From Abstract Crisis to Concrete Relief

A Preliminary Report on Combining State Abstraction and HTN Planning

Susanne Biundo and Bernd Schattenberg
Overview

• Motivation
• Basic Concepts
• Combining HTN and State-based Planning
• Implementation
• Conclusions
Motivation

- Applications: e.g. assisting crisis management
  - THW relief mission in flooding disaster
- Domains with rich and widespread tasks
  - Logistics
  - Construction
  - Workflow
  - ...
Motivation (2)

• Properties of the applications:
  – Complex domain models
  – Incomplete information

• Requirements:
  – Adequate modelling language
  – Flexible integration of new tasks
  – “Efficient” plan generation process

→ Hybrid
Basic Concepts

- **Hierarchical Task Network Planning**
  - Abstract tasks as makro operators
  - Primitive tasks represent singular actions
  - Methods decompose abstract tasks stepwise into networks of primitive tasks

- **Integrated State-based Planning**
  - All tasks carry preconditions and effects
  - Identifying condition establishers, inserting new tasks if necessary
  - Detecting and resolving causal conflicts
Combining HTN and State-based Planning

- Relation between abstract and primitive tasks:
  - Methods

method m_{t1}
expands transport (?passengers, ?from, ?to, ?by)
vars ?road Road
nodes (1:board (?passengers, ?from, ?by))
  (2:driving (?by, ?from, ?to, ?road))
  (3:un-board (?passengers, ?from, ?by))
... 
order 1<2, 2<3,...
causal 1—in(?passengers, ?by)—2
binding ...
Combining HTN and State-based Planning (2)

- Relation between abstract and primitive preconditions and effects of the tasks:
  - Sort hierarchy
  - Decomposition axioms

\[
\text{At(Unit } u, \text{Location } l) \iff \\
\text{Standing-at(Vehicle } u, \text{Location } l, \text{Road } r) \lor \\
\text{Aircraft-at(Aircraft } u, \text{Location } l, \text{Height } h) \lor \\
[\text{At(Container } c, \text{Location } l) \land \\
\text{Contains(Container } c, \text{Unit } u)] \lor \ldots
\]
Closing Open Preconditions

**driving**

(Vehicle ?u, Location ?from, Location area4, Road ?r)

P: Standing-at(?u,?from,?r),
Reachable-by-land(?from,?r),
Status(?r,ok).
E: +Standing-at(?u,area4,?r),
–Standing-at(?u,?from,?r).

**logistics-and-supplies**

... evacuating

**informing-population**

securing-population ...

**medical-treatment**

build-support-camp

**establish-camp**

(Passengers group1, Location area4, Location camp2, Unit ?u)

P: At(?u,area4),
At(group1,area4).
E: +At(?u,camp2),
+At(group1,camp2),
–At(?u, area4),
–At(group1, area4).

**move**

(Unit ?u, Location ?from, Location area4)

P: At(?u,?from).
E: +At(?u,area4),
–At(?u,?from).

**fly**

(Aircraft ?u, Location ?from, Location area4, Tower ?t)

P: Aircraft-at(?u,?from), Handled-by(?u,?t),
Clearance(?t,ok), Maintenance-status(?u,ok).
E: +Aircraft-at(?u,area4),–Aircraft-at(?u,?from),
+Maintenance-status(?u,ko),–Maintenance-status(?u,ok).

**sort-hierarchy**

development axioms

classical
Detecting and Resolving Conflicts

move ... 

logistics-and-supplies ... evacuating

informing-population securing-population ...

medical-treatment build-support-camp

board(group1,area4,u)

un-board(group1,area4,u)

move 

(Unit Jeep1, Location ?l, Location depot)
P: At(Jeep1,?l).
E: +At(Jeep1,depot), –At(Jeep1,?l).

move

(UNIT u, Location ?from, Location area4)
P: At(?u,?from).
E: +At(?u,area4), –At(?u,?from).

At(?u,area4)

(Vehicle ?u, Location ?from, Location area4, Road ?r)
P: Standing-at(?u,?from,?r), Reachable-by-land(?from,area4,?r), Status(?r,ok).
E: +Standing-at(?u,area4,?r), –Standing-at(?u,?from,?r).
Detecting and Resolving Conflicts

**move**

(Unit Jeep1, Location ?l, Location depot)
P: \( \text{At}(\text{Jeep1}, ?l) \).
E: +\( \text{At}(\text{Jeep1}, \text{depot}) \),
   −\( \text{At}(\text{Jeep1}, ?l) \).

**move**

(Unit ?u, Location ?from, Location area4)
P: \( \text{At}(?u, ?from) \).
E: +\( \text{At}(?u, \text{area4}) \),
   −\( \text{At}(?u, ?from) \).

**logistics-and-supplies**

... evacuating

**medical-treatment**

build-support-camp

**informing-population**

securing-population ...

**board**

(group1, area4, ?u)

**un-board**

(group1, area4, ?u)

**driving**

(Vehicle ?u, Location ?from, Location area4, Road ?r)
P: Standing-at(?u, ?from, ?r),
   Reachable-by-land(?from, area4, ?r),
   Status(?r, ok).
E: +Standing-at(?u, area4, ?r),
   −Standing-at(?u, ?from, ?r).

**Standing-at(?u, area4, ?r)**
Combining HTN and State-based Planning (3)

- Expansion of abstract tasks
- Closing open preconditions, inserting new tasks if necessary
- Threat handling between any levels of abstraction
  - Promotion, demotion, variable bindings
  - Expansion (“threat splitting” - overlapping)
- Hybrid approach allows for flexible planning strategies
  - From pure HTN to pure POCL
Implementation

- First prototype in Java
- Simple hybrid planning algorithm
  - Recursive task definitions useful (termination!)
- Core module of integrated architecture
Conclusions

• Hybrid planning approach
  – Flexibly integrating HTN and POCL concepts
  – Powerful modelling language
    • ...that is “easy” to use and to read
  – Semantics based on many-sorted FOL
    • Decomposition axioms defining legal expansions

• Future work:
  – Experiments with strategies and their validation
  – Resources