Goal-directed Software Assistant for a Plan Advisory System

MAS.761 Final Project

Peng Yu
Contents

• Background

• Problem statement

• Approach

• Demo and conclusion
Plan Advisory System for Deep Sea Explorations

- I am working with the Woods Hole Oceanographic Institute to apply state-of-the-art planning and scheduling algorithms on managing expedition cruise plans.
During an expedition cruise, the chief scientist needs assistance for planning and scheduling to maximize the science return, esp. when things go wrong.

- Task sequencing and scheduling.
- Goal relaxation and failure recovery.
- Human resources and assets management.
The Interface
There are more than 200 operations in the UI!

The learning curve is very steep, and no scientist is willing to devote that much time studying the operations.
There are two challenges for the users of a high-functionality program:

- Identify the problem(s) that can be addressed by the program.
- Understand the operations to achieve the desired outcome(s).
Previous Approaches

• Previous approaches to this problem focus on the second challenge:
  – Tutorials with step by step instructions are widely used to assist the users in completing their tasks.

• The first challenge is caused by a gap between the users’ goals and the problems solvable by these tutorials.

I want to address my problem: X.
Previous Approaches

- There is a mismatch between the users’ and programs’ problem descriptions.
  - Command dictionary and keyword search are commonly used, but not very helpful.
• Use commonsense reasoning to resolve the mismatch between the users’ and programs’ knowledge bases, hence bridge the gap between their problem descriptions.
Tell me your problem:

storm.is.coming

Solution:
The original problem is "storm,is,coming".
It is explained by concepts "storm", "bad weather", "high wave".
Their consequences include "ship down time", "rov down time", "aav down time", "ctd down time", "core down time", "multibeam down time",

Use the following tools to address the issue, prioritized based on likelihood:

1. At down time scheduling panel:
   Select CORE as affected asset;
   Set the start and duration of the downtime;
   Set the uncertainty in the duration of CORE downtime
How it works?

• Given a problem description:
  – **Explain** its meaning by identifying similar known concepts;
  – **Explore** causes and effects through neighboring assertions with causal relations.
  – **Generate** solutions and instructions using neighboring assertions with resolution relations.
Step 1: Find similar concepts

- Explain the user’s description using known concepts in Kirk-kb.
- The similarity is evaluated using the C4-Kirk-Blend kb.

| typhoon -- low battery: 0.744272348609702 |
| ________________________________________ |
| typhoon -- low water: -0.3701262745244257 |
| typhoon -- multibeam down time: -0.000103761445298952 |
| typhoon -- multibeam down time scheduler: -0.145049072169696 |
| typhoon -- rain: 0.6463570842311839 |
| typhoon -- rov down time: -0.00010376144529895844 |
| typhoon -- rov down time scheduler: -0.14321520410262634 |
| typhoon -- satellite link down: -0.13368756460728107 |
| typhoon -- shaft overheat: -0.3701262745244257 |
| typhoon -- ship communication failure: -0.004030046230603112 |
| typhoon -- ship down time: -0.00010376144529873296 |
| typhoon -- ship down time scheduler: 0.13919460936443573 |
| typhoon -- ship electrical failure: -0.015660917350911102 |
| typhoon -- ship human factor failure: -0.0045961307498643845 |
| typhoon -- ship mechanical failure: 0.004535704379857019 |
| typhoon -- short circuit: 0.490685171277081 |
| typhoon -- sonar broken: -0.13368756460728107 |
| **typhoon -- storm: 0.9026540259783965** |
| typhoon -- strong current: 0.5736174980949411 |
| **typhoon -- thunderstorm: 0.9458139330890203** |
Step 2: Find related concepts

- Expand the coverage of similar concepts by finding related concepts in Kirk-kb:
  - by identifying assertions with \textit{IsA}, \textit{DefinedAs} and \textit{ConceptuallyRelatedTo} relations.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{diagram.png}
\end{figure}
Step 3: Identify causes and effects

- Identify concepts that are possible causes and effects using neighboring assertions with causal relations, such as Causes, CreatedBy, HasA, etc.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Causes</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>storm</td>
<td>ship down time</td>
<td>0.19151911609834685</td>
</tr>
<tr>
<td>thunderstorm</td>
<td>rov down time</td>
<td>0.19151911609834685</td>
</tr>
<tr>
<td></td>
<td>auv down time</td>
<td>0.218064750209997</td>
</tr>
<tr>
<td></td>
<td>ctd down time</td>
<td>0.218064750209997</td>
</tr>
<tr>
<td></td>
<td>core down time</td>
<td>0.19151911609834685</td>
</tr>
<tr>
<td></td>
<td>multibeam down</td>
<td>0.19151911609834685</td>
</tr>
</tbody>
</table>

storm; thunderstorm; bad weather; ship down time; rov down time; auv down time; ctd down time; core down time; multibeam down time;
Step 4: Find solution concepts

- Find solution concepts by exploring neighboring assertions with resolution relations, such as UsedFor and CapableOf.

```
ship down time scheduler, UsedFor, ship down time: 4.0147734018392524e-32
rov down time scheduler, UsedFor, rov down time: 2.5236755089281305e-32
auv down time scheduler, UsedFor, auv down time: 2.6670725263397843e-33
ctd down time scheduler, UsedFor, ctd down time: 9.190983966244161e-33
core down time scheduler, UsedFor, core down time: 1.102739126848916e-31
multibeam down time scheduler, UsedFor, multibeam down time: 5.591052950934684e-32
```
Step 5: Generate instructions

- Maps each solution concept to a set of pre-defined instructions, stored in a separated hash table.

At down time scheduling panel:
Select SHIP as affected asset;
Set the start and duration of the downtime;
Set the uncertainty in the duration of SHIP downtime
Limitation

• Efficiency:
  – When Kirk-kb gets large, the expansions will take much longer to compute due to the lack of contraction steps.

• Accuracy:
  – More than half of the solutions generated by this approach are not accurate. This is mainly due to the inconsistent inference result before and after blending c4 and Kirk-kb.

• Coverage:
  – C4 only has a limit coverage of shipboard concepts. For example, “Engine temperature high” is not known by c4.
• Design and implemented an assistant program that generates instructions for the users of a high-functionality planning software. The program is capable of:
  – bridging the gap between the user’s problem descriptions and the concepts used by the software.
  – explaining the user’s problem with known concepts, causes and effects through commonsense reasoning.
  – generating solutions and instructions for the software to address the user’s problem.
Question