From Abstract Crisis to Conrete Relief

A Preliminary Report on Combining State Abstraction and HTN Planning

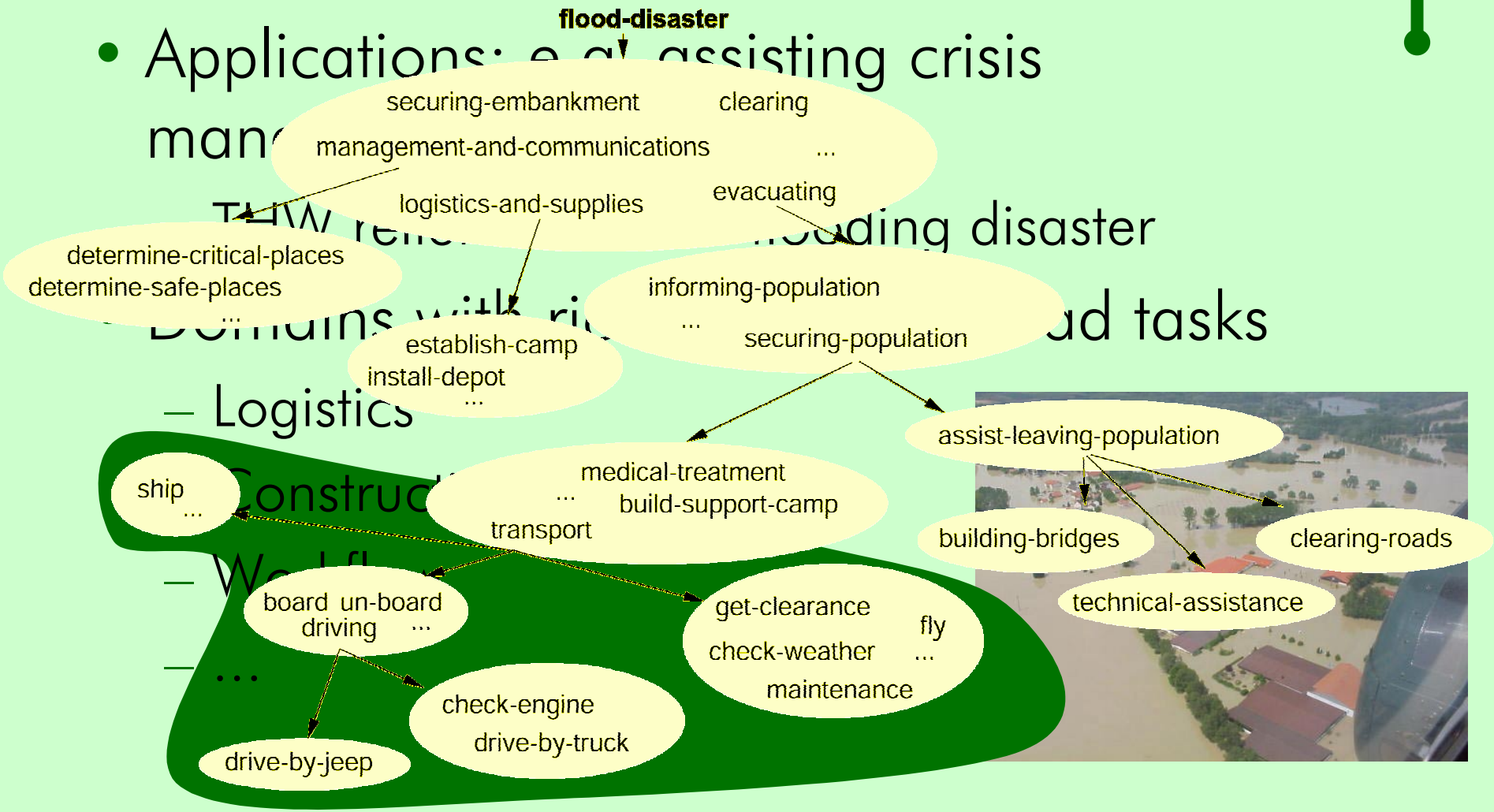
Susanne Biundo and Bernd Schattenberg

Overview

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

Motivation

- Applications: e.g. assisting crisis man
  - THW rele… …loading disaster
  - Domains with ri… …ad tasks
    - Logistics
    - Construc…
    - Workfl…
    - …



Motivation (2)

- Properties of the applications:
  - Complex domain models
  - Incomplete information
- Requirements:
  - Adequate modelling language
  - Flexible integration of new tasks
  - “Efficient” plan generation process

HTN

POCL

→ 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

At(Unit *u*,Location *l*) ⇔
Standing-at(Vehicle *u*, Location *l*, Road *r*)∨
Aircraft-at(Aircraft *u*,Location *l*, Height *h*)∨
[At(Container *c*,Location *l*)∧
Contains(Container *c*,Unit *u*)] ∨ ...

Closing Open Preconditions

logistics-and-supplies
... evacuating

informing-population
securing-population ...

medical-treatment
build-support-camp

driving

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

decomposition axioms

nsport

(Passengers group1, Location area4,
Location camp2, Unit *?u*)
P: At(*?u*,area4),
At(group1,area4).
E: +At(*?u*,camp2),
+At(group1,ca
–At(*?u*, area4),
–At(group1, area4).

establish-camp

(Location area4, Group thw26,
Jeep jeep18, Truck truck9, ...)
P: Status(area4,cleared), ...
E: +At(truck9,area4),
+At(jeep1,area4),
+Status(area4,occup),
–Status(area4,cleared)...

sort-hierarchy

move

(Unit *?u*, Location *?fro*
Location area4)
P: At(*?u*,*?from*).
E: +At(*?u*,area4),
–At(*?u*,*?from*).

classical

decomposition axioms

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).

Detecting and Resolving Conflicts

logistics-and-supplies
... evacuating

informing-population
securing-population ...

... **move**

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

medical-treatment
...
transport build-support-camp

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*).

At(*?u*,area4)

move

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

Detecting and Resolving Conflicts

logistics-and-supplies
... evacuating

informing-population
securing-population ...

... **move**

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

medical-treatment
...
transport build-support-camp

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*).

move

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

At(*?u*,area4)

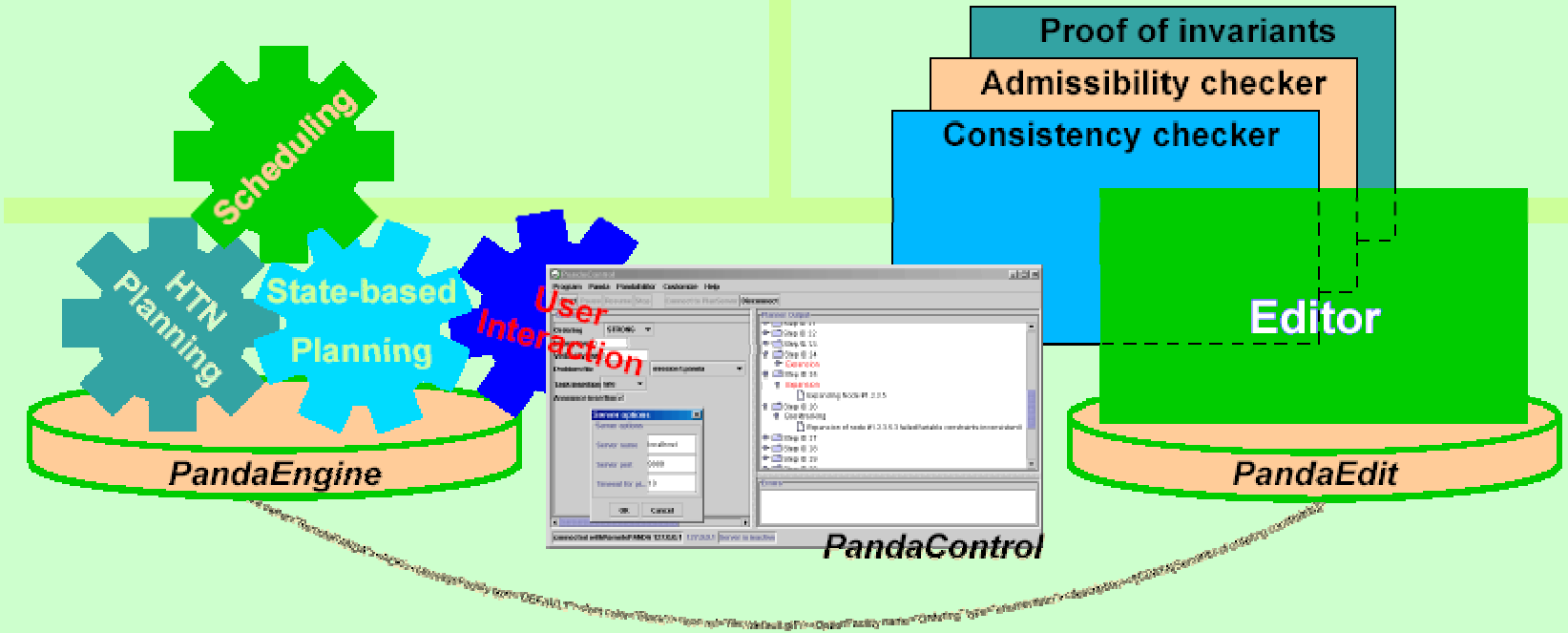
Standing-at(*?u*,area4,*?r*)

Combining HTN and State-based Planning (3)

- Expansion of abstract tasks
- Closing open preconditions, inserting new tasks if neccessary
- 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