## A Task Robot Planner for a Scientific Conference



### Eric Beaudry and \_\_\_\_\_ July – August 2004 (project for IFT702)

## Introduction

- This project is related to Eric Beaudry's Master Project : Task Planner for Mobile Robotics
- A goal of EB's project is to participate at the « AAAI Challenge » competition

## Objectives

- Evaluating some planners applicable to a simplified version of AAAI Challenge
- Trying to model the domain in planner language (PDDL)
- Finding problems caused by limitations of the chosen planner and it's language
- Making some criticism

Participate to a Scientific Conference

- This is a simplified and idealized domain compared to AAAI Challenge
- Robot Tasks to plan
  - Making a presentation
  - Assisting to some presentations
  - Showing a poster
  - Taking pictures of interesting posters
  - Socializing with other attendees
- Full description of scenario :
  - http://siela.dmi.usherb.ca/~eric/ift702/scenarios.html

### Scientific Conference



## Scientific Conference main Tasks

- Make-Presentation & Assist-Presentation
  - At a fixed time (at the beginning of a presentation session)

## Scientific Conference main Tasks

### Show Poster

2 subtasks that have to be done in a time window





## Scientific Conference main Tasks

- Socialize
  - When nothing to do

## **Domain Characteristics**

- Time (action duration + scheduling)
- Resources management (battery)
- No uncertainty
  - At planning time, we assume that environment is already known, static and actions never fails
  - Uncertainty is handled at execution time by well known robotic techniques (like behaviors-based, FSM, ...)

## Planning with Time and Resources

### State-oriented

- View states at a single time
- Adding time and resource variables in the state representation
- Time-oriented
  - View one state along the time line
  - Temporal databases and Chronicles

## **Considered Approaches**

- Chronicles-Based
- State-Based with PDDL compatible planners
- Task-based (HTN)

## Chronicles

- A domain is described by a set of state variables, each being a function of time in a chronicle way
- Use state variables : more expressiveness and conciseness than set-theoretic representation
- e.g.: location-r1: time->locations
- Several chronicle planners
  - IxTeT, ParcPlan, EUROPA, and RAP

## An Example of Chronicles

A scenario partially specifying the whereabouts of r1,c1,and Uranus is represented with three state



The slide taken from Automat ed Planning: Theory and Pract

State-Based with PDDL compatible planner

- PDDL = Planning Domain Description Language
- PDDL is a language for International Planning Competition
- It's easy to find many planners that support PDDL

## Categories of PDDL planner

PDDL 2.1 Level	<u>Characreristics</u>			
Level 1 (STRIPS)	Provide the same level of features like STRIPS			
Level 2 (Numeric)	<ul> <li>Provide the possibility to add numeric values in problems</li> <li>Useful to manage resources</li> </ul>			
Level 3 (Durative action)	Provide the possibility to handle durative and concurrent actions with discrete changes			
Level 4 (Continuous durative action)	<ul> <li>Support continuous change</li> <li>Never used in IPC competitions</li> </ul>			
Level 5 (every	Include every previous level			

## PDDL 2.1 level 2 vs level 3

- We consider both levels for our domain
- Level 2 seems to be more appropriate

## PDDL 2.1 level 2 vs level 3

- Level 2
  - Handle numerics (important for resource)
  - Time planning is possible by adding a « current-time » variable

- Level 3
  - Do not handle numerics
  - Focus on overall duration of the plan with concurrent actions
  - Is not designed to manage deadline and scheduling

## Task-based (HTN)

- HTN = Hierarchical Task Network
- Decomposing tasks into subtasks recursively until reaching primitve tasks
- A recipe for problem-solving
- SHOP2 and STN (Simple Task networks)
- Total-order and partial-order

### Planners available

- Show the web page :
- <u>http://siela.dmi.usherb.ca/~eric/ift</u> <u>702/</u>

## The choice of the planner

- Our first choice was using a planner based on Chronicles like IxTeT
- Since we can not get this planner, we switch to our second choice : State-Based planner with Numerics (PDDL 2.1 level 2)
- We look at IPC-3 results for planners with good performances in numeric domains
- Metrics-FF is chosen

## ScientificConf.pddl (1)

- Object Types
  - Location
    - Places that the robot can go to
    - Room, Stand, Waypoints, Registration, ...
  - Pres-Session
    - Presentation session. This is an object that represents an interval of time
  - Post-Session
    - Poster session (similar to Pres-Session)

## ScientificConf.pddl (2)

### Predicates

- (robot-at ?loc location)
  - Tell that the robot is currently at ?loc
- (link ?x ?y location)
  - There is a direct path between x and y
- (registered)
  - The robot has his badge?
- (registration-place ?loc location)
  - The robot can register at this location

## ScientificConf.pddl (3)

### Predicates (...)

- (presentation-done ?ses pres-session ? loc – location)
  - The robot have done his presentation at this location for the session
- (rechargeable-place ?loc location)
  - The robot can recharge his battery at this place?
- (socializing-place ?loc location)
  - A place that we can schmooze ?

## ScientificConf.pddl (4)

### Predicates (...)

- (assisted-to-pres ?ses pres-session ? loc - location)
  - Tell if the robot has assisted to this presentation. This is similar to presentation-done.

## ScientificConf.pddl (5)

- Actions
  - GOTO
  - REGISTER
  - ASSIT-PRESENTATION
  - MAKE-PRESENTATION
  - WAIT-PRES-SESSION, WAIT-POST-SESSION
  - RECHARGE
  - TAKE-PICTURE-POSTER
  - FIX-POSTER
  - REMOVE-POSTER
  - SOCIALIZE15 (schmoozing for 15 minutes)



## Plan example

- **1. GOTO STARTPLACE REGISTRATION**
- 2. REGISTER REGISTRATION
- 3. GOTO REGISTRATION STANDROOM\_1
- 4. WAIT-PRES-SESSION PRESSESSION0
- 5. MAKE-PRESENTATION STANDROOM 1 PRESSESSIONO
- 6. GOTO STANDROOM 1 STANDROOM 2
- 7. WAIT-PRES-SESSION PRESSESSION1
- 8. MAKE-PRESENTATION STANDROOM\_2 PRESSESSION1
- 9. GOTO STANDROOM\_2 STANDROOM\_3
- 10. WAIT-POST-SESSION POSTERSESSION0
- 11. MAKE-PRESENTATION STANDROOM\_3 PRESSESSION2

## **Metric-FF Planner**

- Metric-FF is the numeric version of FF planner
  - FF means « Fast Forward »
  - This planner does a forward search in statespace
  - Work with heuristics based on relaxed GraphPlan algorithm that ignores delete lists
- Metric-FF is one of the two planners with the best performance at IPC-3 in numeric domains

## Algorithm of Metric-FF

### Enforced Hill-climbing search

Initialize the plan to empty

S:=I

**While** h(S) !=0 **do** 

perform breadth first search for S1 with h(S1)<h (S)

if no such state can be found when then

output Fail, stop

#### endif

add the actions on the path to S1 at the end of the plan

S:=S1

### Endwhile

- Pruning Techniques based on heuristics
  - Helpful actions: to select a set of promising successors
  - Add goal deletion: to cut out branches where goals achieved early

FF – Heuristic based on ignoring delete list (1)

- FF heuristic is based on a relaxed GraphPlan algorithm that ignore delete list
- Exemple
  - Initial state : robot is at StartPlace
  - Goal : show a poster at stand P20



FF – Heuristic based on ignoring delete list (3)

- In the previous slide, from the initial state, we can have more than 50 applicable actions (with full navigation)
- To decide the order of which nodes to visit, FF will compute and heuristic for each application action
- So, FF will call a relaxed GraphPlan for every possible successor



# on ignoring delete list idea (1)

- We can make some analogies
  - Decreasing : negative effect (delete list)
  - Increasing : positive effect (add list)
- Preconditions
  - A unsatisfied > precondition we look for action that have increasing effects
  - A unsatisfied < precondition we look for action that have decreasing effects

## on ignoring delete list idea (2)

- In fact, it is more complicated...
  - Some actions need to be called many time to increase a value
  - Decreasing a negative value as same effect that increasing

## on ignoring delete list idea (3)

- Metrics in relaxed GraphPlan
  - We apply all applicable actions at each propositions layer
  - We keep the Min and Max value of every possible numeric value
  - > preconditions we be tested with Max
  - < preconditions we be tested with Min</pre>

## Limitations

- During the project, we face many limitations from PDDL and/or Metric-FF
  - Difficulties to model domain in PDDL
  - Some performance problems
  - Some limitation of planners

Difficulties to modeling domain in PDDL

- No numerics permitted in action parameters
  - We cannot model an action like : Sleep(x) where x is a numerical value
  - Parameters have to be objects in finite set

## domain in PDDL

- PDDL is Goal-based, not Task-based
  - For some actions, it seems to be easier to specify tasks than goals
  - Example: Showing a poster requires three sub-actions:
    - GOTO stands where to show a poster
    - Fix the poster on the board
    - Remove the poster after

## Show Poster in PDDL

#### Fix-Poster(ps1, P24)

#### :precondition(and

—(robot-at ?loc) (not (poster-fixed ps1 P24))

... :effect(and (poster-fixed ps1 P24)

### GOTO(?from, P24)

### :precondition(a

ስd (robot-at ?loc)

effect(and (robot-at P24)

Goals from user

(poster-done ps1 P24)

#### **Remove-Poster(ps1, P24)**

:precondition(and
 (poster-fixed ps1 P24)
 (robot-at P24)

**:effect**(and (poster-done ps1 P24)





## Show Poster in PDDL (2)

- While we already know that showing a poster needs two actions, we have to add a predicate *poster-fixed* that links both actions
- It seems better to write a HTN method « Show-Poster() »



## Metric-FF is not designed to handle Time

- Metric-FF is good for numerics but not recommended to do time scheduling
  - Simple problem with few goals takes a lot of time and a lot of memory in planning
- Planning with Chronicles seems to be better for the Scientific Conference domain
  - Majority of tasks can be put on a time line before running the planner
  - The planner only has to resolve flaws by adding actions





## planning the socialize action

- An expected feature of the robot is to be able to socialize with other attendees when time permits
- Since socialize action is optional, it's not possible to specify directly this goal
- We face some difficulties with Metric-FF to generate plans that include socialize action

planning the socialize action

- Example :
  - Goal is to make a presentation at 10:30 in Room3
  - Current time is 8:15

### Correct Plan that Metric-FF can Generate :

	Goto(Room3)	Wait(10:30)	MakePresentation()
--	-------------	-------------	--------------------

### Better Plan that we expect :

	Socialize()	Goto(Room3)	Wait(10:30	lakePresentation()
--	-------------	-------------	------------	--------------------

## planning the socialize action

- First approach
  - Metrics planners can optimize plan by minimizing or maximizing a numeric value (this is in PDDL 2.1 level 2 specifications)
  - Adding a «score» variable that is increased when we add action Socialize()
  - Adding «(:metric maximize (score))» in the problem file to force the planner to do optimization
  - This approach failed due to a limitation of Metric-FF : this planner cannot maximize an increasing value

## PDDL code for score approach

### PDDL domain file : socialize action

#### (:action SOCIALIZE

```
:parameters(?loc - location)
:precondition(and
  (robot-at ?loc)
  (socialisable-place ?loc)
  (registered)
  (>= (battery-level) (* 900 (energyconsumed-per-second)))
  (<= (currenttime) 56700) ; before 15:45.... conference ending at 16:00!
  )
:effect(and
  (increase (score) 10)
  (decrease (battery-level) (* 900 (energyconsumed-per-second)))
  (increase (currenttime) 900)
  )
```

### PDDL problem file : maximization of score

```
[...]
(:goal [...] )
(:metric maximize (score))
[...]
```

planning the socialize action

- Second approach
  - We try to minimize total waited time by waiting actions
  - We expect that the planner will introduce socialize actions in resulting plans
  - This approach failed due to another limitation of the optimizer : it does not permit non-constant effect on metric

# PDDL code for time waited approach

### PDDL domain file : waiting action

#### (:action WAIT-PRES-SESSION

```
:parameters(?ses - pres-session)
:precondition(and
  (< (currenttime) (time-pres-begin ?ses))
  (>= (battery-level) (* (- (time-pres-begin ?ses) (currenttime))
(energyconsumed-per-second)))
  )
  :effect(and
  (increase (currenttime) (- (time-pres-begin ?ses) (currenttime)))
     (increase (total-waited-time) (- (time-pres-begin ?ses)
(currenttime)))
  (decrease (battery-level) (* (- (time-pres-begin ?ses) (currenttime)))
```

```
(decrease (battery-level) (* (- (time-pres-begin ?ses) (currenttime))
(energyconsumed-per-second)))
```

### PDDL problem file : minimize total-waited-time

```
[...]
(:goal [...] )
(:metric minimize (total-waited-time))
[...]
```

## Other Problems we meet

- Some problems have no solution with the planner
- When making several presentations and/or posters in different rooms, the computing is very slow. The planner seems to inefficient for our domain
- Enforced hill climbing is not necessary for some cases in which we can not get a right plan as we expect
- Others...

## Another planner: LPG

- A planner based on local search and planning graphs
- Handles PDDL2.1 domains involving numerical quantities and durations
- Choose the best-first algorithm
- See result LPG web page

## **Comparison of LPG and Metric-FF**

### LPG

- Based on local search and planning graphs
- PDDL2.1 involving numerical quantities and durations
- Best-first algorithm
- Faster
- Less robust
- Support optimization
- Do not support "equal"

### **Metric-FF**

- A forward chaining heuristic state space planner
- Deals with PDDL 2.1 level
   2, combined with ADL
- Enforced hill climbing
- Slower
- Robust in producing plans
- Good at optimization
- Support "equal" operator

## Conclusion

- PDDL is too restrictive to model our domain (Scientific Conference)
- Is very difficult to specify optional tasks (or with priority)
- Metric-FF can not handle time very well
- Planner based on Chronicles and Tasks could be more appropriate

### References

- Ghallab, Nau and Traverso. Automated Planning : theory and practive, 2004.
- Fox and Long. PDDL 2.1: An Extension to PDDL for Expressing Temporal Planning Domains. [PS.GZ]
- J. Hoffmann, Extending FF to Numerical State Variables, in: Proceedings of the 15th European Conference on Artificial Intelligence, Lyon, France, July 2002. [PS.GZ]
- Hoffmann. The Metric-FF Planning System: Translating "Ignoring Delete List" to Numeric State Variables, Journal of Artificial Intelligence Research 20 (2003). [<u>PS.GZ</u>]
- Hoffmann and Nebel. The FF Planning System: Fast Plan Generation Through Heuristic Search, Journal of Artificial Intelligence Research 14 (2001). [<u>PS.GZ</u>]
- International Planning Competition 2002 (IPC-3) Web Site : <u>http://planning.cis.strath.ac.uk/competition/</u>
- International Planning Competition 2004 (IPC-4) Web Site : <u>http://www-rcf.usc.edu/~skoenig/icaps/icaps04/index.html</u>