TOWARD TEAM-ORIENTED PROGRAMMING

David V. Pynadath

Information Sciences Institute

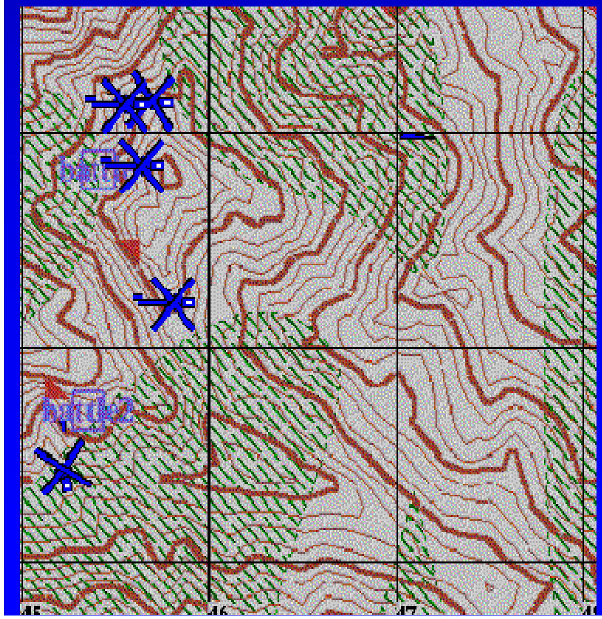
University of Southern California

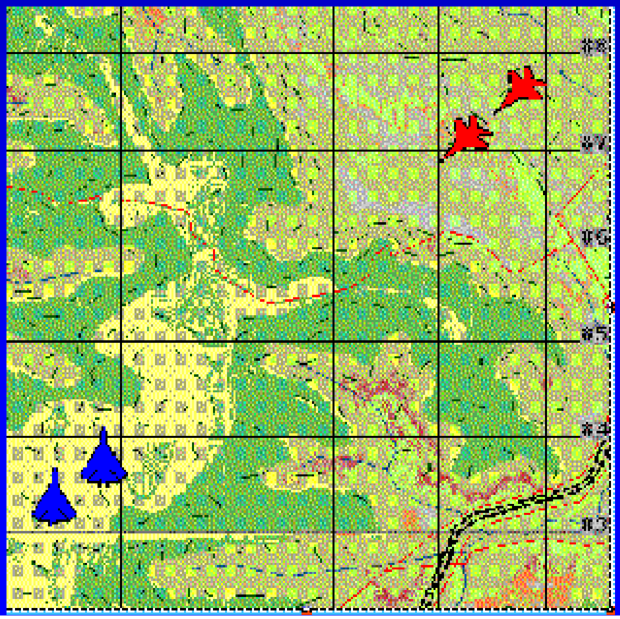
pynadath@isi.edu

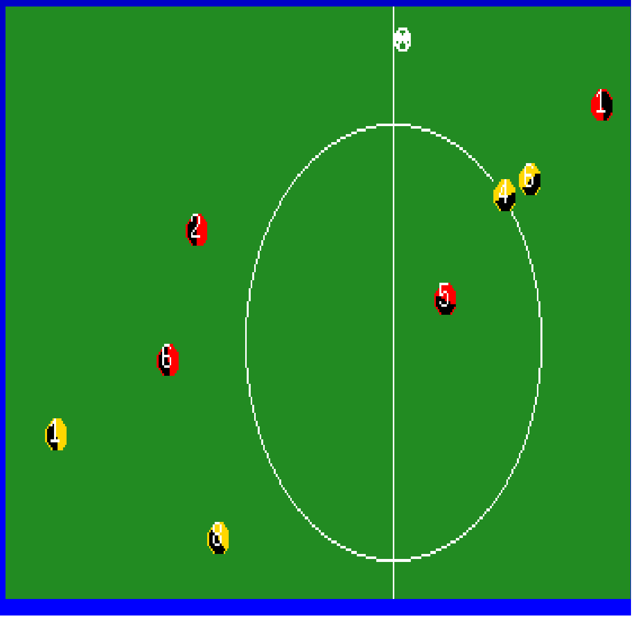
Multi-Agent Environments
Teamwork/Collaboration Application Arenas

In many environments, multiple agents must work together:

- *Interactive simulations*: Training, analysis, education,...
- *Assistants*: Assist human collaboration in planning, scheduling,….
- *Robotic:* Future “uninhabited” aircraft, distributed spacecraft...







TEAMCORE
Objectives & Approach

Objective: Enable rapid development of agent teams

- *Teams must accomplish tasks in dynamic, uncertain domains:*
  - → *Flexible & robust:* Flexible coordination & communication
  - → *Scale-up:* Resource reasoning, organizational hierarchies
  - → *Adaptation:* Learn from experience, avoid repeated failures

Approach: Make agents “team-ready”, program at team-level

- *Integrate core team reasoning & learning, to build teams from:*
  - → *New collaborative agents:* Constructed using “TEAMCORE”
  - → *Existing individual agents:* Wrapped with “TEAMCORE”

Teamwork Challenges
Flexible and Robust Teamwork

Function coherently despite uncertainties of complex domains:

- Members unexpectedly fail or discover opportunities
- Members encounter incomplete, inconsistent views of world
- Communication costly, risky, or failure-prone

Previous approach relies on domain-specific coordination:

- Lacking general teamwork model & organizational knowledge
  - → *Inflexibility:* difficult to address unanticipated events
  - → *No reusability*: within or across domains
  - → *Scalability:* rapid development of organizations difficult

Teamwork Challenges (continued)
Teamwork Adaptation

Adapt team organization with experience

- Modify roles, role-relationships, assignments (not individual skills)
  - *Convoy example: leader & follower may change*

Challenges:

- Identify critical aspects of organization (credit assignment)
- Measure impact of organizational change on performance
- Address inconsistencies in distributed adaptation
  - *Only local information may be available*
  - *Large number of trials may be difficult*

TEAMCORE’s Novel Approach to Teamwork
STEAM: Shell for Teamwork (Tambe,97)

Provides individual agents with a general model of teamwork:

- Explicitly outlines members’ commitments, responsibilities
  - *Preserve coherence, reorganize when failures, limit communication*

Provides a framework for team development:

- *Team organization hierarchy*: Team-subteam hierarchy with roles
- *Team activity hierarchy*: Explicit (reactive) team plan hierarchy

Implemented in the Soar architecture, available on the web

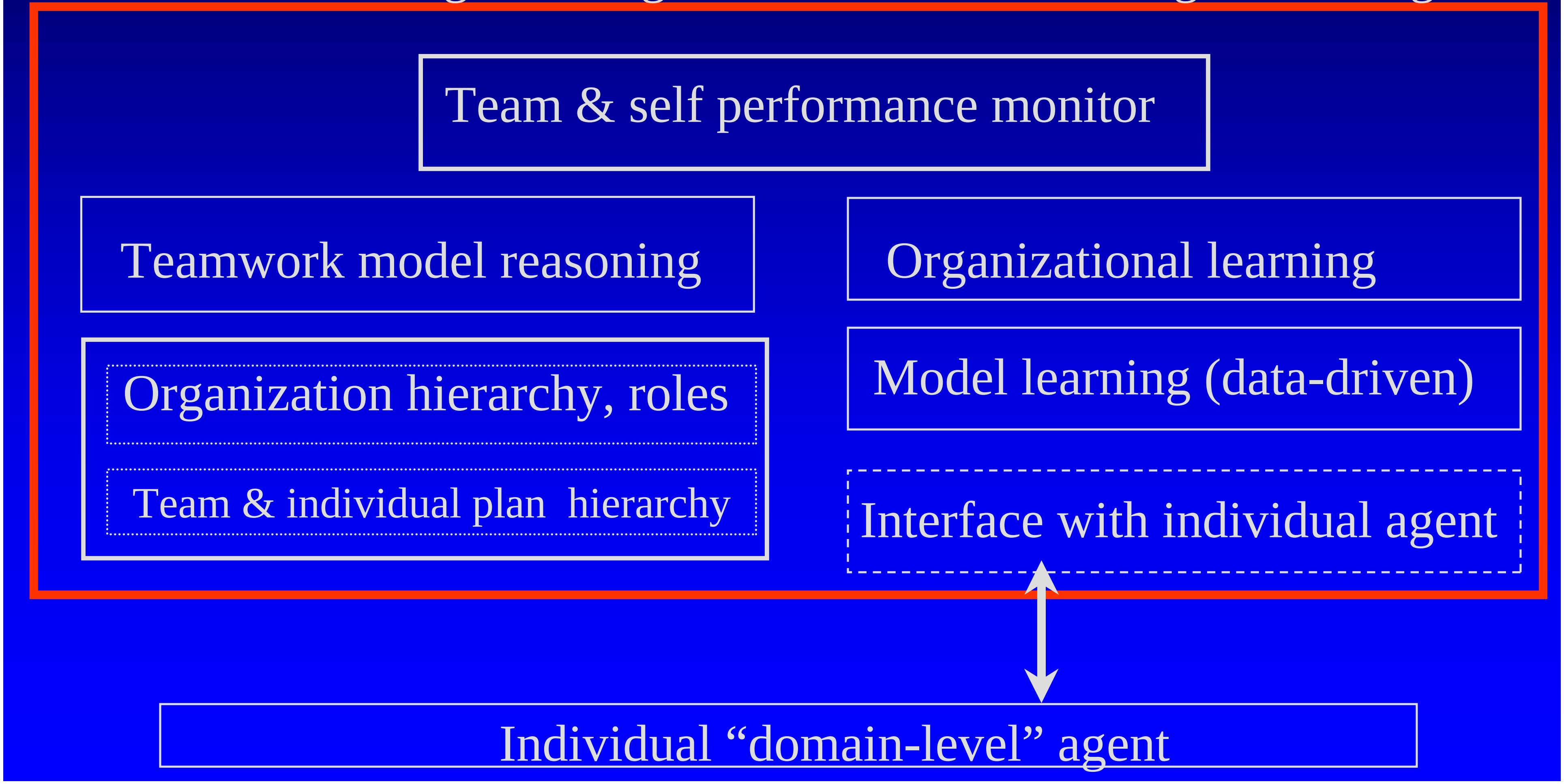
- Used for Army/Marine pilot teams in STOW, UK Navy & RoboCup
- Robustness against unanticipated events, reuse across applications

Teamwork Team

- Milind Tambe: Flexible coordination
  - Gal Kaminka (Social monitoring)
  - Hyuckchul Jung (Collaborative negotiations)
- David Pynadath & Milind Tambe: (Team-oriented programming)
  - Nicolas Chauvat
- Wei-min Shen: Team learning
  - Behnam Salemi and Jay Modi
- Maja Mataric: Team adaptation
  - Dani Goldberg
- Stacy Marsella: Web-based team analysis

TEAMCORE Architecture

TEAMCORE Agent: Integrates teamwork reasoning & learning



Team & self performance monitor

Teamwork model reasoning

Organizational learning

Organization hierarchy, roles

Team & individual plan hierarchy

Model learning (data-driven)

Interface with individual agent

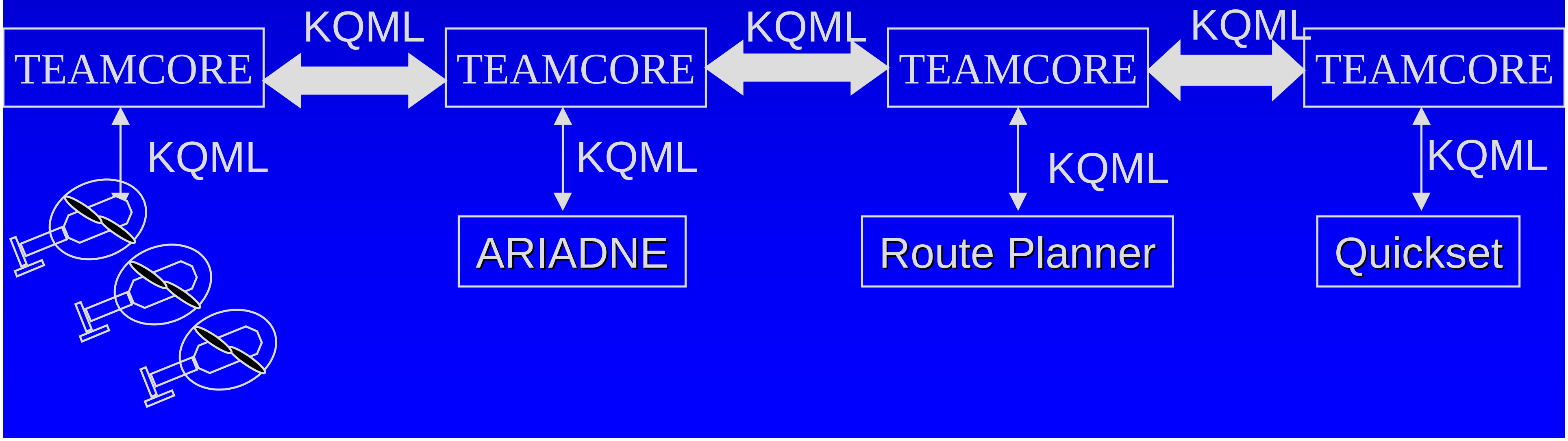
Individual “domain-level” agent

TEAMCORE: Current Status

Developed sets of TEAMCORE agents:

- Integrated STEAM teamwork framework with KQML
- Wrapped Soar and *non-Soar* agents with TEAMCORE:
  - TEAMCORE agents send requests for tasks and monitoring
  - Domain-level agents reply with results of tasks and monitoring

Participated successfully in TIE in April, 99



TEAMCORE ⟷ KQML ⟷ TEAMCORE ⟷ KQML ⟷ TEAMCORE ⟷ KQML ⟷ TEAMCORE

KQML | KQML | KQML | KQML

ARIADNE

Route Planner

Quickset

Team-Oriented Programming (TOP)

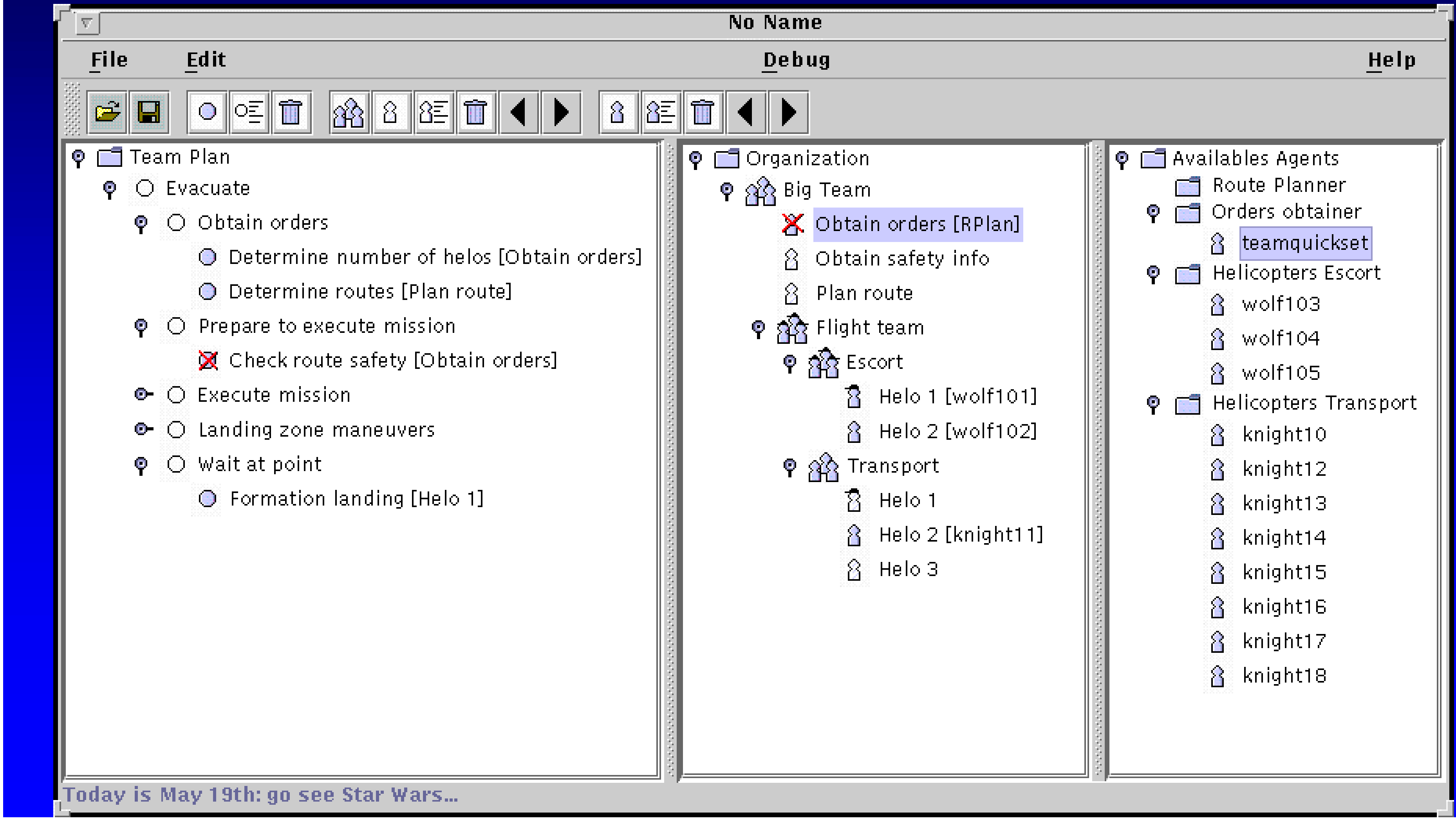
TOP for TEAMCORE, based on:

- Organization hierarchies with roles
- Hierarchical team operators, with coordination constraints
- Capability specification of domain-level agents
- Automated coordination and communication

For the heterogeneous team for evacuating individuals:

- *Organization:* Escort-transport flight, route-planning,...
- *Team procedures:* Standard flight operators, obtaining orders...
- *Capabilities:* Pilot agents can fly to point, monitor for crashes…

TOP GUI for Organizations



- Team Plan
  - Evacuate
    - Obtain orders
      - Determine number of helos [Obtain orders]
      - Determine routes [Plan route]
    - Prepare to execute mission
      - Check route safety [Obtain orders]
    - Execute mission
    - Landing zone maneuvers
    - Wait at point
      - Formation landing [Helo 1]

- Organization
  - Big Team
    - Obtain orders [RPlan]
    - Obtain safety info
    - Plan route
    - Flight team
      - Escort
        - Helo 1 [wolf101]
        - Helo 2 [wolf102]
      - Transport
        - Helo 1
        - Helo 2 [knight11]
        - Helo 3

- Availables Agents
  - Route Planner
  - Orders obtainer
    - teamquickset
  - Helicopters Escort
    - wolf103
    - wolf104
    - wolf105
  - Helicopters Transport
    - knight10
    - knight12
    - knight13
    - knight14
    - knight15
    - knight16
    - knight17
    - knight18

Today is May 19th: go see Star Wars...

TOP Issues

Scalability of team specification

- Minimization of necessary domain-specific knowledge
- Reuse of subteams within different organizations
- Reuse of team plans by multiple teams
- Reuse of domain-independent team coordination knowledge

Teams of heterogeneous agents

- Robust behavior as agents of different abilities go off- & on-line
- Recognizing need for capability transformation/translation

Coordination between team plans and domain-level actions

- Recognizing opportunities for tasking/monitoring
- Selective requests for tasking/monitoring based on available agents

TEAMCORE: Future Work & Novel Issues

Intra-team negotiations for conflict resolution:

- Argumentation-based approach, agents justify proposals & rebuttals
  - *Fully exploits team setting, e.g., teamwork models, for flexibility*
  - Initial implementation

Team monitoring:

- Learn augmented Markov models of interactions to monitor
- Monitoring domain-level agents from TEAMCORE, & implications?

Team learning:

- Learning from failures, via individual, local monitoring