

FRDCSA Temporal Planning and Inferencing for Task Management through the Verber, PSE and FreeKBS Systems

Andrew Dougherty
FRDCSA, POSI, POSIC

- Develop a fully functional free software artificial intelligence system
- Empower individuals by assisting them to solve problems
- Assist individuals to act responsibly

- Algorithmic Information Theory
- Information-Theoretic Computational Complexity of Metamathematics
- For a given (classical) computer, for any theorem prover P , there exists another prover Q , capable of solving a strictly larger set of problems
- Eventually, in order to solve more problems, Q must increase in program length
- Size is therefore a necessary, however insufficient, condition

- Build a sequence of increasingly sophisticated theorem provers
- Must package all quality free software and assemble them under an operating system
- FRDCSA Virtual Machine (~10 GB or w/datasets ~100 GB)

- Numerous Semantic Web problems involved
- Build ontology of software capabilities - Comprehensive Software Ontology (CSO)
- Build ontology of goals
- 9000 statements added constraining core project goals
- Desire proofs in deontic logics of morality of actions
- Importance of ontology and epistemology for priority and value systems

The World Model

- UniLang Multi-Agent System allows users to submit entries representing various facts, statements, opinions, goals, etc. for consideration
 - => "Have mobile wireless on the train",
 - => "Have budget analysis system working",
 - => "Have dependency reasoner working",
 - => "Order mobile broadband",
 - => "Do job for Eric",
 - => "Read books on the laptop",
 - => "Build a goal/knowledge structure here and then grow it until it includes many of the existing unilang entries",
 - => "Develop a system that reasons about what will fail if certain things aren't done.",
 - => "Setup core business processes",
 - => "Make adequate money each month",

Automatic Routing of UniLang Entries

- Basic classification system for UniLang entries
- Assigned using text classification

((?X "donation-request")) (?X "anecdote")) (?X "capability-request")) (?X "complex-statement")) (?X "dangling-clause")) (?X "deleted")) (?X "dream"))
((?X "event")) **((?X "goal"))** ((?X "ibase-capability-request")) ((?X "ibase-input-data")) ((?X "ibase-resource")) ((?X "ibase-solution-to-extant-problem")) ((?X "ibase-task")) ((?X "inspiring-anecdote")) ((?X "intersystem-relation")) ((?X "not-a-unilang-client-entry")) ((?X "observation")) ((?X "policy")) ((?X "poem")) ((?X "political-action-item")) ((?X "priority-shift")) ((?X "propaganda")) ((?X "quote")) ((?X "rant")) ((?X "shopping-list-item")) ((?X "solution-to-extant-problem")) ((?X "SOP")) ((?X "suspicion")) ((?X "system-request")) ((?X "unclassifiable")) ((?X "unilang-client-outgoing-message")) ((?X "verber-task-definition")) ((?X "personal"))

Additional Processing of UniLang Entries

- Sayer attempts to “assert” knowledge (in terms of graphs of Perl function calls and dumped data structures) about these entries
- We can derive theorems from Sayer

```
Computing result and adding to cache
$VAR1 = {
  'SemanticAnnotation' => [
    {
      'CalaisSimpleOutputFormat' => {
        'Topics' => {
          'Topic' => {
            'Taxonomy' => 'Calais',
            'Score' => '0.636',
            'content' => 'Other'
          }
        }
      },
      'Description' => {
        'allowSearch' => 'true',
        'externalID' => 'testing',
        'about' => 'http://d.opencalais.com/dochash-1/5489cd7d-f4fa-3bbc-9c18-ee184f0517c2',
        'id' => 'http://id.opencalais.com/Nk14vYACRFBZCg66pmggHQ',
        'allowDistribution' => 'true',
        'calaisRequestID' => 'd0dc9ff2-fcb7-4b05-88db-4eefc74531b9'
      }
    }
  ],
  'TermExtraction' => [
    []
  ],
  'NounPhraseExtraction' => [
    'unilang entries',
    1,
    'entries',
    1,
    'unilang',
    1,
    'goal/knowledge',
    1,
    'structure',
    1,
    'existing unilang entries',
    1,
    'goal/knowledge structure',
    1
  ],
  'Tokenization' => [
    'Build a goal/knowledge structure here and then grow it until it includes many of the existing unilang entries'
  ],
  'DateExtraction' => [
    '<doc>'
  ],
  '<s><lex pos=vb>Build</lex> <lex pos=det>a</lex> <lex pos=nn>goal/knowledge</lex> <lex pos=nn>structure</lex> <lex pos=rb>here</lex> <lex pos=cc>and</lex> <lex pos=rb>then</lex> <lex pos=vb>grow</lex> <lex pos=prp>it</lex> <lex pos=in>until</lex> <lex pos=prp>it</lex> <lex pos=v>bz>includes</lex> <lex pos=jj>many</lex> <lex pos=in>of</lex> <lex pos=det>the</lex> <lex pos=jj>existing</lex> <lex pos=nn>unilang</lex> <lex pos=nns>entries</lex></s>'
],
  '</doc>'
];
```

PSE Emacs Goal Manipulation Interface

- Possible to derive interrelations between goals using Natural Language Processing
- Interrelations also can be manually added using the FreeKBS and PSE Emacs Interfaces

```
(global-set-key "\C-cpat" 'pse-assigned-to)
(global-set-key "\C-cpab" 'pse-assigned-by)
(global-set-key "\C-cpap" 'pse-add-person)
(global-set-key "\C-cpp" 'pse-assert-property-about-goal)
(global-set-key "\C-cpc" 'pse-completed)
(global-set-key "\C-cpi" 'pse-incomplete)
(global-set-key "\C-cpC" 'pse-query-completed)
(global-set-key "\C-cpo" 'pse-comment)
```

PSE Emacs Goal Manipulation Interface

```
;global-set-key "\C-cpd" "prefix-key for pse-due-date.el functions"  
;global-set-key "\C-cpr" "prefix-key for pse-due-date.el remit functions"  
;global-set-key "\C-cpq" "prefix-key for pse-quick goal functions"  
(global-set-key "\C-cpqg" 'pse-quick-goal)  
(global-set-key "\C-cpqc" 'pse-quick-goal-completed)  
(global-set-key "\C-cpqp" 'pse-quick-precondition-for-eap)  
(global-set-key "\C-cpqd" 'pse-quick-depends-on-eap)  
(global-set-key "\C-cps" 'pse-solution)  
(global-set-key "\C-cpb" 'pse-belongs-to-system)  
(global-set-key "\C-cpe" 'pse-display-entry-for-id-at-point)  
(global-set-key "\C-cpf" 'pse-find-similar-goals-to-goal-at-point)  
(global-set-key "\C-cptr" 'pse-typical-rejected)  
(global-set-key "\C-cpts" 'pse-typical-skipped)  
(global-set-key "\C-cptR" 'pse-typical-ridiculous)  
(global-set-key "\C-cpto" 'pse-typical-obsolete)  
(global-set-key "\C-cpxa" 'pse-extension-do-action-for-goal-at-point)
```

- Interrelations asserted using PSE (Planning, Scheduling and Execution) from Emacs are stored into FreeKBS, the most SemWeb like of our systems
- FreeKBS uses MySQL for persistence, and does not implement nesting of terms or inferencing
- FreeKBS data model and API however are useful and we will integrate an OWL Reasoner as the back-end, provided it is efficient

Part of a Sample (Yet Valid) PSE Context

```
andrewdo@justin:/var/lib/myfrdcса/codebases  
/internal$ corpus --senders PSE-X -s . -d 100  
-k pse-x
```

Starting ModManager...

'Get a new laptop'

```
("eases" "107405" "107420")  
("depends" "107407" "107405")  
("eases" "107405" "107408")  
("depends" "107405" "107409")  
("eases" "107405" "107410")  
("costs" "107405" "$400")  
("goal" "107405")  
("prefer same" "107405" "107408")
```

'install FRDCSA on my new laptop'

```
("depends" "107407" "107405")
```

'Present at Flourish'

```
("eases" "107405" "107408")  
("goal" "107408")  
("prefer same" "107405" "107408")  
("ethicality-concern" "107408" "evangelism")
```

'Make a list of the features we want to have in a laptop'

```
("depends" "107405" "107409")
```

'Have mobile wireless access through phone'

```
("eases" "107405" "107410")  
("eases" "107411" "107410")  
("costs" "107410" "$60 / mo")  
("provides" "107410" "107415")
```

'Get an android based phone'

```
("eases" "107411" "107410")  
("costs" "107411" "$200")  
("depends" "107412" "107411")  
("eases" "107414" "107411")
```

'Have FRDCSA Interactive Execution Monitor working'

```
("depends" "107412" "107411")  
("depends" "107412" "107413")  
("goal" "107412")  
("prefer same" "107412" "107416")
```

Symbolic Knowledge Manipulation Environment

- Dozens of key-combination bound functions for rapidly constructing and {assert,query,unassert}ing FreeKBS formula using text in Emacs buffers, regions, thing-at-points, entry-at-points
- Can quickly snarf various syntactic entities (symbols, words, sentences, URIs, UniLang entries, named-entities*) under the cursor and include them in your formula
- Tab completion of predicates from FreeKBS
- Sundry niceties

Some FreeKBS Functions

"\C-cs>" 'freekbs-get-id-of-assertion-at-point	"\C-csP" 'freekbs-unshift-predicate-onto-stack
"\C-csx" 'freekbs-select-context	"\C-cs!" 'freekbs-craft-not-formula
"\C-csc" 'freekbs-clear-stack	"\C-csE" 'freekbs-craft-exists-formula
"\C-csv" 'freekbs-view-stack	"\C-csV" 'freekbs-craft-forall-formula
"\C-cse" 'freekbs-edit-stack	"\C-csA" 'freekbs-assert-relation
"\C-csE" 'freekbs-read-from-minibuffer	"\C-csa" 'freekbs-assert-relation-read-predicate
"\C-csp" 'freekbs-pop-stack	"\C-csU" 'freekbs-unassert-relation
"\C-cs." 'freekbs-push-entry-at-point-onto-stack	"\C-csu" 'freekbs-unassert-relation-read-predicate
"\C-cs," 'freekbs-push-entry-in-region-onto-stack	"\C-csQ" 'freekbs-query-relation
"\C-csl" 'freekbs-load-assertion-into-stack	"\C-csq" 'freekbs-query-relation-read-predicate
"\C-css" 'freekbs-push-symbol-onto-stack	"\C-csk" 'freekbs-knowledge-editor
"\C-csS" 'freekbs-push-search-unilang-for-entry-onto-stack	"\C-csM" 'freekbs-map-function-to-relations-made-from-entries-in-region
"\C-csr" 'freekbs-push-region-onto-stack	"\C-cck" 'critic-knowledge-editor
"\C-csy" 'freekbs-push-yank-onto-stack	"\C-ccv" 'critic-unilang-view-recent-entries
"\C-csm" 'freekbs-push-read-from-minibuffer-onto-stack	"\C-ccc" 'critic-critique-entries
"\C-cst" 'freekbs-push-tap-onto-stack	"\C-ccKe" 'critic-ke-edit
"\C-csn" 'freekbs-push-variable-onto-stack	"\C-ccKu" 'critic-ke-unassert
	"\C-cct" 'critic-classify

Planning, Scheduling and Execution

Remember the Complicated World Model

```
andrewdo@justin:/var/lib/myfrdcса/codebases  
/internal$ corpus --senders PSE-X -s . -d 100  
-k pse-x
```

Starting ModManager...

'Get a new laptop'

```
("eases" "107405" "107420")  
("depends" "107407" "107405")  
("eases" "107405" "107408")  
("depends" "107405" "107409")  
("eases" "107405" "107410")  
("costs" "107405" "$400")  
("goal" "107405")  
("prefer same" "107405" "107408")
```

'install FRDCSA on my new laptop'

```
("depends" "107407" "107405")
```

'Present at Flourish'

```
("eases" "107405" "107408")  
("goal" "107408")  
("prefer same" "107405" "107408")  
("ethicality-concern" "107408" "evangelism")
```

'Make a list of the features we want to have in a laptop'

```
("depends" "107405" "107409")
```

'Have mobile wireless access through phone'

```
("eases" "107405" "107410")  
("eases" "107411" "107410")  
("costs" "107410" "$60 / mo")  
("provides" "107410" "107415")
```

'Get an android based phone'

```
("eases" "107411" "107410")  
("costs" "107411" "$200")  
("depends" "107412" "107411")  
("eases" "107414" "107411")
```

'Have FRDCSA Interactive Execution Monitor working'

```
("depends" "107412" "107411")  
("depends" "107412" "107413")  
("goal" "107412")  
("prefer same" "107412" "107416")
```

How to Decide What to Do?

- Some goals are easier to satisfy after other goals have been completed (the “eases” predicate)
- Some goals introduce various costs
- Goals can have complicated dependency structures
- In general, what are my options given the complex world state as defined in the system
- Ideally, there is a 1-1 correspondence between facts in FreeKBS and actual world state and position

Deadlines and Schedules

- Sometimes it is only possible to complete a goal at certain times or when certain preconditions hold
- Sometimes people forget these constraints
- Deadlines, action durations, timing considerations are of course modeled using assertions in FreeKBS
- Federated search of web services and database provide another modality for constraint information

- Verber is the actual planning system behind PSE
- Inspired by and named after the late Senior Chess Master Richard Verber
- Develops (conformant) plans and interactively guides us through their execution



- Uses free software temporal and conformant planners (mainly PDDL and NPDDL)
- Will have a plan cycle in which it plans the period under consideration and the planning process of the next period
- Can run programs as required by plans
- Can compose information sources such as Semantic Web sources or (soon) wrappers for web services like Google Transit/Maps
- Can plan with preferences and constraints

- Rather than a linear plan, conformant plans are plans that reach the goal conditions from any subsequent state within the state space (where reachable)

```
(define (plan generated_plan)
  (:domain robot_navigation)
  (:problem navigation_problem)
  (:body (repeat
    (switch
      (case (and (= (robot_position) dep))
        (done))
      (case (and (= (robot_position) store))
        (action (move_robot_down)))
      (case (and (= (robot_position) lab))
        (action (move_robot_down)))
      (case (and (= (robot_position) NE_room))
        (action (move_robot_down)))
      (case (and (= (robot_position) SW_room))
        (action (move_robot_right)))
      (else (fail))))))
```

- Able to reason with timing information (durative actions)
- Developed (simple) extensions to PDDL3.0
- for managing units on timing information – called format “.verb” symbolizing action
- able to include subdomains (:includes) (similar to Opt - :extends)

World State to Plan Representation

- Verber exports the PSE domain from FreeKBS to PDDL
- Can include hand written domains

Template ~PDDL3.0 Domain and Problem Spec

```
andrewdo@justin: ~
(define
  (domain PSEX)

  (:timing
   (units hours)
  )

  (:requirements :timed-initial-literals :negative-preconditions
   :equality :typing :fluents :durative-actions
   :derived-predicates)

  (:types
   unilang-entry person - object
  )

  (:predicates
   (completed ?e - unilang-entry)
   (depends ?e1 ?e2 - unilang-entry)
   (provides ?e1 ?e2 - unilang-entry)
   (eases ?e1 ?e2 - unilang-entry)
   (plan-start-date)
  )

  (:functions
   (costs ?e - unilang-entry)
   (earns ?e - unilang-entry)
   (budget ?p - person)
  )

  (:durative-action Complete
   :parameters (?e1 - unilang-entry ?p - person)
   :duration (= ?duration 1)
   :condition (and
     ;; ensure that we have made it to the future of now
     (over all (plan-start-date))
     ;; ensure we have enough money
     (at start
      (>= (budget ?p) (costs ?e1)))
    )
  )
)

(define
  (problem PSEX)
  (:domain PSEX)
  (:includes)
  (:timing
   (start-date TZID=America/Chicago:20090313T053356)
   (units hours))
  (:objects work_out_meal_planning update_the_VM_with_the_latest_git_r$
  (:init
   (= (budget andy) 500)
   (= (costs Get_a_new_laptop) 400)
   (= (costs Get_an_android_based_phone) 200)
   (= (costs Have_mobile_wireless_access_through_phone) 60)
   (= (costs Order_mobile_broadband) 60)
   (= (earns Do_job_for_Eric) 250)
   (at TZID=America/Chicago:20090318T183000 (overdue present_at_Chicag$
   (completed Agenda___Breakfast___Make_breakfast___Wed_Mar_11_08_00_00$
   (completed Agenda___Call_Chris_Lampkin___Check_on_Chris___Wed_Mar_1$
   (completed Agenda___Call_Justin___Check_on_Justin___Tue_Mar_10_21_5$
   (completed Agenda___Call_Justin___Check_on_Justin___Wed_Mar_11_21_5$
   (completed Agenda___Dinner___Make_dinner___Wed_Mar_11_16_00_00_CDT_2$
   (completed Agenda___Ensure_Mom_is_hydrated___Ensure_Mom_is_hydrated$
   (completed Agenda___Evening_Meds___Take_your_evening_meds___Tue_Mar$
   (completed Agenda___Evening_Meds___Take_your_evening_meds___Wed_Mar$
   (completed Agenda___Fetch_snail_mail___Bring_in_the_mail___Thu_Mar_1$
   (completed Agenda___Have_Mom_Call_Grandma___Have_Mom_check_on_Grand$
   (completed Agenda___Morning_Meds___Take_your_morning_meds___Wed_Mar$
   (completed Agenda___Organize___Organize_per_Joe_s_recommendation___W$
   (completed Agenda___Read_scan_snail_mail___Read_scan_snail_mail___Th$
   (completed Agenda___Read_scan_snail_mail___Read_scan_snail_mail___We$
   (completed Agenda___Talk_to_Mom___Check_on_Mom___Wed_Mar_11_16_55_0$
   (completed Develop_emergency_procedures_for_psychotics_)
   (completed eventually_convert_all_of_this_into_pse)
   (depends Get_a_new_laptop Make_a_list_of_the_features_we_want_to_ha$
   (depends Have_FRDCSA_Interactive_Execution_Monitor_working Get_an_a$
   (depends Have_FRDCSA_Interactive_Execution_Monitor_working Write_an$
   (depends Make_adequate_money_each_month Setup_core_business_process$
   (depends Make_adequate_money_each_month get_hired_for_various_jobs)
   (depends Present_at_Flourish present_at_Chicago_Semantic_Web_user_q$
  )
)
uu:*-F1 psex.d.pddl Top L1 (PDDL)-----uu:---F1 psex.p.pddl Top L1 (PDDL)
```

(Pass out POSI mailinglist sign up sheet)

Plan Visualization and Execution

Plan Processing

- This stage is largely incomplete, expect to be working well in 1 month (jinx)
- Here is an (old) example plan

0.0003: (MOVE ANDY CS-LOUNGE BAKER-HALL) [0.1500]
0.1505: (UNLOCK ANDY BAKER-LOCKER-18 BAKER-HALL) [0.1000]
0.2508: (UNLOAD ANDY TOWEL BAKER-LOCKER-18 BAKER-HALL) [0.1000]
0.3510: (ARM ANDY TOWEL BAKER-HALL) [0.0000]
0.3512: (LOCK ANDY BAKER-LOCKER-18 BAKER-HALL) [0.1000]
0.4515: (MOVE ANDY BAKER-HALL SQUIRREL-HILL-GIANT-EAGLE) [0.1500]
7.0017: (MOVE ANDY SQUIRREL-HILL-GIANT-EAGLE UC-GYM) [0.1500]
7.1520: (SHOWER ANDY TOWEL UC-MENS-LOCKER-ROOM-SHOWER UC-GYM) [1.0000]

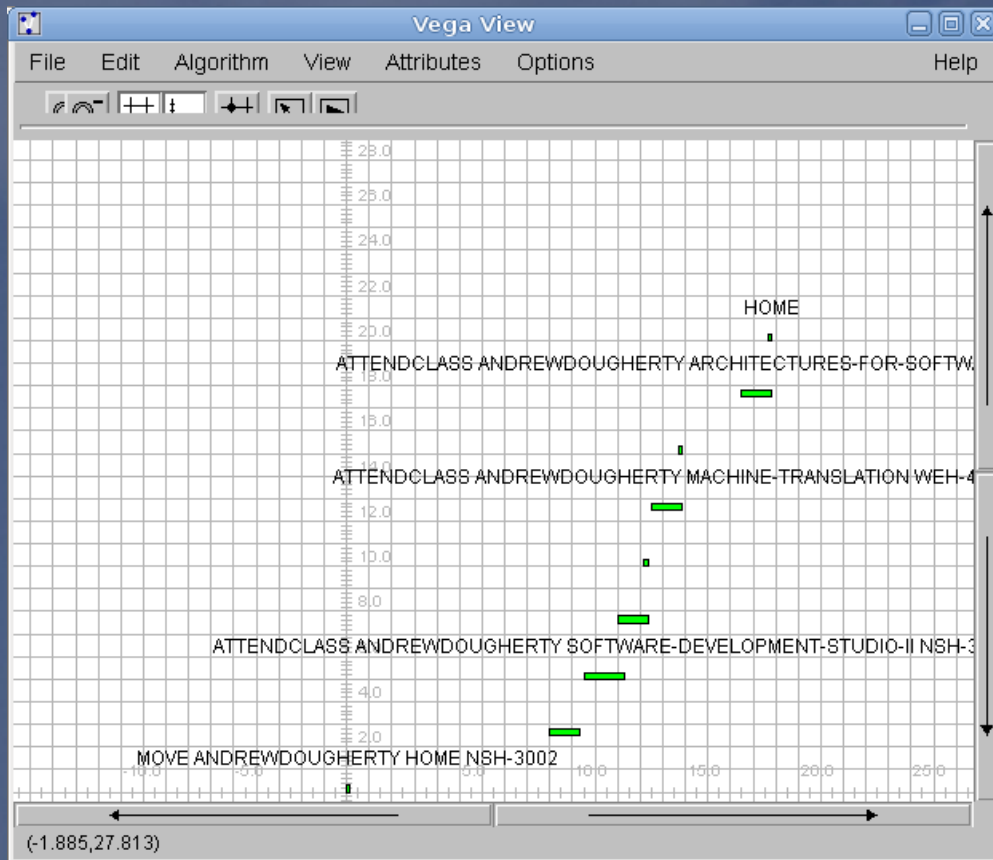
- The scalar time values are translated back into dates (just about to be implemented)

8.2938: (COMPLETE PRESENT_AT_CHICAGO_SEMANTIC_WEB_USER_GROUP
ANDY) [2.0000]

407.7943: (COMPLETE PRESENT_AT_FLOURISH ANDY) [1.0000]

Plan Visualization

- Use a Gantt chart visualizer
- Use free graph visualization for viewing / manipulating dependency relations (currently using Ubigraph but that is non-free)



Plan Visualization

- Export back to a calendar

Google Calendar - Iceweasel

File Edit View History Bookmarks Tools Help

http://www.google.com/calendar/render

Most Visited Getting Started SEARCH Latest Headlines

Inbox (93737) - Ya... Spam (376) - Yah... Your Meetup Grou... The Chicago Sem... Google Calendar

adougher9@gmail.com | Offline Beta | Sync | Settings | Help | Sign out

Google Calendar BETA

Search My Calendars Show Search Options

Create Event Quick Add

March 2009

Today Mar 15 - 21 2009 Refresh Print Day Week Month 4 Days Agenda

Sun 3/15 Mon 3/16 Tue 3/17 Wed 3/18 Thu 3/19 Fri 3/20 Sat 3/21

IDCAI-09 Author Feedback Perio Movie-Watch e-r GNOME 2.26 rel Movie Watch a

Freezing Our H Ostara a

GULF WARS VIII

4pm

5pm

6pm

7pm

8pm

9pm

My calendars

Andrew Dougherty

Other calendars

Add a friend's calendar

2007 tech conferenc

Find: rate Previous Next Highlight all Match case Phrase not found

Done Adblock

Interactive Execution Monitor

- Software for walking the user through plans
- Because it knows the status of the user (where she is, what communication options she has, etc), can plan for how to contact her (intend to do develop a domain for this planning how to contact user)
- Runs on desktop, alerts user using “adjustable autonomy” and advisability
- Knows if the user is AFK

Interactive Execution Monitor



- Developing Android software for interactively walking the user through plans
- Submitted GSOC application including this and Android speech recognition as project
- Provide options for re-planning, updating world state

- What is the most advanced AI planning language?
- What are the best freely available (hopefully free software) planners?
- Will we be able to do preferences somehow with LPG-td?
- Will SGPlan5 work on our domains?
- Can we do **temporal** conformant planning?
- Can we do mixed-initiative planning?

- Increased ability to translate natural language into domain and problem definitions, to make use of the actual contents of UniLang goal statements
- Multi-agent planning
- Adoption, usability
- Increased integration with other project areas
- Meal planning, Pedagogy planning, etc.
- More powerful planning systems, more optimal, more featureful, and support systems

- Interested in sharing tools and involving other developers to improve their time management and the tool
- Integration with other apps
- Better integration with existing SemWeb standards and tools (simple examples are capitalization, datetime formats, OWL reasoners, etc)
- Contact me if interested: andrewdo@frdcsa.org

Collaborative Infrastructure

- Despite having dozens of functioning, capable systems, limited distribution and lack of documentation are crippling the project
- Private information strewn through FRDCSA release, needs to be deidentified
- Started POSI (POSI Open Source Initiative) with several hackers to promote collaboration, uses FRDCSA core systems

POSI Open Source Initiative

- Elicit from users their goals, interests and abilities (into a knowledge base)
- Guide them (perhaps also using Verber) to satisfy their collective goals
- Learn their abilities from their writings (IRC Logs, AIM chat logs, email, writings, resumes) using NLP software

Google Summer of Code Project Proposals

● Feel free to consider any of these anyways

Difficulty: Easy

Intelligent Tutoring Systems (SystemX) - write natural language processing software that maps out subject areas, pedagogical materials, and dynamically composes lessons and tests that fit within a temporal plan for overall subject coverage. Develop domains for teaching programming, system administration, etc.

If this is written (in fact a lot has been written, but still some work remains), we will be able to offer to the community software that helps users rapidly ramp up on certain technology areas. Note that this will work with the [CLEAR system](#). This will be useful for training for companies and projects and will result in better skills. We are already using [CLEAR](#) and it's associated tools (like its Firefox extension) to great success.

GNU+Linux/Android Speech Recognition - for Android - get a basic small vocabulary grammar based speech recognition system (like CMUSphinx) working for the Android phone (using the simulator). Or, help with the acquisition of speech data for training free software Large Vocabulary Continuous Speech Recognition systems.

Developing voice recognition will enable the Verber Personal Planner to walk the users through doing various tasks, such as cooking (in the case of Gourmet), or any other task, by integration and RPC with their phone.

POSI Core Systems - Write software (semweb stuff) for the modeling of users Goals/Interests and Abilities for POSI systems such as it's IRC bots and web interfaces. Then help to organize contributors by accumulating data about their interests and shared goals (through free [textual entailment recognition](#) systems) and developing algorithms to schedule collaborative work on these tasks.

If the POSI Core Systems function, it will be easier for the project to identify the research intentions of its community and establish successful projects.

Difficulty: Medium

[RADAR/Packager](#) - extend the existing RADAR/Packager tool to provide better automation of automatic Debian package creation from online software sources. Package hundreds of free software packages and place in the online repository.

If RADAR/Packager are making more packages, it will ensure both the portability of the FRDCSA project, as well as massive reuse of the software that we package (provided we advertise our repository well enough).

Google Summer of Code Project Proposals

Experience Modeling System (EMS) - develop software that, by analyzing writings (such as IRC logs, AIM chats, email, and text documents), as well as resumes and user's input, models what areas the subject is familiar with. Develop a subject area classification by mixing the LCC (Library of Congress Classification) system, with existing FLOSS subject ontologies, and extraction by text processing Wikipedia with techniques such as LSI (latent semantic indexing).

The EMS works with other areas of POSI such as the core systems and the [Job Search system \(Sample Resume/Job Matching system\)](#). It is vastly important for figuring out who can perform specific tasks. Who knows, maybe GSOC itself will use this eventually.

[Gourmet Meal Planner](#) - Develop the worlds first free and open source food ontology that can be used for various free software meal planners like [Gourmet Meal Planner](#) and [Gourmet Recipe Manager](#). Do this by writing a module for Gourmet Recipe Manager that propagates ingredient description to SR20 entry mappings to a server.

If this is done, we will have done about as much as is mathematically possible to eliminate hunger, or at least bad-diet.

Difficulty: Hard

[Verber Personal Planner](#) - (Verber) develop interesting extensions to the existing Verber personal task manager, develop PDDL3.0 and Opt domains that include more commonsense information and increase coverage. Write an Android Interactive Execution Monitor application to walk users through resultant plans, update world state, and allow them to initiate replanning. Integrate mixed initiative planners, and communicate with leaders in the field of AI Automated Planning and Scheduling. Integrate with existing calendaring systems.

Verber is going to be a life saver for people with ADD, Schizophrenia, or Autistic Spectrum Disorders. It will provide the so-called execution function or "time-management skills" that they so injuriously lack. However, it will boost the time management skills of people without these conditions...

Natural Language Processing - (Sayer/Thinker) Develop state of the art natural language processing techniques for converting text to semantic representations and invoking theorem proving for understanding, as part of the Formalize/Sayer/Thinker systems.

Proper understanding of natural language will prove pivotal in the automatic creation of PDDL3.0 domains from goal statements (to help Verber along), as well as an abundance of other projects.

FRDCSA Virtual Machine Release - Complete the release of the entire FRDCSA on a Virtual Machine. While it is already in progress, this has been the achilles heel of the project - so far only privileged developers have access to the full project.

About the author

- Presenting and table at Flourish on FRDCSA/POSI
- Starting POSI Consultancy (POSIC) in order to promote the development and usage of free software and AI systems
- Developing software to automate business processes
- Looking for clients
- Looking for consultants
- Looking for users
- Looking for developers

Andrew J. Dougherty
A.I. Researcher
FRDCSA/POSI/POSIC
630.300.5565
andrewdo@frdcsa.org
<http://frdcsa.org/~andrewdo>

The End

Thank you!!!