

# The Planning Community Related FRDCSA Projects

Andrew J. Dougherty

2024-11-21

## Contents

<b>1</b>	<b>Core Projects</b>	<b>1</b>
1.1	Planning Software Knowledge Base . . . . .	1
1.2	Unofficial Reference Manual about Planning and Modeling Tricks and Techniques . . . . .	2
1.3	Semi-Auto-Containerization of Planning Software for planutils	2
1.4	Analysis of theplanningcommunity.slack.com Logs . . . . .	2
1.5	Planning Knowledge Engineering in Prolog . . . . .	3
<b>2</b>	<b>Comments</b>	<b>3</b>
<b>3</b>	<b>Contributing</b>	<b>3</b>
<b>4</b>	<b>Notes</b>	<b>4</b>

## 1 Core Projects

### 1.1 Planning Software Knowledge Base

- A Prolog/Cyc KB of all indexed planning software and their properties, initially using a web crawler and a system implementing Text2KG (e.g. Knowledge Graph Construction/Population), Text2KB (e.g. Knowledge Base Construction/Population) or Text2Logic, or with Claude's help to convert text to Prolog/Cyc, followed by manual editing:
  - Web-Agent: [WebAgent2](#)
  - Text2KG: for instance: [itext2kg](#)

## 1.2 Unofficial Reference Manual about Planning and Modeling Tricks and Techniques

- Example question: How can we create a domain which references specific ground constants in fluents for planners that do not do grounding in domains, like LPG-td-1.4 (as opposed to LPG-td-1.0)?
  - Also KB-based, maybe roundtrip from KB to text, using Org-mode and Prolog and parsers and generators
  - Using LLMs to map out the space of planning techniques:
    - \* e.g. this KB regarding PDDL compilation techniques:
      - [Planning Compilations "Ontology"](#)
    - \* Ask Claude 3.5 Sonnet to help formulate this using theplanningcommunity.slack.com logs
    - \* LLM-based Text Mining of Planning Literature by Focused Crawling using AAA Web-Agent
      - again: [WebAgent2](#)

## 1.3 Semi-Auto-Containerization of Planning Software for planutils

- Using agentic LLM software: [AAA/SPAMI/Prolog-Agent](#)
  - Same tool we are developing for semi-auto-containerization of <https://tptp.org/cgi-bin/SystemOnTPTP>
  - Intend to expand to other communities, such as ASP, Model Checking, and beyond

## 1.4 Analysis of theplanningcommunity.slack.com Logs

- Retrieval Augmented Generation (RAG) over logs
- Social Network Analysis (SNA) over logs
  - If desirable
  - Similar to [analysis performed for FRDCSA-Logismoo Discord logs](#)
- Generate Parts of Reference Manual as already Mentioned

## 1.5 Planning Knowledge Engineering in Prolog

- Somewhat mature KBS Implementation
- Parsers / generators for PDDL in Prolog
  - Declarative rewrite in "pure" Prolog of our PDDL $\leftrightarrow$ Prolog ensemble
- World state monitor / interactive execution monitoring using Prolog "microtheory" implementation

## 2 Comments

- No guarantees of completion
- Not starting from scratch, starting with [FRDCSA](#) 25-year-old system of systems with many preexisting subsystems that already implement many of these capabilities/features
  - [~150 released to GitHub of extant internal/minor systems](#)
  - [Early/outdated VirtualBox VM Release of these ~150 FRDCSA Systems](#)
- Already have much working here:
  - [FreeLifePlanner Reference Manual](#)

## 3 Contributing

- We welcome contributions in any of the following areas:
  - Knowledge base development
  - Tool containerization
  - Parser improvements
  - Documentation
  - Testing and validation
  - Anything else

## 4 Notes

- This is a living document that will be updated as projects evolve and new priorities emerge.