Mission planning

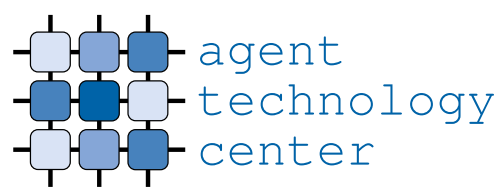
Logic and game theory in multi-robot applications

Peter Novák

Agent Technology Center

Czech Technical University in Prague

March 10th 2011



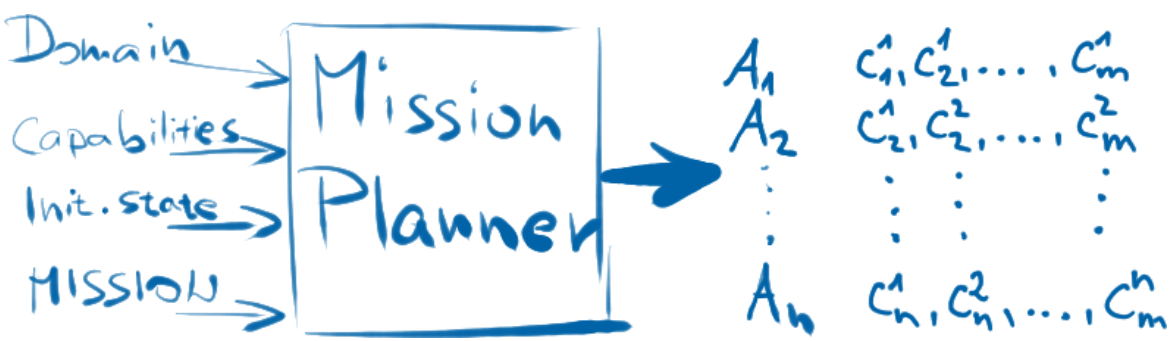
Motivation

Teamwork is a cooperative effort by the members of a team to achieve a common goal, in which individual team members contribute different skills, knowledge and interests to the unity and efficiency of the group.

Multi-agent planning

MASPlan: $\Pi = (\mathcal{L}, K_{dom}, \{(C, \phi)\}_{i=1}^{n}, S_o, \gamma)$

- $\mathcal{L}$: a language of domain and control knowledge
- K_{dom}: domain and control knowledge about synchronised actions
- $\{(C, \phi)\}$: sets of capabilities of agents $A_1, \ldots, A_n$ annotated by formulae from $\mathcal{L}$
- S_0: the initial state
- γ: a goal



Temporally extended goals and capabilities

starting assumptions

$\phi \in \mathcal{L}_{\text{LTL}}$: temporally extended annotations of the agents' capabilities

$\gamma \in \mathcal{L}_{?\text{TL}}$: temporally extended goal (mission specification)

Is this reasonable?

1. modularity of behaviours $\rightsquigarrow$ different approaches are appropriate for solving different problems
2. mission $\rightsquigarrow$ constraints on the plan, i.e., what should/shouldn't happen *during* execution of the mission

Bank robbers example revisited

Mission:

$[?]\Diamond checkedin(money) \wedge$
$\Box safe\_team \wedge$
$\Diamond call\_boss$

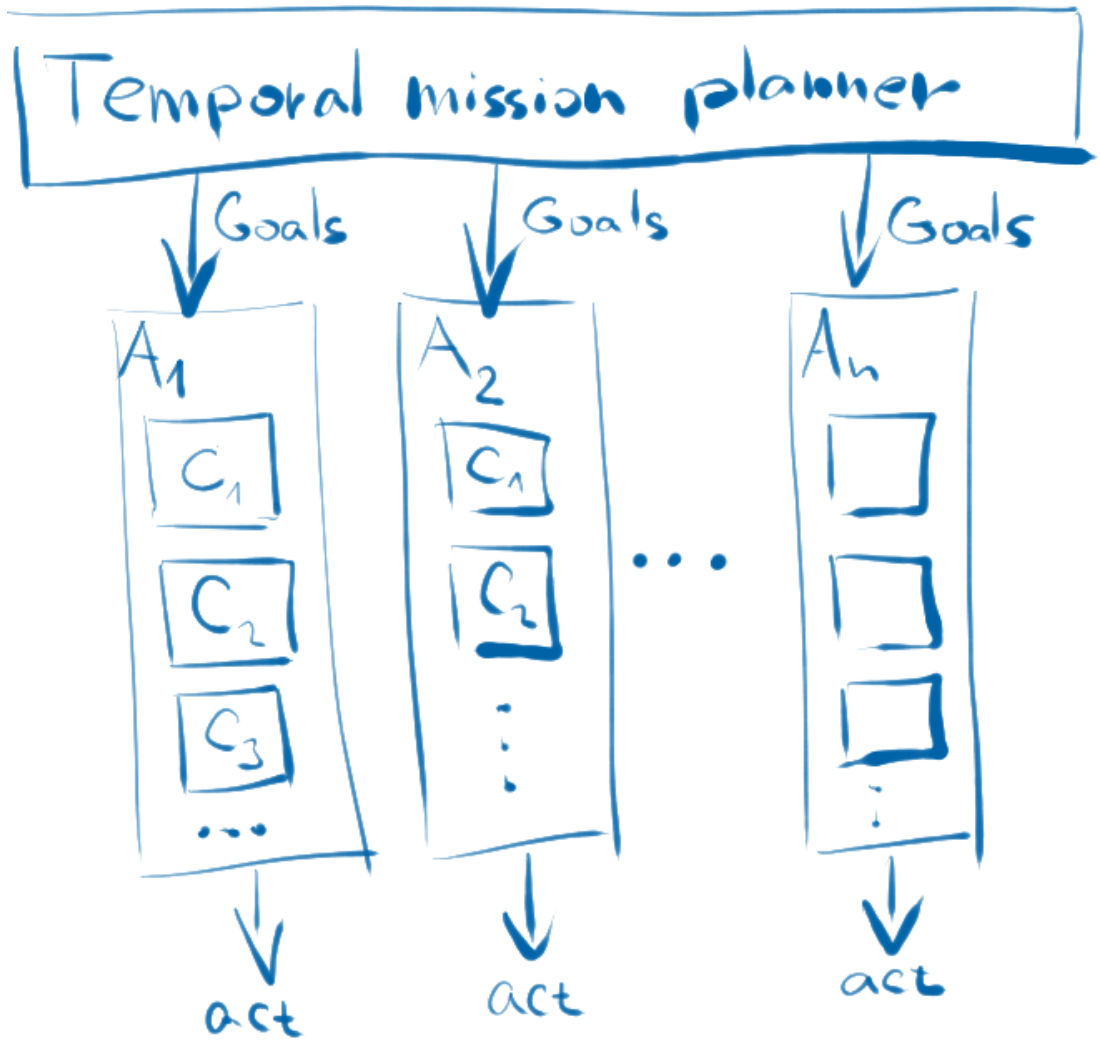
Capabilities (DCTL*/Process Logic)

lockpick: $[P_l](pos(near\_tresor) \rightarrow \Diamond open(tresor)) \wedge \Diamond swear$
$[P'_l] \bigcirc swear \wedge \Diamond call\_boss$

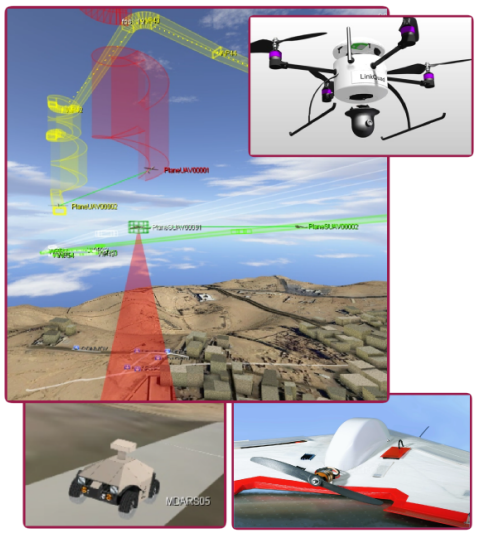
guard: $[P_g]\Box(safe\_team\,\mathcal{U}\,(called\_off \vee danger))$
$[P'_g]\Box(danger \rightarrow fight\,\mathcal{U}\,(dead \vee safe\_team))$

carrier: $[P_c]has(I) \wedge team\_ready \rightarrow$
$\Diamond(pos(airport) \wedge checkedin(I))$

Modular architecture sketch



Urban tactical missions
captive extraction/rescue operations



1. exploration
2. advance to the target location
3. extraction
4. retreat

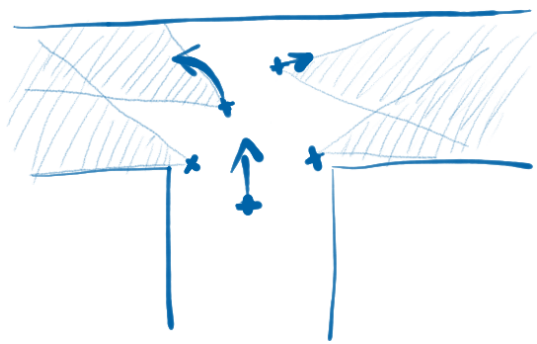
Problems and challenges

Adversarial planning:

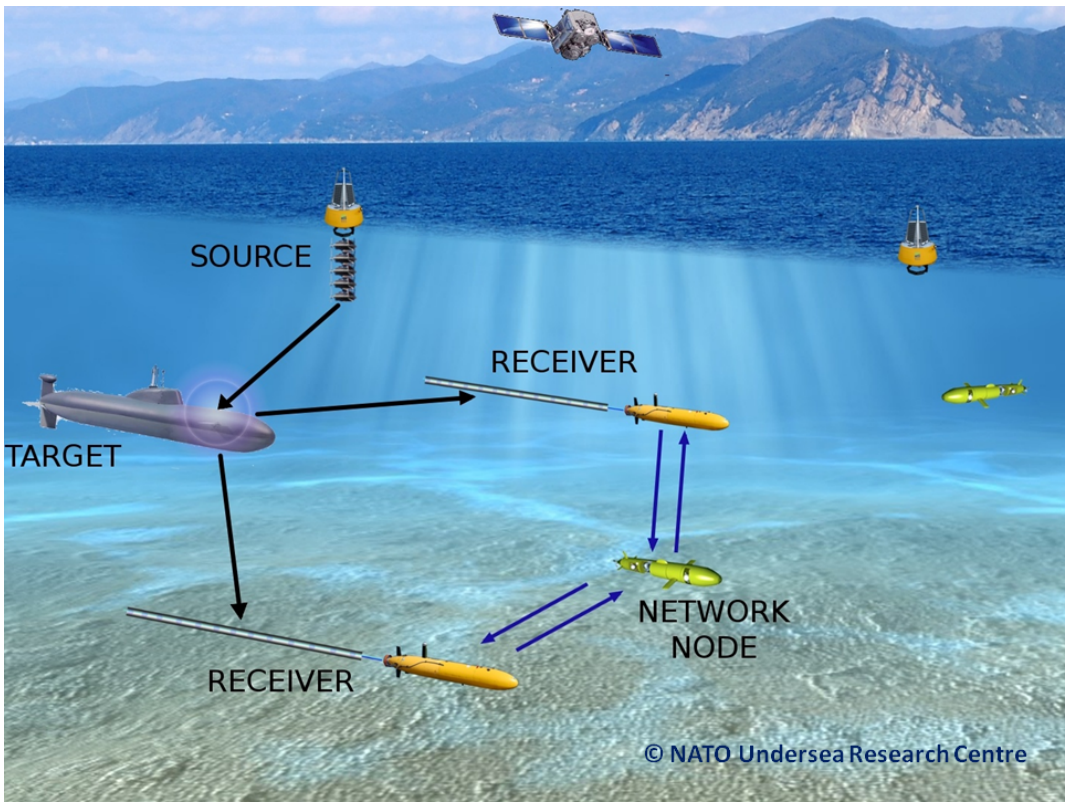
- *smart targets modelling*
  - pursuit-evasion (large map)
- *patrolling*
  - static targets (houses)
  - mobile targets (convoys)

Lower level planning problems:

- *coordinated crossing intersections*
- *dealing with unexpected events*
  - re-planning, plan repair, reactive planning, agent programming



Undersea domain
Anti-submarine warfare/port guarding



(courtesy of NATO Undersea Research Centre)

communication
extremely constrained

- high autonomy
- decentralised com-
  putation/planning
- optimise for low
  communication
  complexity

Timeliness and prior art

- hardware ⇝ routinely deployed & relatively cheap
- tele-operation prevalent ⇝ 1 asset vs. 1 operator

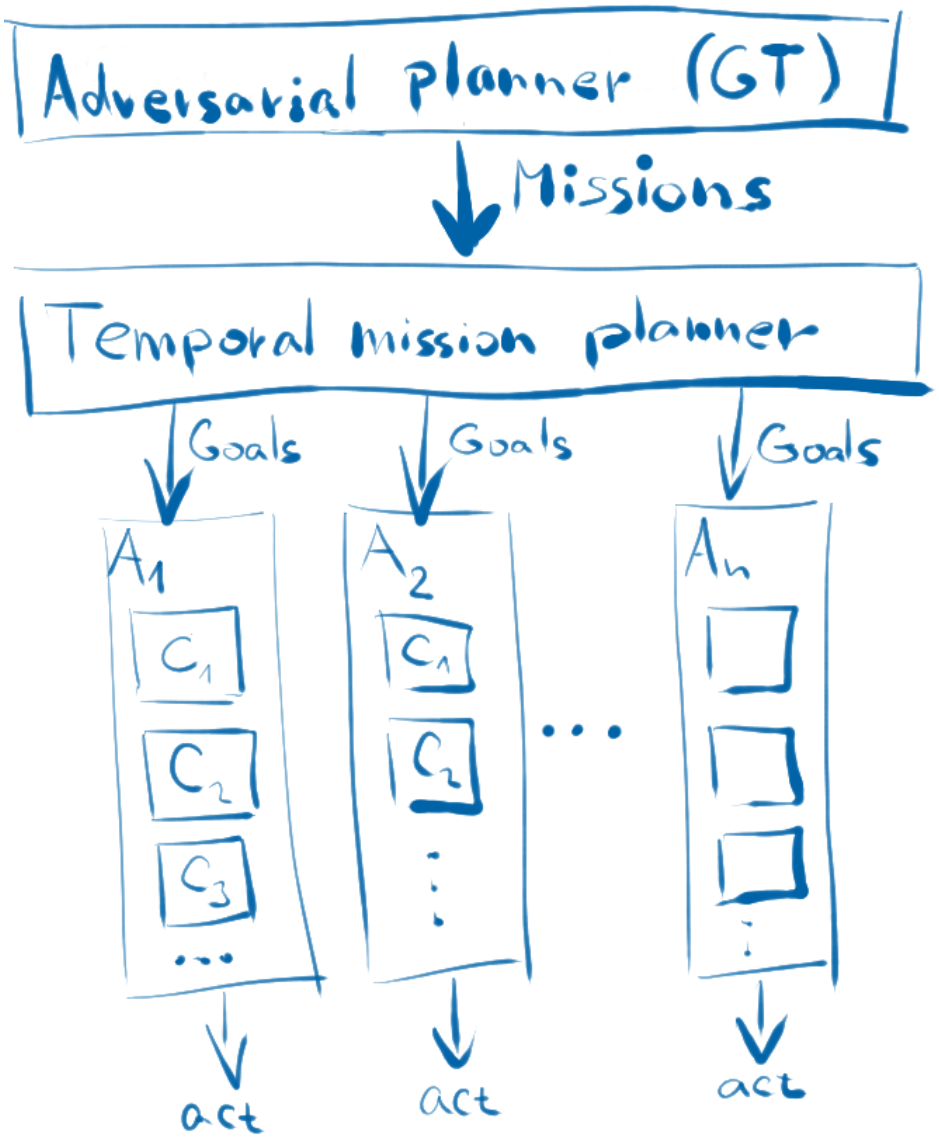
  ⇝ high costs!

- N assets vs. 1 operator ⇝ autonomy & teamwork

Prior work:

- planning as model checking
- planning with temporally extended goals/control knowledge (TLPlan)

High-level strategic planning



Strategic planning (general)

Tactical planning (teams)

Individual behaviours (agents)

Conclusions

1. What can logic do for applications?
2. What can game-theory do for applications?

- e.g., planning, adversarial planning, team-level reasoning, etc.

multi-robot applications

human in the loop!

- systems should be understandable, believable, ... of help
- social choice, communication/argumentation, etc.