2.3 'Mission planning for reconfigurable multi-robot systems' (NP-T-03)

Thomas M. Roehr[(1)]

(1) Robotics Innovation Center, DFKI GmbH, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: thomas.roehr@dfki.de

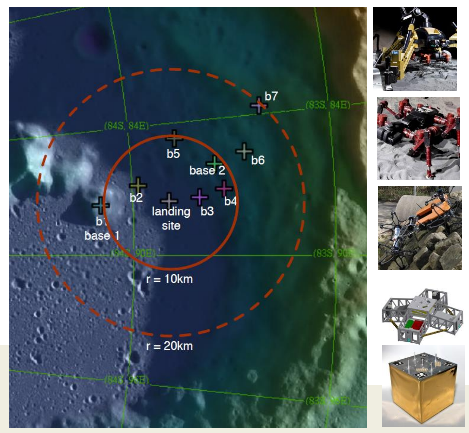
Abstract

This talk present an approach to mission planning for reconfigurable multi-robot systems. It briefly introduces some formal background to the topic, and illustrates the current work-in-progress for developing a temporal mission planning system. The planning system operates on an OWL-based organization model in order to fully exploit reconfigurability. The planner implementation relies on a large collection of state of the art technologies and combines them in a novel way to solve the problem at hand.

Mission planning for reconfigurable multi-robot systems

by Thomas M. Roehr

Project Day 17.9.2015
Workgroup Navigation & Planning



Universität Bremen

Do what you can, with what you have, where you are.

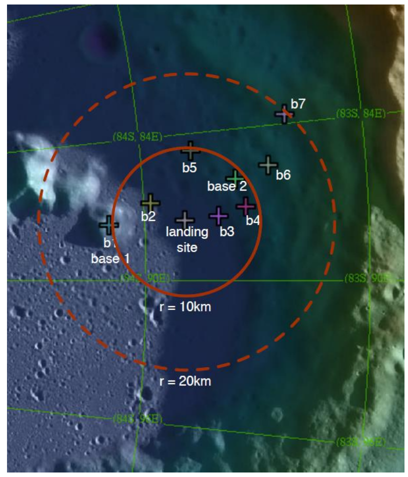
Theodore Roosevelt

Universität Bremen

2

Mission

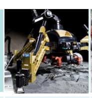
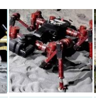
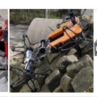
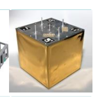
- autonomous multi-robot exploration of the lunar surface driven by science targets
- Example science targets
  - take samples at location b_3,b_4,b_5
  - take pictures from location b_3,b_4,b_5
  - map area around *landing site*
  - place infrastructure elements/sensor equipment at *base 1* and *2*



Universität Bremen

3

The set of available resources

Robots Capabilities					
Locomotion	✓	✓	✓		
Manipulation	✓		✓		
Imaging	✓	✓			✓
Power	✓	✓	✓	✓	✓
Mapping	✓	✓	✓		
…					
Count *(Example Scenario)*	*1*	*1*	*1*	*3*	*10*

Universität Bremen

4

A reconfigurable multi-robot system

Definition 1.1

A physical robotic system represents an **atomic actor** $a \in A$, when it cannot be separated into two or more robotic systems

Definition 1.2

A physical coalition of two or more atomic actors is a **composite actor** CA, i.e.

$$CA = \{a_i, \dots, a_j\}, where\ a_i, \dots, a_j \in A, |CA| = 1$$

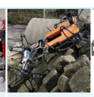
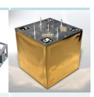
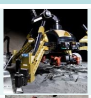
Definition 1.3

Atomic and composite actors are single minded, individual robotic actors.

Definition 1.4

A **reconfigurable multi-robot system** $RMRS$ is a set of fully cooperative atomic actors that can temporarily from composite actors

A reconfigurable multi-robot system

Robot + Robot					
	✓	✓	✓	✓	✓
	✓				✓
	✓			✓	✓
	✓		✓		✓
	✓	✓	✓	✓	✓

|CA| <= |A|

possible combinations $\leq 2^{|A|}$

The number of available and compatible electro-mechanical interfaces limits the possible combinations, but, e.g., finding an optimal coalition is $O(2^N)$

Modelling the robotic system

- Organization model to represent
  - atomic actor capabilities and services
  - reconfigurability with other robots
  - Inference of composite actor capabilities and services
- Implementation using Description Logic (DL) related Web Ontology Language (OWL)
  - qualified cardinality constraints

$$PayloadItem \sqsubseteq Actor \sqsubseteq Resource \sqsubseteq T$$
$$EmiActive \sqsubseteq Interface \sqsubseteq Resource \sqsubseteq T$$
$$EmiActive \sqcap EmiPassive \equiv \bot$$
$$PayloadItem \equiv Actor \sqcap 1has.EmiActive \sqcap 1has.EmiPassive$$
$$EmiPassive \sqsubseteq Interface \sqsubseteq Resource \sqsubseteq T$$

Universität Bremen 7

Defining the planning problem

- Mission description
  - $M = (A_a, STR, C)$, where
    - A_a is the set of available atomic actors
    - STR is the set of spatio-temporally qualified requirements
    - C is the set of (temporal) constraints
  - $r \in STR$ is a spatio-temporally qualified expression ($steq$) of the form $(S, A_r)@[l, t_s, t_e]$, where
    - S is the set of required services
    - A_r is the set of required atomic actors
    - l is a location variable
    - t_s and t_e are temporal variables, such that $t_s < t_e$

Universität Bremen 8

Defining the planning problem

- Mission description example:
  - Constants:
    - locations $L = \{lander, b_1, \ldots, b_n\}$; timepoints $T = \{t_0, \ldots, t_n\}$
  - $(S, A_r)@[l, t_s, t_e] \rightarrow$
    $(\{ImageLocationProvider, EmiPowerProvider\}, \{\})@[lander, t_0, t_1]$


```
  - <location>
      <id>b7</id>
      <radius>moon</radius>
      <latitude>-83.34083</latitude>
      <longitude>84.64467</longitude>
    </location>
  </constants>
- <requirements>
  - <requirement>
    - <spatial-requirement>
          <!-- where it is required -->
      - <location>
          <id>lander</id>
        </location>
      </spatial-requirement>
        <!-- when it is required / mixing qualitative and quantitative information -->
    - <temporal-requirement type="persistence-condition | event">
        <from>t0</from>
        <to>t1</to>
      </temporal-requirement>
        <!-- what is required at this very position -->
    - <service-requirement>
        <service>http://www.rock-robotics.org/2014/01/om-schema#LocationImageProvider</service>
        <service>http://www.rock-robotics.org/2014/01/om-schema#EmiPowerProvider</service>
      </service-requirement>
    </requirement>
  - <requirement>
    - <spatial-requirement>
          <!-- where it is required -->
      - <location>
          <id>b1</id>
```


Universität Bremen

9

Solving the planning problem

- Planning algorithm:
  (1) *typing*
  assign actor types that fulfill the (service and resource) requirements for each spatio-temporal tuple
  (2) *role assignment*
  pick a solution from (1) and instantiate, i.e. assign roles to each atomic actor type
  (3) *timeline construction*
  for each role create a timeline
  (4) *time-expanded network construction*
  create a transport network from system movements
  (5) *flow optimization for immobile systems (e.g. payload)*
  compute flow/"transport lines"
  (6) *assign roles to robots*

Universität Bremen

10

Typing

(1) Example: typing for
$(\{ImageLocationProvider, EmiPowerProvider\}, \{\})@[lander, t_0, t_1]$

feasible atomic actor types *feasible composite actor types*

Robots / Capabilities						
Locomotion	✓	✓	✓			✓
Imaging	✓	✓			✓	✓
Power	✓	✓	✓	✓	✓	✓
ImageLocation-Provider	✓	✓				✓
EmiPowerProvider	✓	✓	✓	✓	✓	✓

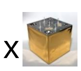
. . .

Universität Bremen 11

Role assignment

(2) Example role assignment

- composite actor that fulfills requirements:
  - Coyote III + (Camera)-Payload

- 1 x  available → role: $coyote_0$
- 10 x  available → role: $payload_0, \ldots, payload_9$

This is not a direct assignment to a particular system and allows:

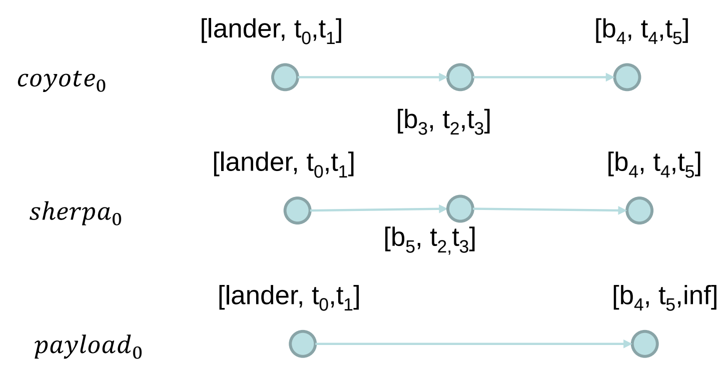
- optimization of particular actor assignment
- timeline construction

Universität Bremen 12

Timeline construction

(3) Example timeline construction

- time-ordered (temporally qualified) path of a system's locations



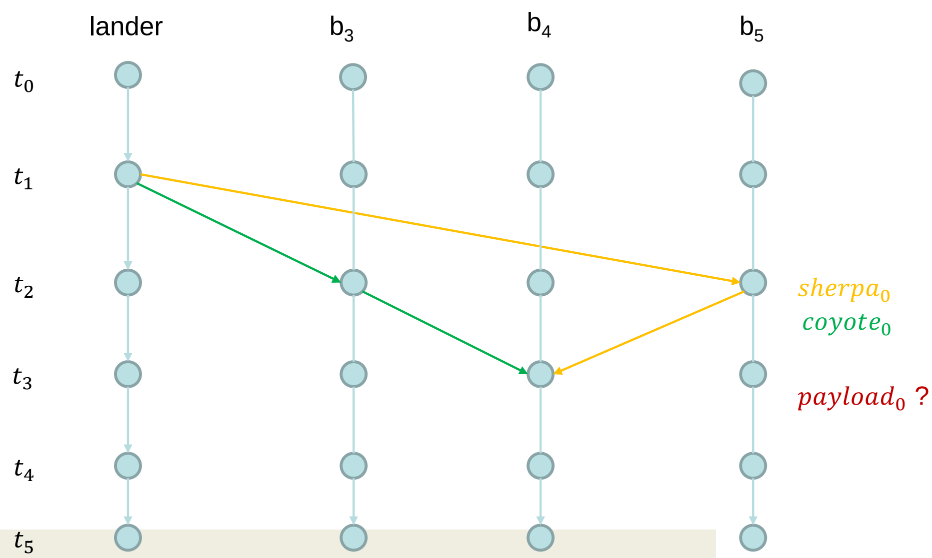
$$t_0 < t_1 < t_2 < t_3 < t_4 < t_5$$

Universität Bremen

13

Time-expanded network construction

(4) construction for mobile systems

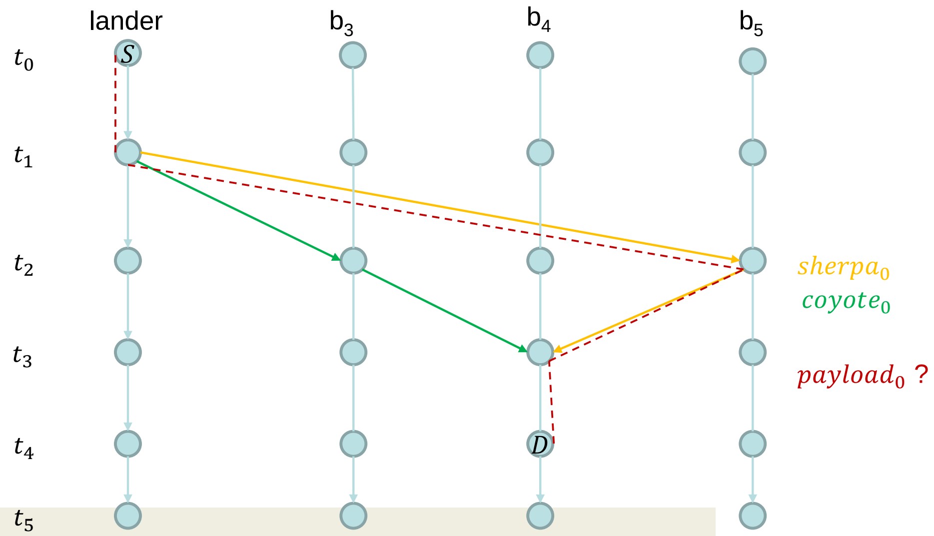


Universität Bremen

14

Flow optimization

(5) For payload transport assign supply S to origin location, demand D to destination and solve for min-cost flow



Universität Bremen

15

Final assignment

(6) Find a good assignment to actual robots

- What is good?
  - efficient: minimum energy cost
  - safe: keep a high or given level of redundancy to guarantee mission success

Universität Bremen

16

Conclusion

- Looks like mission planning for reconfigurable multi-robot systems is doable
  - current implementation yet lacks the construction of the time expanded network
  - flow optimization has been implemented as linear program, thus in this form likely not scalable

- Technologies involved:
  - knowledge-based reasoning (OWL) (using my C++ implementation of owlapi)
  - constraint-based problem solving (using Gecode)
  - flow optimization, linear programming (using GLPK)
  - temporal constraint satisfaction (using my C++ implementation)
  - graphs and graph algorithms (integration using my graph_analysis library – a wrapper for lemon, SNAP, and boost)

Universität Bremen

17