Extended Capabilities of High-Level Planners

Lasse Einig

einig@informatik.uni-hamburg.de

University of Hamburg, Group TAMS

December 1, 2013

RACE

SEVENTH FRAMEWORK PROGRAMME

Table of Contents

1. Introduction
2. Basic Concepts
   - HTN Planner
   - SMACH
3. Parallelizing Plans
   - Analysis
   - Implementation
   - Evaluation
4. Re-planning
5. Conclusion

Outline

1 Introduction

2 Basic Concepts
- HTN Planner
- SMACH

3 Parallelizing Plans
- Analysis
- Implementation
- Evaluation

4 Re-planning

5 Conclusion

What is planning?

Target

Achieve a desired goal state

- Low-level
    - Motion
    - Path
- High-level

High-level Planner

- Classical
  - Forward chaining
  - Backward chaining
- Temporal
- Probabilistic
  - Markov decision process

Robotic High-level planning

Hierachical Task Network (HTN)

Outline

1. Introduction
2. Basic Concepts
   - HTN Planner
   - SMACH
3. Parallelizing Plans
   - Analysis
   - Implementation
   - Evaluation
4. Re-planning
5. Conclusion

HTN planner

- SHOP, SHOP2, NONLIN, SIPE-2, O-PLAN, UMCP
- Recursive task decomposition
    - Task size
- Domain specification
    - Operators
    - Methods
    - Axioms
- Problem

HTN Operators

```
(:operator (!name ?arg1 ?arg2)
  (on ?table ?arg1)
  (on ?table ?arg1)
  ((on ?arg2 ?arg1) (:protection (on ?arg2 ?arg1)))
)
```

A sample operator in a SHOP planning domain

HTN Methods

```
(:method (name ?arg1)
   plane
   (enough-cash ?cash)
   ((!buy-ticket ?plane ?x) (!travel ?arg1))
   train
   (not-enough-cash ?cash)
   ((!buy-ticket ?train ?x) (!travel ?arg1))
)
```

A sample method in a SHOP planning domain

HTN Axioms

```
(:- (travel-by-plane ?x)
  (( ticket-price  ?x ?t) (cash-available  ?c) ( call  >= ?c ?t)
    ( distance  ?x ?d) ( call  >= ?d 50)
    ( airport-available ))
)
```

A sample axiom in a SHOP planning domain

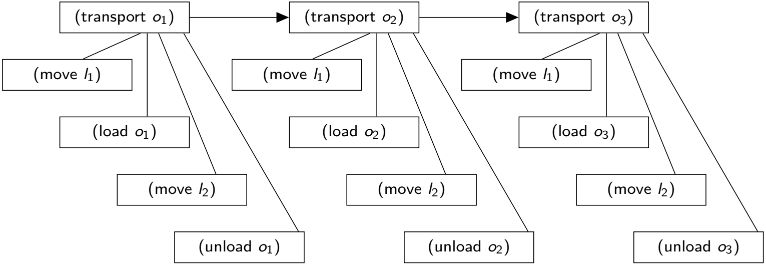
Decomposition

- Structured
- Hierarchical

Decomposition

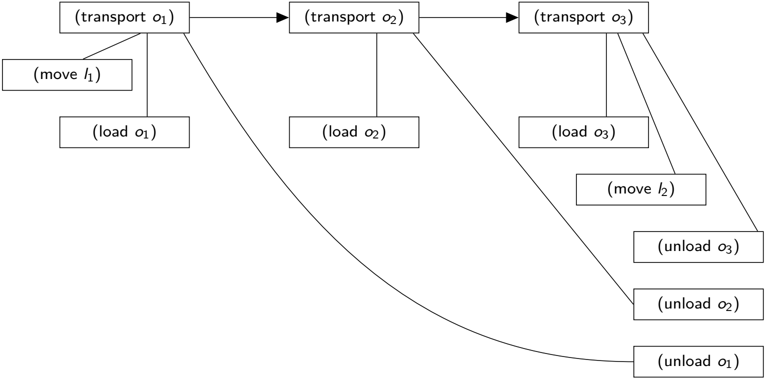
- Structured
- Hierarchical
- SHOP2
  - Partial Order

Partial Ordering



A sample plan for moving three objects generated by SHOP algorithm using total ordering

Partial Ordering



A sample plan using partial ordering

SMACH

- State Machine
  - Native ROS library
  - ROS-independent

SMACH

- State Machine
  - Native ROS library
  - ROS-independent
- States
  - Corresponds to an executable task

SMACH

- State Machine
  - Native ROS library
  - ROS-independent
- States
  - Corresponds to an executable task
- Containers
- Nesting possible

Containers

- State Machine
- Concurrence
- Sequence
- Iterator

Outline

1 Introduction

2 Basic Concepts

- HTN Planner
- SMACH

3 Parallelizing Plans

- Analysis
- Implementation
- Evaluation

4 Re-planning

5 Conclusion

Problem

- Sequential plans
- Resources lie wasted
  - Head
  - Torso
  - Base
  - Right arm
  - Left arm
- Time-inefficient

Operators

- !tuck_arms ?arms
- !move_arm_to_side ?arm
- !move_torso ?position
- !move_base ?to
- !move_base_blind ?to
- !pick_up_object ?object ?arm
- !place_object ?object ?arm ?to

Problem Description

Room contains:

- Robot
- Counter
  - Coffee cup
- Table

Problem Description

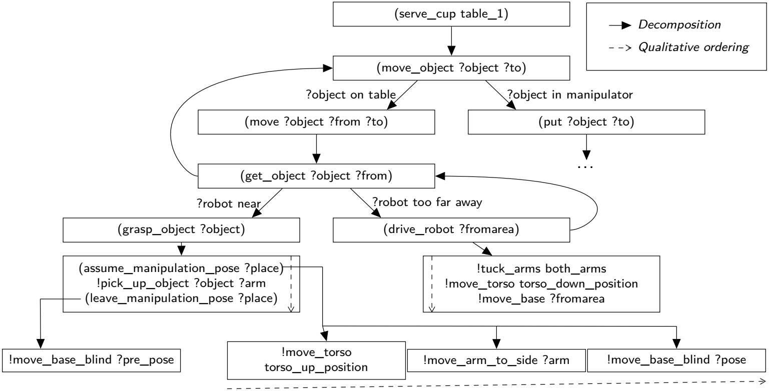
Room contains:

- Robot
- Counter
  - Coffee cup
- Table

Objective

Serve coffee cup

Methods and Decomposition



Domain decomposition for the “Serving Beverages” scenario

Sequential Plan

Analysis

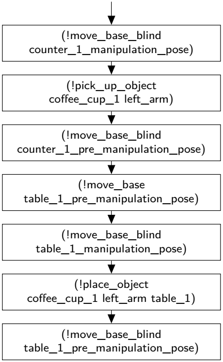
1. *!tuck_arms both_arms*
2. *!move_torso torso_down_position*
3. *!move_base counter_1_pre_manipulation_pose*
4. *!move_torso torso_up_position*
5. *!move_arm_to_side left_arm*
6. *!move_base_blind counter_1_manipulation_pose*
7. *!pick_up_object coffee_cup_1 left_arm*
8. *!move_base_blind counter_1_pre_manipulation_pose*
9. *!move_base table_1_pre_manipulation_pose*
10. *!move_base_blind table_1_manipulation_pose*
11. *!place_object coffee_cup_1 left_arm table_1*
12. *!move_base_blind table_1_pre_manipulation_pose*

Sequential Plan

Analysis

1	*!tuck_arms both_arms*	[RA, LA]
2	*!move_torso torso_down_position*	[T]
3	*!move_base counter_1_pre_manipulation_pose*	[B, RA, LA, T]
4	*!move_torso torso_up_position*	[T]
5	*!move_arm_to_side left_arm*	[LA]
6	*!move_base_blind counter_1_manipulation_pose*	[ALL]
7	*!pick_up_object coffee_cup_1 left_arm*	[LA, H]
8	*!move_base_blind counter_1_pre_manipulation_pose*	[ALL]
9	*!move_base table_1_pre_manipulation_pose*	[B, RA, LA, T]
10	*!move_base_blind table_1_manipulation_pose*	[ALL]
11	*!place_object coffee_cup_1 left_arm table_1*	[LA, H]
12	*!move_base_blind table_1_pre_manipulation_pose*	[ALL]

Parallel Plan – End



(!move_base_blind counter_1_manipulation_pose)

(!pick_up_object coffee_cup_1 left_arm)

(!move_base_blind counter_1_pre_manipulation_pose)

(!move_base table_1_pre_manipulation_pose)

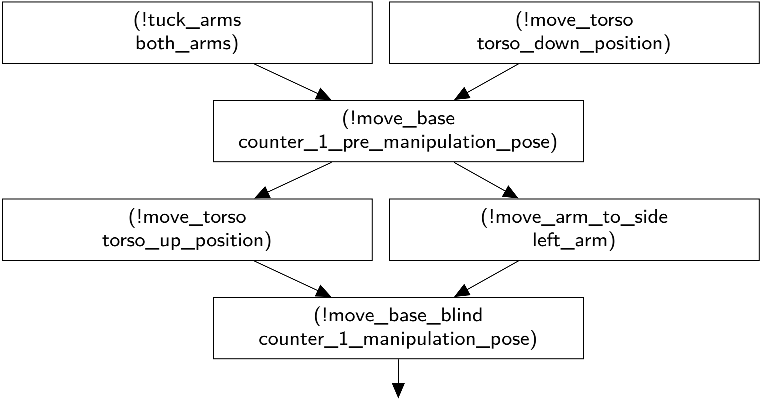
(!move_base_blind table_1_manipulation_pose)

(!place_object coffee_cup_1 left_arm table_1)

(!move_base_blind table_1_pre_manipulation_pose)

Sequential section of the *Serving Beverages* scenario

Parallel Plan – Defensive

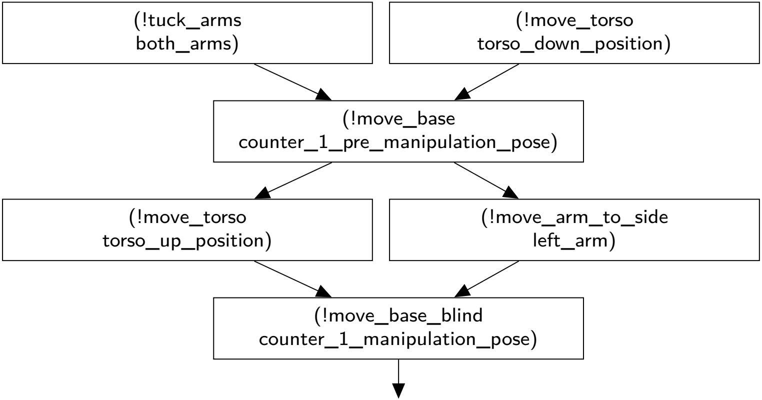

(!tuck_arms both_arms)
(!move_torso torso_down_position)
(!move_base counter_1_pre_manipulation_pose)
(!move_torso torso_up_position)
(!move_arm_to_side left_arm)
(!move_base_blind counter_1_manipulation_pose)

Parallel section of the *Serving Beverages* scenario with security constraint

Implementation

- Linking actions by resources
- Find start actions
- Remove start actions from graph
- Assign actions to *PList* or *SList*
- Loop until graph is empty

Parallel Plan – Defensive



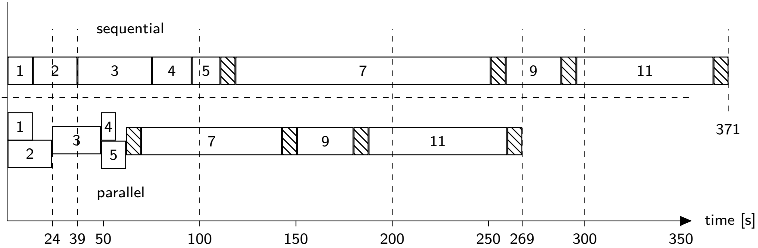
Parallel section of the *Serving Beverages* scenario with security constraint

Soundness

```
(
    [
      —!tuck_arms both_arms—
      —!move_torso torso_down_position—
    ]
  —!move_base counter_1_pre_manipulation_pose—
    (
        [
          —!move_torso torso_up_position—
          —!move_arm_to_side left_arm—
        ]
      —!move_base_blind counter_1_manipulation_pose—
      —!pick_up_object coffee_cup_1 left_arm—
      —!move_base_blind counter_1_pre_manipulation_pose—
      —!move_base table_1_pre_manipulation_pose—
      —!move_base_blind table_1_manipulation_pose—
      —!place_object coffee_cup_1 left_arm table_1—
      —!move_base_blind table_1_pre_manipulation_pose—
    )
)
```

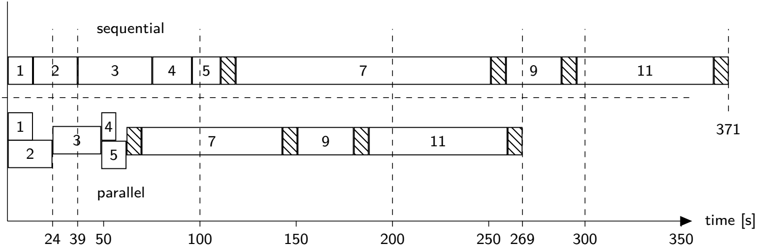
Plan resulting from the parallelization algorithm optimizing the Serving Beverages scenario using security constraints

Practical Experiment - Time



Result from the practical experiment of the Serving Beverages scenario

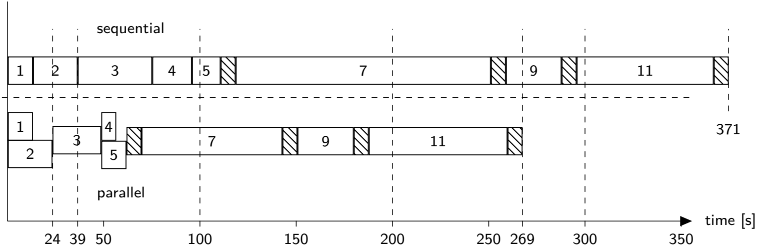
Practical Experiment - Time



Result from the practical experiment of the Serving Beverages scenario

- 102 seconds $\widehat{=}$ 27.5 %

Practical Experiment - Time



Result from the practical experiment of the Serving Beverages scenario

- 102 seconds $\widehat{=}$ 27.5 %
- 15 seconds $\widehat{=}$ 39.5 %

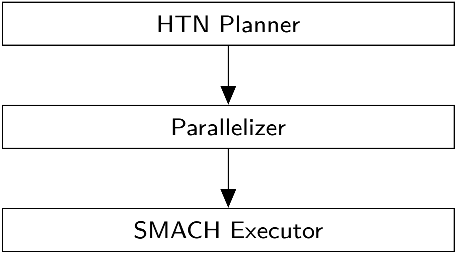
Outline

1 Introduction
2 Basic Concepts
- HTN Planner
- SMACH

3 Parallelizing Plans
- Analysis
- Implementation
- Evaluation

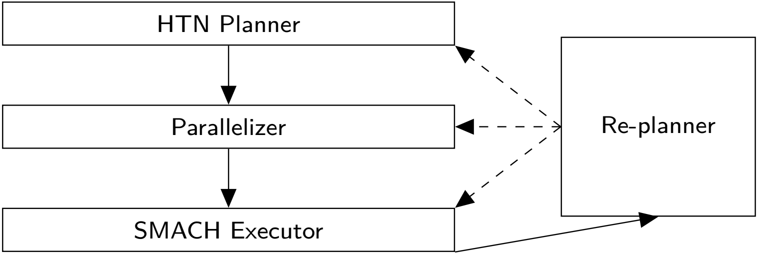
4 Re-planning
5 Conclusion

Closed-loop Re-planner



3-Layer parallelization architecture

Closed-loop Re-planner



Closed-loop Re-planning architecture

Cost-success decision

	Cost	Expected Success Rate	Cost-Success
Recognition			
Head	1	1	1
Torso	2	3	$.6\overline{6}$
Base	4	4	1
Grasping			
Head	1	.1	10
Torso	2	2	1
Base	4	8	.5

Cost based vs. cost-success based adaption

Outline

1. Introduction
2. Basic Concepts
   - HTN Planner
   - SMACH
3. Parallelizing Plans
   - Analysis
   - Implementation
   - Evaluation
4. Re-planning
5. Conclusion

Conclusion

- Introduction to High-level planning
- Challenges
- 3-layered extension for parallel execution
- Closed loop extension for re-planning

References I

J. Bohren and S. Cousins.
The SMACH high-level executive [ROS news].
Robotics Automation Magazine, IEEE, 17(4):18 – 20, dec. 2010.

Luis Castillo, Juan Fdez-Olivares, Óscar García-Pérez, and Francisco Palao.
Temporal enhancements of an HTN planner.
In *Proceedings of the 11th Spanish association conference on Current Topics in Artificial Intelligence*, CAEPIA'05, pages 429 – 438, Berlin, Heidelberg, 2006. Springer-Verlag.

Sam Ade Jacobs, Kasra Manavi, Juan Burgos, Jory Denny, Shawna Thomas, and Nancy M. Amato.
A scalable method for parallelizing sampling-based motion planning algorithms.
In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2529 – 2536, may 2012.

Denis Klimentjew, Sebastian Rockel, and Jianwei Zhang.
Active scene analysis based on multi-sensor fusion and mixed reality on mobile systems.
In Fuchun Sun, Dewen Hu, and Huaping Liu, editors, *Foundations and Practical Applications of Cognitive Systems and Information Processing*, volume 215 of *Advances in Intelligent Systems and Computing*, pages 795–809. Springer Berlin Heidelberg, 2014.

A. R. Lingard and E. B. Richards.
Planning parallel actions.
Artif. Intell., 99(2):261 – 324, March 1998.

References II

J.Y.S. Luh and C.S. Lin.
Scheduling parallel operations in automation for minimum execution time based on pert.
Computers & Industrial Engineering, 9(2):149 – 164, 1985.

D. Nau, H. Munoz-Avila, Y. Cao, A. Lotem, and S. Mitchell.
Total-order planning with partially ordered subtasks.
In *Proceedings of the 17th international joint conference on Artificial intelligence*, volume 17, pages 425–430, Seattle, WA, 2001.

Dana Nau, Yue Cao, Amnon Lotem, and Hector Munoz-Avila.
Shop: simple hierarchical ordered planner.
In *Proceedings of the 16th international joint conference on Artificial intelligence - Volume 2*, IJCAI'99, pages 968–973, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

Dana Nau, Okhtay Ilghami, Ugur Kuter, J. William Murdock, Dan Wu, and Fusun Yaman.
SHOP2: An HTN planning system.
Journal of Artificial Intelligence Research, 20:379–404, 2003.