Towards Human-like Adversaries for MOUT Training

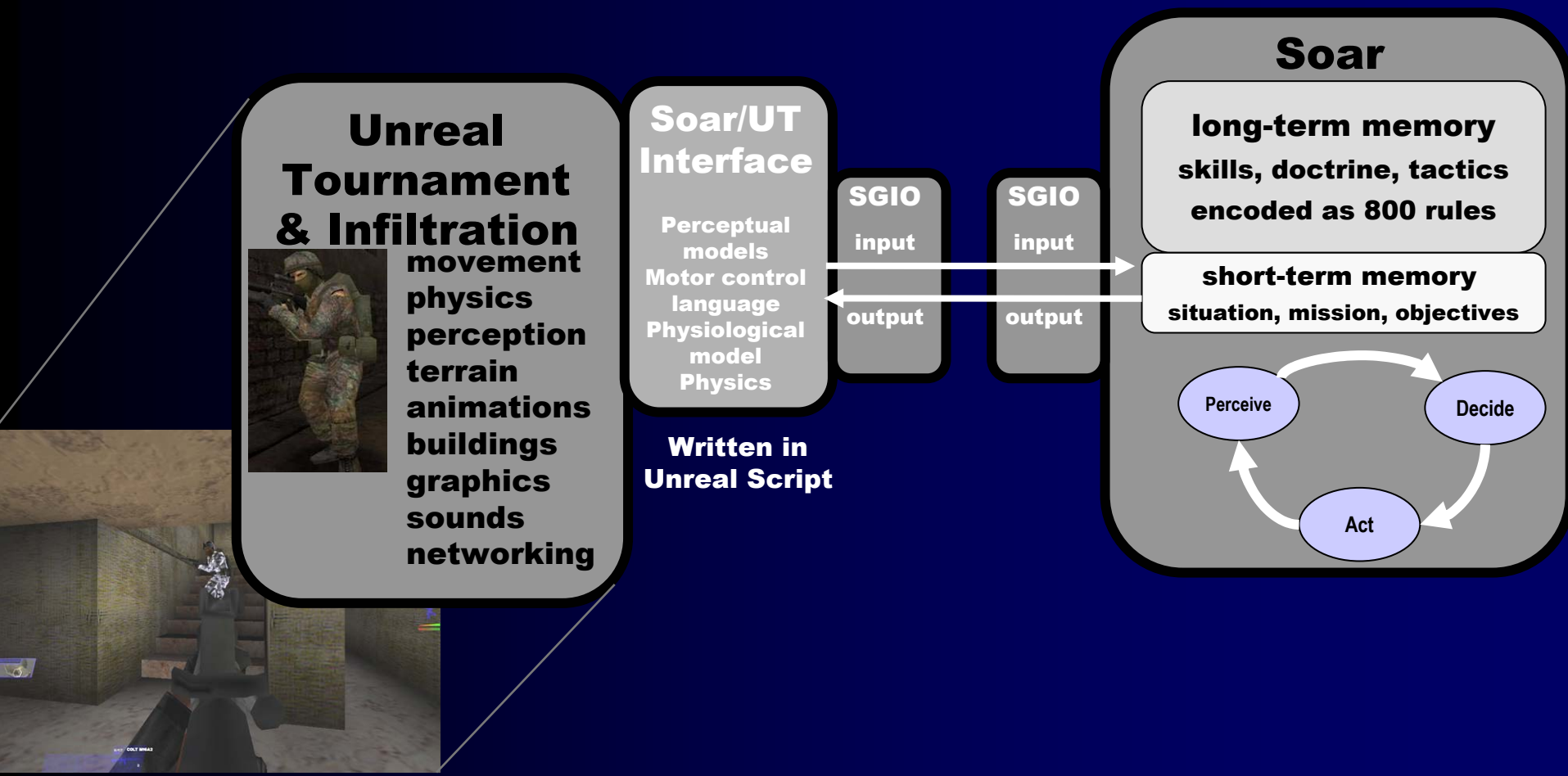
John E. Laird
University of Michigan
Robert Wray
Soar Technology

23rd Soar Workshop

Project Objectives

- Develop *human-like adversaries* for MOUT training
  - Independent of specific simulation environment
  - Human-like, realistic behavior:
  - Adversaries
- Efficient and scalable implementation
- Discover and explore emergent research issues

Approach: Overall Design



Comparison to Earlier Work

- Compared to TacAir-Soar
  - Compressed time scale
    - Faster reaction time required
    - Shorter, more dynamic missions
  - Less complex mission structure
  - Looser teamwork and coordination
  - More spatial reasoning

- Compared to Quakebot
  - Different missions and objectives
  - Based on expert knowledge
  - Some teamwork, communication, and coordination

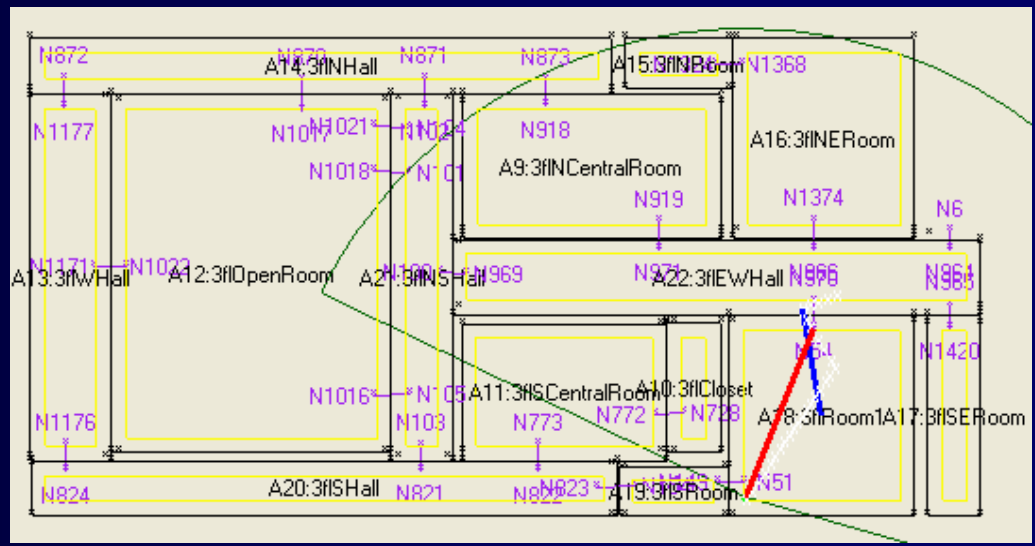
Development Principle: Observational Fidelity

- Complex domain with many elements to consider:
  - doctrine and tactical knowledge, spatial and temporal knowledge
  - coordination and communication with other entities
  - courage and cowardice, indecision, leadership
  - startle responses, reaction to light, noise, smoke, debris
  - emotion, mood, physiological moderators, etc.

- Observational fidelity:
  - Concentrate on elements *observable to trainees*
    - Simplify non-observable behaviors (eg., at-ease)
    - Avoid detailed internal models when behavioral role is minimal (simple model of visual perception)
  - Represent what would be observable – don't cheat
    - Simulation of all physical movement through space
    - Coordination via observation, common knowledge, and communication when necessary

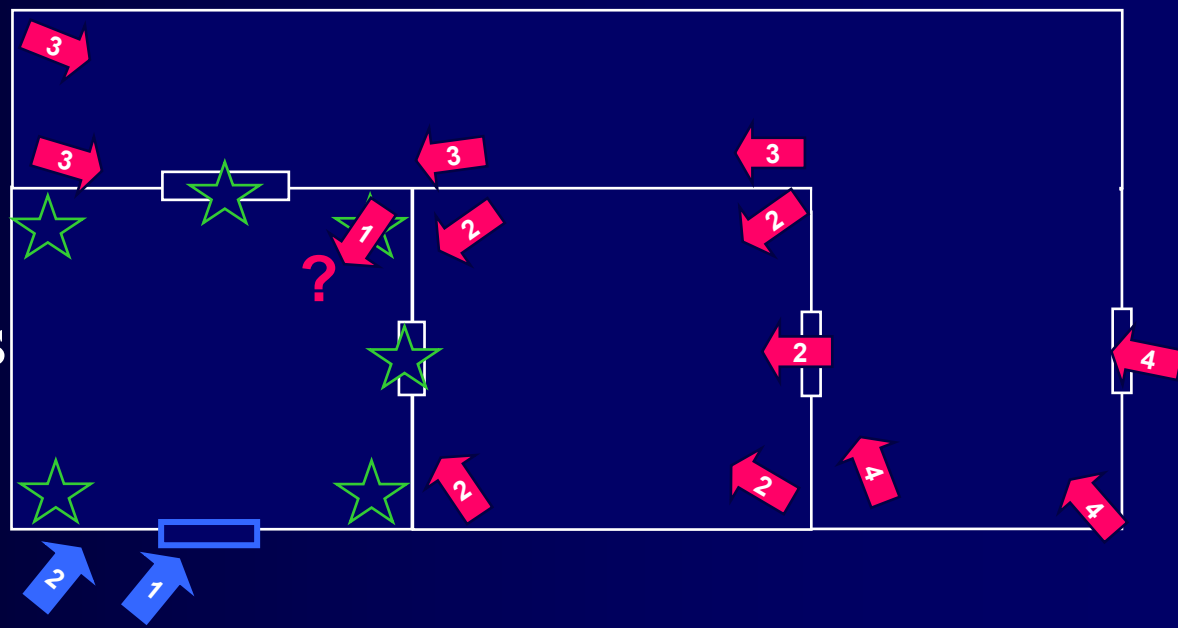
Competent Across Wide Variety of Situations

- Movement within and between rooms
- Understands topology of building
  - Knows paths between rooms, visibility between rooms, types of rooms
  - Built up automatically via exploration of building
- Situational Awareness
  - Categorize situation: available weapons, ammo, enemy, health, …
  - Record and manage information about threats
  - Record and manage information about friendlies
- Weapon Management
  - Reload, unjam weapon
  - Choose weapon based on situation
- Tactics used appropriately
  - Attack with gun
  - Attack with grenade
  - Retreat
  - Hide and pop out
  - Defend a room
  - Roam
  - Sentry
  - Surrender
- Mission-based behavior
- Communication & coordination via realistic messages



Variability

- Defensive positions
  - Positions in corners, near doors, away from threats
  - Standing, kneeling, or prone
- Reaction to sighting enemy
  - Attack with gun or grenade (only limited situations)
    - Reload, unjam, …
  - Retreat
  - Hide
  - Surrender
- If enemy disappears
  - Defend
  - Retreat
  - Hide
  - Attack with grenade



Controlled Variability

- User can easily modify weights of selecting between actions
  - Can control generally and in specific situations
  - *Uses new numeric indifferent preferences*
- Our experimental weights

Action	Weight	Action	Weight
Attack	80	Defend	40
Attack-Grenade	10	Sentry	10
Hunt	40	Hide	20
Reload	44	Retreat	30
Roam	10	Surrender	50

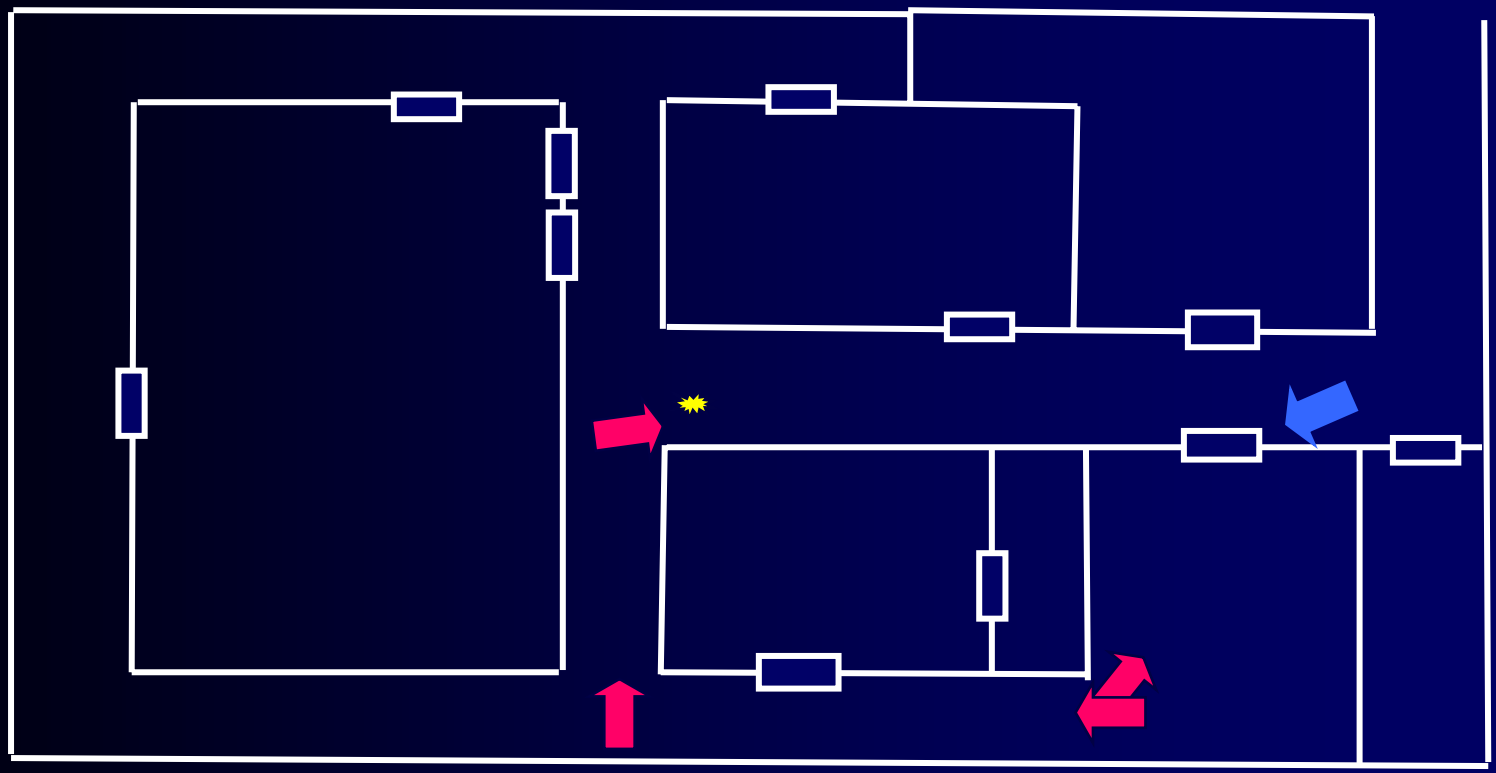
- User can also make decisions deterministic
  - Testing and debugging
  - Highly constrained behavior
- Different bots can have different probability distributions
  - Provides across subject variability

Multiple Types of Coordination

- Missions
  - Assignment of leaders & subordinates
  - Multiple teams: individual, fire team, squads
- Situational Awareness
  - Monitor location of threats, teammates
  - Move to maximize fire on targets, limit fratricide
- Communication
  - Group situational awareness
    - Location of threats, status, etc.
  - Changing positions
    - Retreating
    - Relocation (orders from leader)

Surprises

- Never know what to expect:
  - Grenade thrown as coming round the corner
  - Multiple retreats so the bots outflank the humans



Potential Future Research

1. Expert Evaluation of Behavior
2. Mission Specification and Variability Control
3. Tactic Acquisition Tool
4. Adaptation and Learning
5. Automated Pedagogical Direction of Training Exercise
6. Behavior Moderators

- *Interface to VIRTE Demo II Simulation Environment*
  - *JSAF & NetImmerse*
- Refine and add new behaviors as necessary

Nuggets and Coal

- Builds on QuakeBot, SGIO
- Compelling demonstration of Soar
- Inspired us to add architectural variability: numeric indifferent preference
- Inspired us to consider agent building tool based on generalize examples
- Potentially lead to more use of Soar in training
- Still very time consuming to develop Soar Bots
- Still very time consuming to develop interface to environment