system) -> integrated but not available due to licensing
() -> no system implements this capability yet

Misc

Multi-agent system (UniLang)
Self-directed goal setting (Architect)

Natural Language Processing

Question Answering (QUAC/OpenEphyra/Aranea/ %TextMine/AnswerBus)
Machine Translation (Apertium|Google)
Query Expansion (lucqe)
Recognizing Textual Entailment (#FreeRTE/CAndC/#Stanford-RTE)
Natural Language to Logic (CELT/#Formalize/! ResearchCyc/%APE)
Controlled Language (APE|ACEWiki)
Shallow Parsing (Assert|#Shalamaneser)
Sentiment Analysis (OpinionFinder)
Anaphora Resolution (JavaRAP|#BART)
Sentence Similarity (%Meteor|%Stanford-RTE)
Text Clustering (AI::Categorizer/Rival::AI::Categorizer/ PerlLib::Cluster/crossbow/OSKM/)
TFIDF (PerlLib)
Topic Gisting (ConceptNet)
Affective Analysis (ConceptNet)
Word Sense Disambiguation (SenseRelate)
Named Entity Recognition (Cage/CAndC/NamedEntity/OpenNLP/Simple/ Stanford/TagEn/TextMine)
Tokenization (treebank/TextMine)
Grammar Checking (LinkGrammar)
Summarization (ConceptNet/MEAD)
Parsing (Charniak/Lingua::EN::Tagger/LinkGrammar)
Predicate Parsing (%Assert/Enju)
Abbreviation Extraction
Body Text Extraction (BTE)
Concept Extraction (#TMT)
Date Extraction (TempEx)
Definitional Extraction ()
Fact Extraction (#fact-extraction/%Assert)
Information Extraction (T-Rex/MinorThird/Elie/#MDR/#AIE)
Logic Form Extraction (FreeLogicForm/CAndC)
Temporal Ordering Extraction (TARSQI)
Terminology Extraction (TermEx)
Semantic Annotation (OpenCalais)
Topic Detection and Tracking ()
Definitional Question Answering ()
Bad Word Lists ()
Word Connotation (EAT)
Rhyming (rhyme)

HTML Processing

Event Extraction (%GeekEventAggregator)
URI Extraction (WWW::Mechanize/PerlLib::URIExtractor)

AI/Knowledge Representation

Knowledge Based System (#FreeKBS/OpenCYC/# %Sayer)

Studying

Digital Library (DigiLib/Greenstone/#DSpace)
Intelligent Tutoring System (CLEAR/#SystemX)

Humanitarian/Life Support

Task Management (#Manager Todo/#PSE)

Gourmet Meal Planner

Personal Inventory Management

and many more:

akahige
all
alleyoop
antispam-console
antispam-node
apartment-finder
architect
assess
audience
bard
boss
brainleach
broker
busroute
canvasser
classify
clear
coauthor
code-monkey
corpus
crawler
critic
cso
diamond
digilib
disciple
do
ems
ender
evangelist
event-log
event-system
fieldgoal
formalforum
formalize
frdcsa-el
frdcsal
frdcsa-reference
freekbs
freekbs2
fweb
fweb2
generix
gourmet
gourmetjr
job-search
kbfs
kmax
learner
manager
mantra
meeting
mmi
mush
myfrdcsa
normal-form
ocra
packager
paralegal
parking-system
perform
perllib
pgourmet
picform
posi
problemspace
pse
pverber
quac
radar
reasonbase
rwhois
samwise
sanctus
savant
score
seeker
setanta
setanta-client
shops
silo
sinless
socbot
source-hatchery
study
svrs
symbiosis
systemx
task1
transport
unilang
vanguard
verber
wow
3d-model-library
3d-scene-reconstruction
academician
acewiki-extensions
action-planner
advanced-ttyplay
agent
aie
aigo
analysis-engine
android
android-frdcsa-client
archives-manager
argument-system
audience-message-handler
auto-builder
auto-packager
better-locate
bill-tracking-system
biometrics
bluetooth-tools
bookmark-clustering
bookmark-clustering-old
bootstrapping-minority-language-resources
brainstorm
broker-buy-sell-system
budget-system
car-maintenance
cause-effect
ccpp
chap
coauthor-interactive-natural-language-generator
code-language
code-search
component-systems
computational-ethicist
consultant-support
cooperative-game-archive
corpus-manager
critic-browser
crm
csa
cultural-systems
dandd
data-manager
debt-manager
definition-extraction
dialog-interface
discourse-representation
distributed-scraper
documentary-generator
documentation-central
drivers-education-and-safety-training
e-health-clinic
elderly-affairs-management
emacs-nlp
emacs-poetry-mode
email2speech
email-prioritization-and-urgency-routing
encoder
entertainment-center
entertainment-ideas
entity-database
ethical-consumer-system
ethical-systems-analysis
event-extractor
fact-extraction
faster-apt-file
federated-agents
file-organizer
files
folksonomy
food-ontology
frd
frdcsa-at-home
frdcsa-browser-extensions
frdcsa-bug-tracking
frdcsa-dashboard
frdcsa-installer
free-knext
free-logic-form
free-rte
free-wordseye
friend-finder
full-planner
fweb-translate
game-time
genealogy-system
get-help
golden-retriever
google-summer-of-code-application
gourmet2
grant-manager
grocery-bill-splitter
handwriting-recognition
historian
home-automation
house-rules
icodebase-testing
id-card-creator
idiom-database
infolab
interactive-execution-monitor
internet-research-tool
interrelator
ipvord
irc-tdt
irish-nlu
its
jess-debugger
js-application-agent
js-form-filler
js-rapid-response
language-learning
lessons-learned
license-manager
link-checker
linux-media-acquisition
list-of-lists
location-logic
log-analysis
logic-form-to-nl-generation
market-analyzer
marketing-manager
market-researcher
mass-service-registration
mdr
media-library
mental-benchmarks
menu-annotator
metasite-extractor
metaverse
miscellaneous-games
model-based-problem-solver
monitoring-agent
movie-recommendation
musical-system
music-composition-system
music-control-system
news-monitor
nl-to-fol
nl-to-pddl
nlu
nooks-and-crevices
notification-manager
object-recognition
ofcs
openoffice-gramadoir-integration
option-classifier
ossmole-assistant
package-installation-manager
paperless-office
password-manager
perl6-stdlib
personal-adaptive-filtering
poverty-survival-system
pposi
problem-database
problem-monitor
profession-agent-factory
pse2
pse-x
raiders-of-the-ftp-sites
reading-analyzer
reading-comprehension
recovery-frdcsa
requirements-optimization
research-ontology
resource-reservation-system
restaurant-music-system
resume-critic
robotic-book-scanner
rsr
rte
ryans-video-game
save-tara
sayer
sayer-learner
schizophrenai
setanta-agent
social-intelligence
social-recreational-game-playing-and-development
software-evaluator
software-finder
software-indexer
software-ontology
special-lug-admin
spreading-activation-package-search
spse
story-generation
story-teller
study-memorization-subsystem
suppose-suppose
suppositional-decomposer
suppositional-reasoner
system
system-ie
system-implementor
system-recommender
task-manager
task-tagging-system
tdt
temp-agency-locator
temporary-system
termios
terms-and-conditions-analyzer
text-analyzer
the-frdcsa-book
thinker
to.do
todo-systems
travel-system
twitter-follower-analyzer
unilang-message-classifier
vanity-search
various-perl-libraries
virtual-hacklab
vm-manager
vm-release
web-analytics
web-backup
windows-power-operating-system
workflow-planning
workhorse
xwordnet-utils

Grant Proposal:

FRDCSA Initial Release

Sign Petition On Table

Amount Requested:

$2250 +- $750

Synopsis

The FRDCSA consists of, among other things, over 1700 Perl5 modules in various stages of completion that operate in a tightly interwoven fashion. There are many entire systems for novel and important applications, especially from various subjects in Artificial Intelligence, as well as wrappers and APIs for other useful systems. Unfortunately, owing to their interdependence, lack of tests and documentation, need for a systematic renaming and deidentification - distributing them through CPAN has not yet been accomplished, although most of the software required to compute their distribution has been written. I seek a grant to facilitate their release either in majority or totality.

Benefits to the Perl Community

This grant IMHO would greatly improve the situation for Artificial Intelligence through Perl. But the benefits are not limited to pure theory, they are very practical - as these techniques, and the systems for which wrapping is provided, would then be instantly accessible to the entire Perl community. I do not expect the project to be understood from the getgo, however, as Perl programmers come across it, by searching CPAN for various tools, it should draw a small amount of attention - enough to bring the true benefits of the cohesive and interwoven system to the attention of Perl power users. As I am not a Perl expert, I am not familiar with all of the greatest aspects of Perl - however, I do know that there are many systems lacking for which I have labored over 10 years to provide the matching capabilities. Getting some attention for my project would be important to secure the goals of the system. The system, such as it is, is an attempt to implement a transfinite implementation of Hilbert's program. In this sense, it contains as its goal to solve all mathematical problems, which are thought to impose themselves on the real world. Goedelian incompleteness is not an obstacle, it is the cornerstone, based on associating increasingly complete systems of logic with ordinals ala Turing 1939.

How will this advance the release of Perl 6?

Interesting question that was not in the first template I filled out. I have often wondered whether this system is necessary to complete Perl6. I cannot comment precisely on exactly what it will do wrt the existing Perl 6 spec. However, if that spec is mutable, I think there are essential components. That is to say, if we can think of Perl 6 as being a more capable or "intelligent" version of Perl, then the availability of these tools will greatly enhance and furnish that goal. For instance, there is limited existing CPAN support for Semantic Web, Theorem Proving, or certain individual techniques from Natural Language Processing. One component of the FRDCSA is FRDCSAL which is a Perl based

programming language where you write the code in English, and the code is translated to logic and executed. True, these are Perl5 modules, but when the automated refactoring system is complete, and the formal specification is complete - it will be possible to translate semi-automatically to Perl6. There is also a sub-project called Perform: which is a knowledge base about algorithms and data structures - which includes their complexity and all kinds of other features, intended as a devastatingly complete standard library for Perl6.

Deliverables

Many to all modules released.

Here is a very recent module list:

http://frdcsa.org/~andrewdo/projects/module-list.txt

For more information, please see the ancient description. Upon request I can regenerate the pages to reflect the current situation.

http://frdcsa.onshore.net/frdcsa

I have just this year started writing GUIs as frontends for various systems: Here is a page on that:

http://frdcsa.org/~andrewdo/projects/frdcsa-guis

There is also:

http://frdcsa.org
http://positho.org
http://intranet.positho.org

I am not familiar with milestone documents. The critical path is in the process of being computed, but won't be finished prior to the actual release. Here is an outdated document that sort of documents that:

http://files.meetup.com/970635/semweb.pdf

I have since added nested formulae and First Order w/Equality inferencing to FreeKBS2 - which makes the planning system much more capable - however, it has not been fully updated to use the new backend and so is not usable at the moment.

Also see number 13 at:

http://frdcsa.org/~andrewdo/projects/frdcsa-guis/

Project Details

The project is based primarily on Algorithmic Information Theory, and "Information Theoretic Limitations of Formal Systems". The basic idea is that, given a particular Turing machine, in order to

accomplish tasks of increasing complexity, the software must ultimately grow in program length. Therefore, a necessary but insufficient condition for an "Artificial Intelligence" that has many capabilities is that it is large. This means that one heuristic for finding existing such systems is that the system is large. This immediately makes one think of Perl, Debian and Emacs - which all check out as extremely capable systems. The system is based upon these, but will not be exclusive to them. However, as comprehensive as CPAN is, there are still elements missing.
Hence, we contemplate how to grow the system - there are two approaches - write software by ourselves - or find existing software. Both are taken. To find existing software, both existing large metasites (Sourceforge/Freshmeat/Google Code/etc) and active focused crawling for the location of new software. The Flossmole project releases data. To classify the data, the Debian tags and Sourceforge categories have been trained via an SVM text classifier to classify new entries according to these folksonomies. All this information goes into the CSO (Comprehensive Software Ontology). Focused crawling is achieved with the radar-web-search system, a multithreaded spider that searches several ply deep on websites for files that would seem to be useful. These are downloaded and go through an entire system which I cannot help but to compare to a digestive system. They are (semi-) automatically packaged for Debian using the Packager system.

For writing software, there are an equal number of sundry systems that expedite that process. For instance, all kinds of project templates and functions exist in the BOSS system. A particularly useful trick is ppi-convert-script-to-module.pl, which takes a script and converts it bit by bit to the standard Object Oriented Perl Module format. It is an example of a hand-coded automated refactoring that is going to be rearchitected into a deliberative automated refactoring system.
What is the goal of the project? The project aims to characterize all arguments for and against various beliefs, audit them for relative considerations, and present people with, if possible, a uniquely action guiding moral support intelligent agent. The basic goal of the project is to resolve problems that affect sentient beings in an equinaminous and peaceful manner. Many resource conflicts can be avoided through more rational behavior.

Inch-stones

There are several primary tasks for the project wrt CPAN. They are: Release, Refactor, Test and Document. The first three have corresponding automatic implementations: Release -> Task1, Refactor -> CodeMonkey, Test -> icodebase-testing. Documentation will be largely by hand and/or copied from my existing documents. Task1 is mostly complete. Refactor is 10-15% complete. Test is not really started. To fit within the time frame of the 3 month release schedule, I will focus only on Release (unless it happens faster than expected).
Release is divided into: Dependencies, Renaming, Deidentification, Testing. Dependencies are already computed but will be tidied up (have to reacquaint myself with the Task1 codebase), and possibly will now use FreeKBS2 logic programming to simplify the tasks. Testing will be only as necessary. Only about 2% of tests have been written. Renaming requires testing (to prevent large scale breakages), but is otherwise straightforward. Deidentification is tedious, however the Classify deidentification system is about 50-60% complete.

This should be enough to get the project into use and should generate future interactions which will ensure its completion.

Project Schedule

Project will take <= 3 months.
Can begin immediately upon receipt of initial funding segment.

Completeness Criteria

The project will be complete if most of the important systems of the FRDCSA are in CPAN and are not in a catastrophic condition. Given that there are 1700 modules, it is not reasonable (unless the automated refactoring system is finished, populated with adequate rules, and extremely accurate) to expect me to bring the code up to the quality standards that I myself do not know. Therefore, having them available in CPAN should be enough to attract attention to them, at which point in time what I need to do will be told to me rather than queried blindly and annoyingly of Perl experts. Many of them already function perfectly adequately or even superbly.

Natural Language Understanding and Computational Semantics

The presentation on Natural Language Understanding and Computational Semantics will outline recent work on several areas of interest to natural language processing - especially Recognizing Textual Entailment and Natural Language Understanding. The method employed is conversion of natural language (through various approaches such as compositional grammars based on Lambda Calculus, statistical dependency parsers, etc) to a logic based interlingua. In our case we focus on LCC-style logic forms in first order logic and their usage in Discourse Representation Theory. We then show how to reason with the interlingua using automated theorem proving tools and model building tools.

The second part of the talk will focus on independent research into Natural Language Understanding based on breaking the text down and using classifiers, knowledge sources, various NLP software systems, etc to assert additional information (i.e. text markup). The idea is to build chains of proofs to create a system of constraints delineating the possible meanings in context of any textual/document source. Metadata extends to arbitrary knowledge regarding the document and source. The idea is to build a set of tools for literary interpretation that allows us to reason more precisely with the meaning of utterances and to use this information to support persons. A recent project has been undertaken to create annotated documents for all freely available knowledge sources, such as Wikipedia, Gutenberg and Google Books public domain content.

If approved, the concept of the talk will be refined to include all the practical prerequisites required to understand the talk. I would also like to illustrate some of the practical issues involved that others can become involved with immediately, especially the packaging and wrapping of existing NLP software.

Three Things the Audience Will Learn: How to formalize language, how to develop logically consistent metadata and insights, how to contribute to such software.

The Automated Lifestyle: Using and Developing Panoply GNU+Linux

This workshop is focused on getting the users familiar with usage and development of Panoply GNU+Linux. Panoply Linux, currently only available as a 20 GB VM image, contains the full FRDCSA system (but without the data files, that would be over 100 GB), covering all of my research into artificial intelligence. There are many systems that are only available with Panoply (until such time as proper Ubuntu packages can be built and CPAN modules created): such as Paperless-Office, Shared Priority System Editor v2, Job-Search, RADAR, and CLEAR, and tools such as FreeKBS2 and UniLang. Many more are not finished but already present in substantial form. I wish to teach the various specialized systems of the FRDCSA, so that users will have a deep understanding of how to use these systems for their application development. The primary systems that they will learn are UniLang,

a multiagent interprocess communication system, and FreeKBS2 - a Semantic Web knowledge-based system for storing and reasoning with assertions in many logics and notations.

In additional to the internal systems, I will also cover a wide variety of external software, mostly in the area of natural language processing and understanding.

We'll then proceed to illustrate basic packaging techniques and also packaging using the Packager system, which helps to semi-automatically retrieve and package software for Debian and Ubuntu. We will actually roll a package in the workshop.

We'll then show how to agentify the packaged software for use with UniLang, or separately under the Org::FRDCSA::System or Org::FRDCSA::Capability namespaces.

The actual workshop will be represented as a pedagogical domain with the SPSE2 system - it will be represented as a series of individual learning activities with specified content dependencies and expected durations. The SPSE2 will then generate a precise schedule and walk us through the completion of the workshop. For more information on this process you can refer to an upcoming and as of yet unfinished paper: "Temporal Planning and Inferencing for Personal Task Management with SPSE2". http://frdcsa.org/~andrewdo/writings/icaps-2011-paper.pdf

Lastly, the concepts behind the POSI group will be presented. Users may wish to index and share their capabilities and doing so begin collaboration using the SPSE2 system. I gave a talk on POSI at Flourish 2009 and this will be a follow-up to that talk, now that the POSI systems are mostly working: http://www.slideshare.net/aindilis/posi-overview

Three Things the Audience Will Learn:

How to package software for Debian&Ubuntu; how to use and develop Panoply GNU+Linux; how to collaborate with others using SPSE2.