Flexible Group Behavior

Randall Hill, USC/ISI

ASTT Interim Progress Review
February 9, 1998
The Approach: Flexible Group Behavior

- Develop an integrated model of:
  - Group understanding
  - Group planning
  - Group execution
  - Group learning

- Explore in context of RWA Attack Battalion
  - Soar/RWA/IFOR drives entity-level behavior
  - Soar/RWA/CFOR drives command-level behavior
Soar: Artificial and Human Cognition

- Interact with external environments
- React quickly in a context-sensitive manner
- Represent knowledge in a parallel-associative memory of rules
- Integrate knowledge in decision making
- Hierarchically decompose complex tasks
- Reflect on problems
- Learn from experience
- Use natural language
Soar in a Nutshell

I/O

Deliberation

Reaction

Reflection

Learning

Faster

Smarter
Soar/RWA

- Three kinds of missions with up to 24 helos
  - Army deep attack; Marine transport and escort
- Deployed in STOW ‘97
Soar/RWA/IFOR: Entity Level

- Accept CCSIL orders from commanders
- Hierarchically instantiate behavioral plans
- Understand (groups of) other entities
- Group execution via a model of teamwork
- React quickly and appropriately
- Deployed in all 3 missions for 3 types of helos
Soar/RWA/CFOR: Command Level

- Accept CCSIL orders and adapt to local unit
- Transmit adapted plans to subunits
- Monitor execution and generate reports
- Deployed in Army attack at company level
Towards Flexible Group Behavior

**Battalion**

- Understand ➔ Plan ➔ Execute

**Company**

- Understand ➔ Plan ➔ Execute

**Entity**

- Understand ➔ Execute ➔ Learn
Group Understanding

- **Issue**: Understanding groups of entities
  - Goals, intentions, behavior, coordination, etc.

- **Pre-ASTT SOA: RESC**
  - Small homogeneous groups
  - Use self as model of others
  - When there is a choice, select worst case
  - Use model selectivity and sharing for efficiency
  - Only works at the entity level
Group Understanding: Goals

- Larger heterogeneous groups
  - Use models of dissimilar others
    » e.g., ground forces
  - Model functional aggregates of entities
    » e.g., a company consisting of light and heavy teams
  - Use focused group perception
    » Mostly see groups rather than individuals
  - Incorporate into command level for planning use
    » Requires extending control structure of commander
    » Requires extending algorithm to full plan representation
Group Understanding: SOA 9/97

- **Dissimilar others**
  - Run models of others: “think like them”
  - Exhaustive search of model space
    » Avoid always selecting worst case hypothesis
  - Track groups of entities rather than individuals
    » Based on bottom-up, cluster-based, group perception

- **Dynamic Group Perception**
  - Perceive sets of entities as “group objects”
  - Now only perceive individual entities for one group
    » See closest group with live entities
    » Used for engaging, suppressing, etc.
Group Understanding: Progress 1

- Developed model of groups for planner
  - Enables planner to reason about groups
    » Command entity needs to monitor group execution status
    » e.g., Part of company unable to reach objective, requiring commander to replan
  - Represent groups in plan language
    » Groups represented as a “meta-constant” used in operator definitions
    » Symbols denoting groups can appear as predicates
  - Revised domain theory to include groups
Group Understanding: Example

- **PERFORM_TACTICAL_MOVEMENT**
  - **PRE-CONDITIONS**
    » AT{?GROUP ?START}
    » AT-CM{?GROUP ?START-CM}
    » ACTIVE{?GROUP}
  - **ADD:**
    » AT{?GR ?END}
    » AT-CM{?GROUP ?END-CM}
  - **DELETE:**
    » AT{?GROUP ?START}
    » AT-CM{?GROUP ?START-CM}
Group Understanding: Progress 2

- Identified perception enhancements
  - Scale up to large groups, focus of attention
  - Force formations (shape, dispersion)
  - Support to behavior recognition (scattering, ...)
  - Wrote AAAI paper describing current methods

- Raised deliberative group modeling issues
  - Issues in exhaustive model space search
  - Requires explicit representation of ambiguity
  - Making commitments under time-pressure
  - Need to abandon uniformity assumption
Group Understanding: Plans

- Extend group understanding model
  - Command entity
    » Needs to recognize others’ plans
    » Extend fidelity of group monitoring
    » Represent intent
  - RWA pilot entity
    » Ability to revise conclusions
    » Cost and probability analysis
    » Commitment under time pressure
Group Understanding: Plans

- Modify perceptual system
  - Re-implement perception strategy
    » Scale-up to large groups
    » Improve group pop-out strategies
  - Perceptual understanding of formations, behavior
    » Extend group objects with attributes for shape
    » Extend group objects with attributes to support behavior interpretation
Group Planning

- **Issue**: Plan creation for a group by a group
- **SOA**: Centralized planning by commander
  - No model of superior and minimal model of group

![Diagram of Group Planning](image_url)
Group Planning 2

- **SOA 2**: Hierarchical partial-order planning
  - Model tasks as preconditions and effects
  - Decompose complex tasks into sets of simpler ones
  - Add ordering constraints among tasks as required
    » For example, when the effect of one task can clobber the precondition of another
  - Track dependencies of planning decisions
Example Plan Fragment

1. Attack
   - Move (HA)
   - Move (BP)
   - Engage (E)

   - Move (LD)
   - Passage of Lines

   - Destroyed (E)

2. Move (HA) -> @(HA)
3. Move (BP) -> @(BP)
4. Engage (E) -> @(HA)
Group Planning: Goals

- Collaborate during planning
  - Accept feedback on plans and modify accordingly
  - Generate feedback to superior and subordinates
  - Exploit group/superior understanding in planning
  - Violate received plan if necessary, staying within intent
  - Use differential capabilities in allocating tasks
- Replan when necessary during execution
  - Replan when current plan becomes infeasible
  - Use dependencies to guide replanning
- Time, conditionality, and information gathering
Group Planning: SOA 9/97

- **Meta-planning**
  - Support for replanning and collaborative planning
  - Makes use of meta-literals & meta-operators
    - “these tasks are a plan”, “this plan is shared knowledge”,
    - “create a plan”, “transmit a plan”, etc.

- **Replanning**
  - Commander replans when threats to plan exist
    - Retract and add tasks and constraints
  - Meta-information determines if changes are part of shared knowledge, and thus must be transmitted
Planner Architecture

Domain Theory

Plan Network

Plan 1

Plan 2

Declarative World Description

ModSAF
Meta-level Planning

Meta-level Plan

Generate (Co, P)

Transmit (Co, P)

Execute (Co, P)

Plan Network

Plan P

Tsk1

Tsk2

Tsk3

World Description

Planner

ModSAF

Co = Company (Group)
P = Plan
Meta-level Replanning

Plan Network

Meta-level Plan

Generate (Co,P)
Repair(Co,P)
Transmit (Co,P)
Transmit(Co,P)
Execute (Co,P)
Execute(Co,P)
Failure causing Information

Plan P

Tsk1
Tsk2
Tsk3

World Description

Planer

Co = Company (Group)
P = Plan

ModSAF
Group Planning: Progress 1

- Developed concise plan language
  - Abstract language for expressing domain theory
  - Soar-independent
  - Significant step toward technology transfer and reuseability of methods for flexible group behavior

- Plan language incorporates facts, run-time variables, filter conditions
  - Computes binding values to facts
  - Gets run-time variable bindings by executing task
  - Filters indicate when to decompose a task
Group Planning: Progress 2

- Reimplemented company domain theory
  - Used new plan language
  - Incorporated ability to reason about groups
- Developed Soar wrapper
  - Provides interface between plan language and Soar
  - Compiles plan operators into Soar rules
  - Such a wrapper needed for any planner implementation
Group Planning: Progress 3

- Iterative repair approach to replanning
  - Random walk
- Developing more formal understanding of iterative repair approach underlying planner
  - Investigate soundness and completeness
  - Prove properties and establish limitations
- Wrote AAAI paper describing Soar independent algorithm for meta-reasoning
Planner Architecture Progress

Plan Network

Plan 1

Plan 2

Declarative World Description

Domain Theory

Planner

Domain interface language

Input interface language

Output Interface language

ModSAF
Group Planning: Plans

- Extend plan representation
  - Subplans
    » Supports replanning portions of a plan
    » Need to be able to send plan changes (i.e., frag orders)
  - Enable parts of plans to be contracted out
    » Collaborative planning with subordinates
    » Design contracting conventions
  - Allow concurrent plan execution and plan repair
  - Focus of attention issues in planner
    » Attend to changes in environment, versus resolving outstanding issues in the current plan
Group Planning: Plans

- Develop Battalion Commander domain theory
  - Extend company commander domain theory
  - Need an SME—outstanding issue
  - Otherwise, use common sense & documentation
Group Planning: Plans 3

- Begin work on temporal reasoning
  - Significance: reason about time for group monitoring, planning, and execution
  - Ability to plan events at specific time
    » e.g., arrive at control measure by 0800 hours
  - Time-outs for tasks
    » e.g., commander assumes control measure not achieved if no confirmation by 0800 hours
  - Temporal constraints
    » e.g., time between leaving Holding Area and Engaging enemy cannot exceed 20 minutes
Group Execution

- **Issue**: Coordinated behavior in groups
  - Maintain even when faced with unexpected
- **SOA**: Goal-oriented teamwork (STEAM)
  - Model goals, actions, roles, and commitments
    - Model for both individuals and groups
  - Use models (and situation) to drive action selection, communication, role assumption, etc.
  - Only works at the entity level
- **Goal**: Teamwork at command level
  - Replanning must take into account current group composition and capabilities
Group Execution: Progress

- Command entity
  - Capable of stating how to react to failures
    » Based on the extended domain representation
    » Facilitates replanning
    » Coordination when a plan fails
  - Capable of detecting and planning around events that can lead to failure
    » Based on extended constraint representation
      - Maintenance goals
      - Anticipating failures
Group Learning

- **Issue:** Improve understanding from experience
  - Only one small piece of overall group learning

- **SOA:** Inductively propose model changes
  - Compare successful and failed predictions
  - Use self-explanation to focus on relevant features
  - Only for one kind of change, in one situation, with domain-dependent knowledge

- **Goal:** Broader classes of changes and situations in a domain-independent manner
Milestone 2: 12/97

Design Review 1

- Approach to scaling up group understanding
- Approach to temporal, conditional and information-gathering planning (cut)
- Approach to teamwork and group understanding in commander (Changed to group monitoring)
Milestone 3: 9/98 (revised)

Technology POP Demonstration 2

- RWA Attack Battalion
- Demonstrate advanced group understanding
  » Scale up to larger groups of entities
- Demonstrate advanced group planning
  » Temporal (delayed to ‘99), conditional and information-gathering planning (cut)
  » Group understanding (monitoring only)
  » Collaborative planning (added)
Milestone 3: 9/98 (revised) 2

- Demonstrate advanced group execution
  » Commander utilizes teamwork model (scaled down and delayed to ‘99)
- Deliver software and domain independent descriptions of new capabilities
Architecture as of Milestone 3

Understand → Plan → Execute

Company

Understand → Plan → Execute

Entity

Understand → Execute

Battalion

Understand → Plan → Execute
Milestone 4: 12/98

Design Review 2
- Approach to learning improved group models
- Approach to group emotions (cut)
- Approach to collaborative planning
Milestone 5: 9/99

Technology POP Demonstration 3

- RWA Attack Battalion
- Demonstrate advanced group understanding
- Demonstrate more advanced group planning
  » Collaborative planning (moved to ‘98)
  » Temporal planning (moved from ‘98)
  » Group understanding: plan recognition
- Demonstrate advanced group execution
  » Commander utilizes teamwork model (scaled down)
- Demonstrate group learning
  » Improve group models through experience
Milestone 5: 9/99

- Demonstrate group emotions (cut)
  » Fear
- Deliver software and domain independent descriptions of new capabilities
Architecture as of Milestone 5

Understand → Plan → Execute

Battalion

Company

Understand → Plan → Execute

Entity

Understand → Execute → Learn
Projected Results Summary

- Group understanding techniques
- Group planning techniques
- Group execution techniques
- Group learning techniques
- Transition plan
Projected Results 1

- **Group understanding methods**
  - Understand large heterogeneous groups
    » Goals, intentions, behavior, coordination, etc.
  - Use models of dissimilar others
    » e.g., ground forces
  - Model functional aggregates of entities
    » e.g., a company consisting of light and heavy teams
  - Use focused group perception
    » Mostly see groups rather than individuals
  - Incorporate into command level for planning use
Projected Results 2

- Group Planning Techniques
  - Collaborate during planning
    » Accept feedback on plans and modify accordingly
    » Generate feedback to superior and subordinates on plans
  - Exploit group/superior understanding in planning
    » Violate received plan if necessary, while staying within intent
    » Use differential capabilities of subgroups in allocating tasks
  - Replan when necessary during execution
    » Replan when current plan becomes infeasible
    » Use dependencies to guide replanning
  - Incorporate temporal reasoning
Projected Results 3

- Group execution techniques
  - Teamwork at command level
  - Replanning must take into account current group composition, capabilities, and state
Projected Results 4

- Group learning techniques
  - Broader classes of changes and situations in a domain-independent manner
Transition Plan

- Viability of flexible group behavior techniques
  - Demonstrate software on automated force
- Domain-independent description of techniques
  - Behavioral and architectural requirements
  - Algorithms
  - Technical and conference papers, journal articles
  - Personal meetings with designers and engineers