Planning and Scheduling
Planning Project

Prof. Dr.-Ing. Gerhard K. Kraetzschmar — M.Sc Iman Awaad

Chaitanya Hebbal — Mazin Eltayeb — Hashem Talaat

9th January 2017

Choosing a planner

In this project, we started by looking into the results of the 2014 IPC planning competition [1] for the category sequential optimal. The approach was to start at the top of the results list and use the first planner we can compile and run as all planners in the category sequential optimal should be suitable for the given problem.

At first we installed SymBA*-2 [3] (After a struggle in compiling the code. Where eventually we had to correct one line of code for a syntax error !!). However, after trying the planner on the first problem given, the resulted output file was not understood and it was not possible to recognize a plan easily. Moreover, the planner severely lack documentation (and not even a readme file was provided). Therefore, SymBA*-2 was abandoned and we moved into cGamer [4] as SymBA*-1 [3] is just an earlier version of SymBA*-2 and suffers from the same problems mentioned above.

In summary, cGamer was the chosen planner as it is the highest performing planner (according to competition results) that we were able to compile, run and use.

Planner Installation & Challenges

As expected compiling the planner start with a ton of errors. These errors were fixed by:

1. Installing missing package: flex and gcc-multilib.

2. Editing the cmake files to be compatible with 64 bit processor as instructed in the readme file fo the planner.

3. Editing the cmake files to use the correct jdk directory.

After successful compilation, the plan was run successfully on the simple domain (with no action costs and no constraints) and was able to produce results for all three problems in the simple domain.

However, when running the planner for with the extended domain that considers action costs (distance), the planner failed to parse the PDDL problem definition files for all three problem. Although, the files are syntactically accurate as it worked with other planners.

We suspect that the planner supports only a subset of PDDL syntax. For the extended domain we solved using MIPS and METRIC_FF planners provided as part of the itSimple planning IDE.

Modelling domains and problems

Fro modelling the domains and problems, we used itSimple planning IDE [2]. The IDE has uses UML diagrams to model the domains and create problem instances. Following a one hour tutorial, it was straight forward to model the simple domain with no constraints and no actions' costs.

The domain has three object types: robot, object and station. Only robot is an active agent and can : move from one station to another station, pick an object and drop an object. Figure 1 below, shows the use case diagram for the all domains.

To consider distances, the domain was altered by adding a utility function that calculates the distance between two location. A utility attribute was added to keep track of the travelled distance. An effect was added tot he move action to increase the total distance by the amount of the distance travelled.

The constraint on the number of objects the robot can carry is implemented by adding two attributes one represent the capacity of the robot and another for the current load. The capacity is fixed at 3. A new precondition for the pick action was added to make sure the current load is less than the capacity. Effects were added to pick action to increase current load and to drop action to decrease current load. Figure 2 shows the final class diagram.
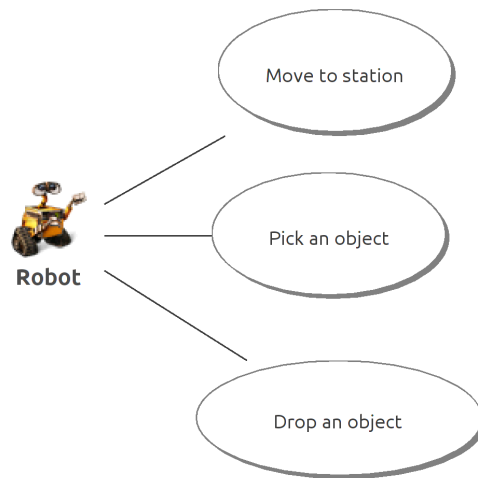
Figure 1: Robocup at work domain: Use case diagram

Problems are defined by instantiating the class diagram with all possible objects and then creating snap shots of the initial state and the goal state. Figure 3 below shows the initial state for problem 1 while figure 4 shows its goal state.

The models of domains and problems are then translated to PDDL. All PDDL files can be found in the files attached to this report.
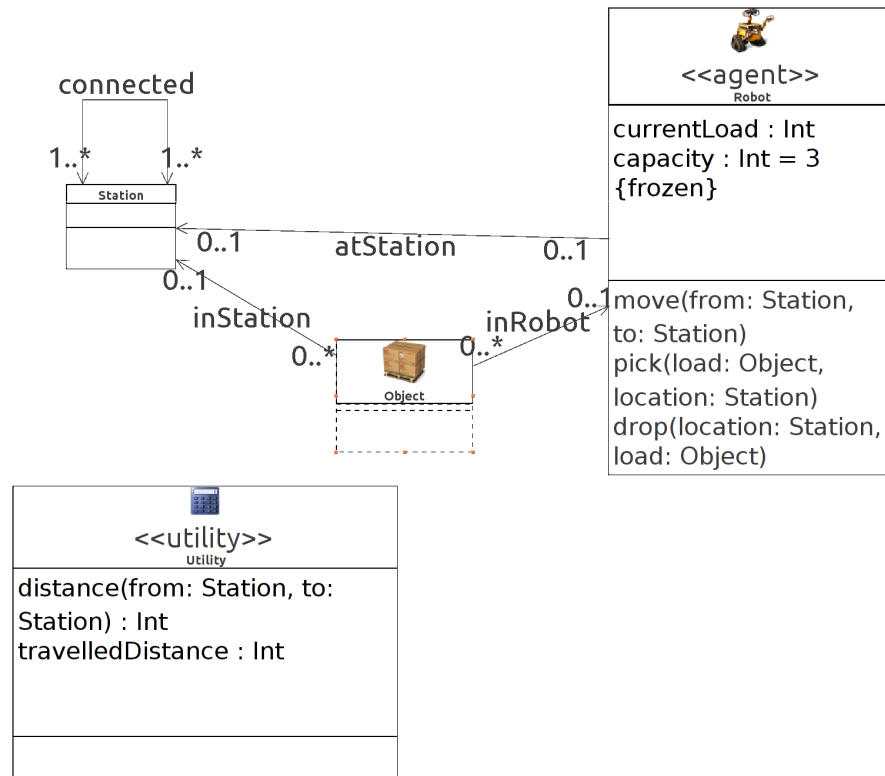
connected

1..*   1..*

Station

0..1   atStation   0..1

0..1

inStation   inRobot

0..*   0..*

Object

<<agent>>
Robot

currentLoad : Int
capacity : Int = 3
{frozen}

0..1   move(from: Station,
to: Station)
pick(load: Object,
location: Station)
drop(location: Station,
load: Object)

<<utility>>
Utility

distance(from: Station, to:
Station) : Int
travelledDistance : Int

Figure 2: Robocup at work domain: Final class diagram

Results

A summary sheet of the results is attached in the end of this report.

Metric_FF was found to be the fastest and most stable planner in (cGamer, MIPS and METRIC_FF). cGamer was not able to run the extended domain and raised a parser exception. MIPS could not find a plan for the third P03 with distances and capacity constraints Metric_FF planned successfully for all the problem and all domains.
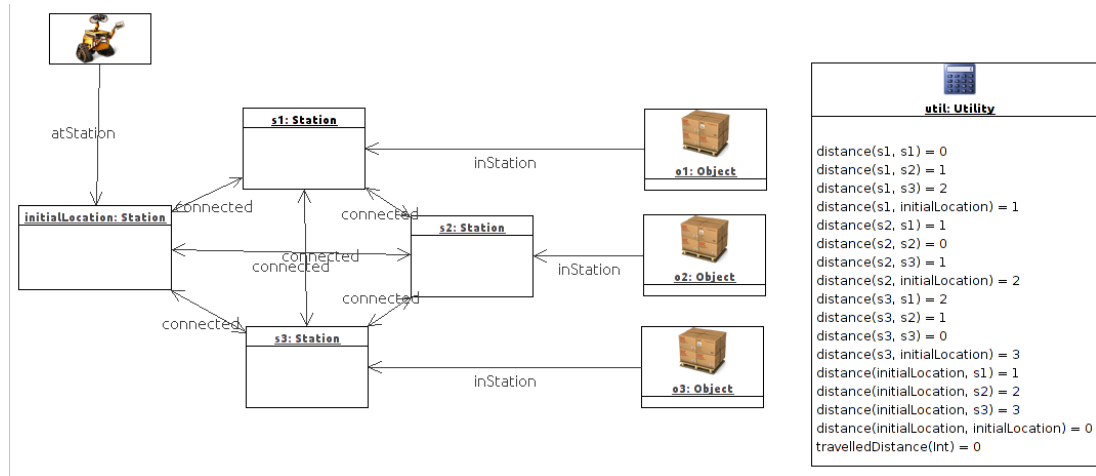
Figure 3: Robocup at work domain: P01 Initial state

References

[1] Ipc 2014 - https://helios.hud.ac.uk/scommv/ipc-14/resdoc.html.

[2] itsimple - https://code.google.com/archive/p/itsimple/.

[3] Alvaro Torralba, Vidal Alcazar, Daniel Borrajo, Peter Kissmann, and Stefan Edelkamp. Symba: A symbolic bidirectional a planner. In *International Planning Competition*, pages 105–108, 2014.

[4] Alvaro Torralba, Vidal Alcázar, Peter Kissmann, and Stefan Edelkamp. cgamer: Constrained gamer. *International Planning Competition (IPC)*, pages 74–76, 2014.
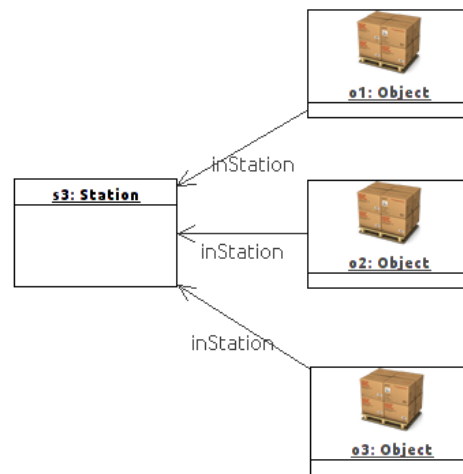
Figure 4: Robocup at work domain: P01 Goal state

1) AtWork Domain with no actions' cost - Planner used: Gamer

Problem	Solution Steps	Peak Node Count		Total time
P01	7	5,110.00		0.270
P02	10	6,132.00		0.312
P03	22	2,097,144.00		2.336

1) AtWork Domain with no actions' cost - Planner used: MIPS-XXL Ve

Problem	Solution Steps	Peak Node Count		Total time
P01	7			0.019
P02	10			0.027
P03	30			0.789

1) AtWork Domain with no actions' cost - Planner used: Metric-FF

Problem	Solution Steps	Peak Node Count		Total time
P01	7			0.033
P02	10			0.014
P03	22			0.022

2) AtWork Domain with actions' cost - Planner used: MIPS-XXL Versi

Problem	Solution Steps	Peak Node Count		Total time
P01	7			0.019
P02	10			0.008
P03	30			178.083

2) AtWork Domain with actions' cost - Planner used: Metric-FF

Problem	Solution Steps	Peak Node Count		Total time
P01	7			0.055
P02	10			0.017
P03	22			0.038

3) AtWork Domain with actions' cost - Planner used: MIPS-XXL Version: IPC-6 v1.2 Linux

Problem	Solution Steps	Peak Node Count		Total time
P01	7			0.012
P02	12			0.010
P03	NA			NA

3) AtWork Domain with actions' cost - Planner used: Metric-FF

Problem	Solution Steps	Peak Node Count		Total time
P01	7			0.009
P02	12			0.009
P03	29			0.014