Thinking

...*inside* the box

The Design Space of Control Options for Computer-game AIs

Bob Wray
25th Soar Workshop
16 Jun 2005

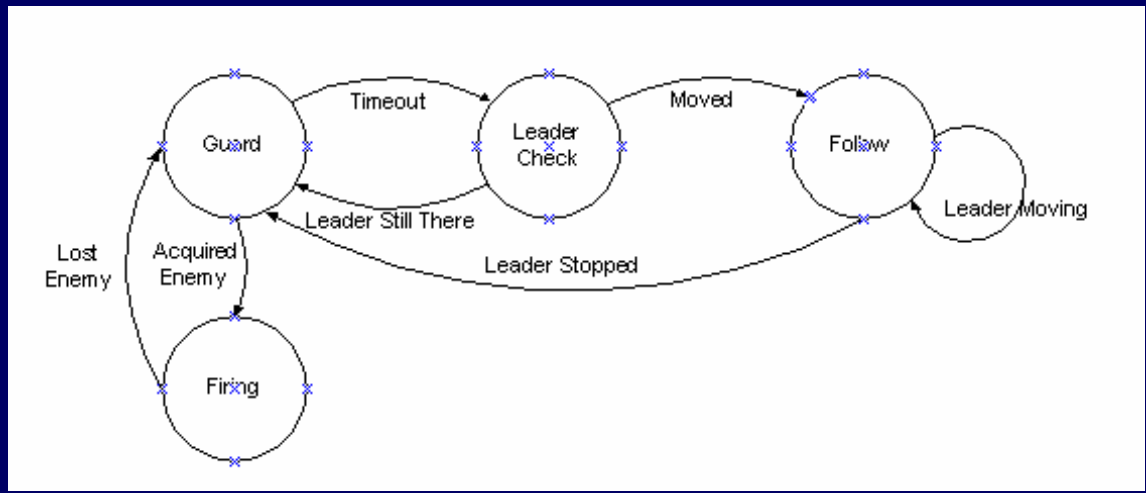
Other Contributors: Mike van Lent, Jonathan Beard, Paul Brobst

Problem

- Want 100's – 1000's realistic entities on a single PC (simulation or game platform)
- Current technology forces a difficult tradeoff:
  - Finite state machines (FSMs)
    - High multi-entity scalability
    - Limited knowledge scalability
    - Limited realism (especially coordination)
  - Agent-based approaches (E.g., Soar)
    - Much greater levels of realism feasible
    - Individual entities are resource intensive (relatively large memory and CPU footprints)
      - Current multi-entity scalability: 20-40 Soar agents (MOUTBots)
- Initial Research:
  - Define space of possible hybrid solutions
  - Analysis to identify approaches that promise greater realism and greater entity scalability
  - (Paper to be presented at IJCAI 2005 workshop)

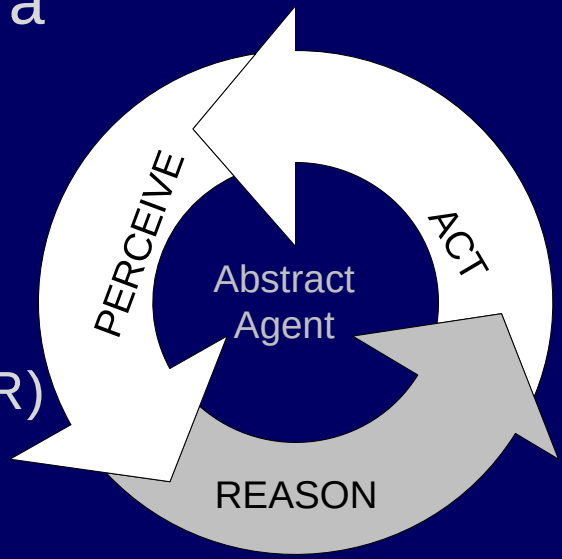
Finite State Machines

- Most computer games & many simulation systems use simple behavior representation technology
  - Finite state machines
    - "State" of agent defines actions available
    - Combinatorial explosion in states generally limits knowledge representation
  - Simulation examples: JCATS state editor, OneSAF CBT
- Strengths
  - Cheap to build
  - Small computational overhead



Agent technology

- Agents are often constructed from a "virtual machine" that attempts to capture processes and representations needed for intelligent decision making
  - *Cognitive architecture* (eg, Soar, ACT-R)
  - Built-in support for:
    - Pattern-directed processing
    - Least commitment (run-time decisions)
    - Conflict resolution (choosing between available options)
    - Learning
  - Often have some validation wrt human psychology (e.g., both Soar and ACT-R used for cognitive modeling and HBRs)

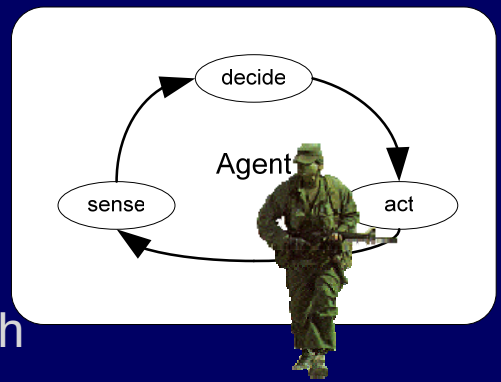


Evaluation Dimensions (for game AI)

- Believability
  - Quality of behavior generated by entities
  - Not an absolute measure; dependent on overall requirements (observational fidelity)
- Multi-entity scalability
  - Maximum number of entities supportable in a specific game environment
  - More entities: richer (potential) interactions among agents and player(s)
- Knowledge scalability
  - Ability of technology to support increasingly large knowledge store
  - Assumption: Knowledge scalability is required for realism and rich interaction
- Simplicity of systems engineering
  - Ease of use, ease of integration with simulation environment
  - Implementation technology needs to fit in game software development cycle

Potential Hybridizations

- Controller point-of-view
  - Entity (egocentric)
  - Commander (global view)
- Grain-size of behavior representation
  - Individual (one entity in simulation) OR
  - Aggregated (control of a group of entities)
    - Assume multiple vehicle control (platoon of tanks vs. tank crew)
- Implementation technology
  - Homogeneous (Agent XOR FSM) OR
  - Heterogeneous (both)
- Examples:
  - Entity, Individual, FSM: Typical game approach
  - Entity, Individual, Agent: Typical Soar approach
  - See paper for review of (almost) all combinations

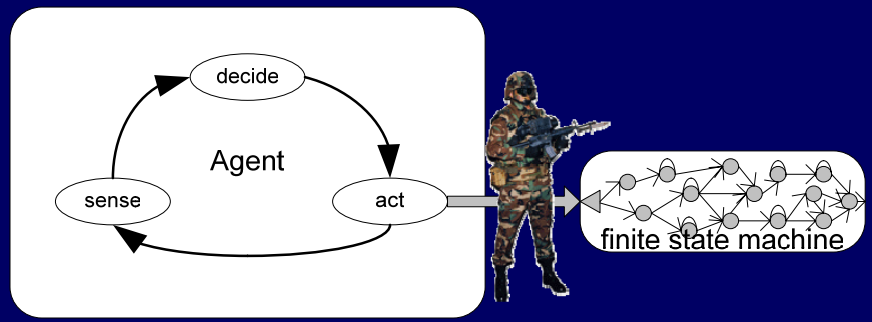


Multiple controllers/entity

PoV: Entity
Grain-size: Individual
Technology: Heterogeneous

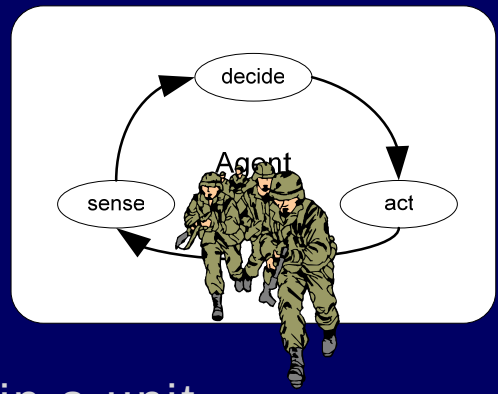
Distinct sub-options

1. Static mix: Use Soar to select behaviors; use FSMs to execute them
   - Comparable to 3T(iered) architectures (sequencing & reactive layers)
   - Requires both Soar and FSMs; likely will not improve multi-entity scalability
2. Dynamic mix: Switch back-and-forth between FSM and Soar
   - Potentially improves both scalability and realism by matching right technology to situation
     - Selectively reduce overall Soar footprint
     - Selectively generate high-fidelity behavior when needed
   - Research needed: When to choose between FSMs & Soar
   - More complex & costly development process (duplicate behavior representations)
   - Related Research: IMPRINT & ACT-R dynamic switching



Multiple entities/controller

PoV: Entity
Grain-size: Aggregate
Technology: Heterogeneous

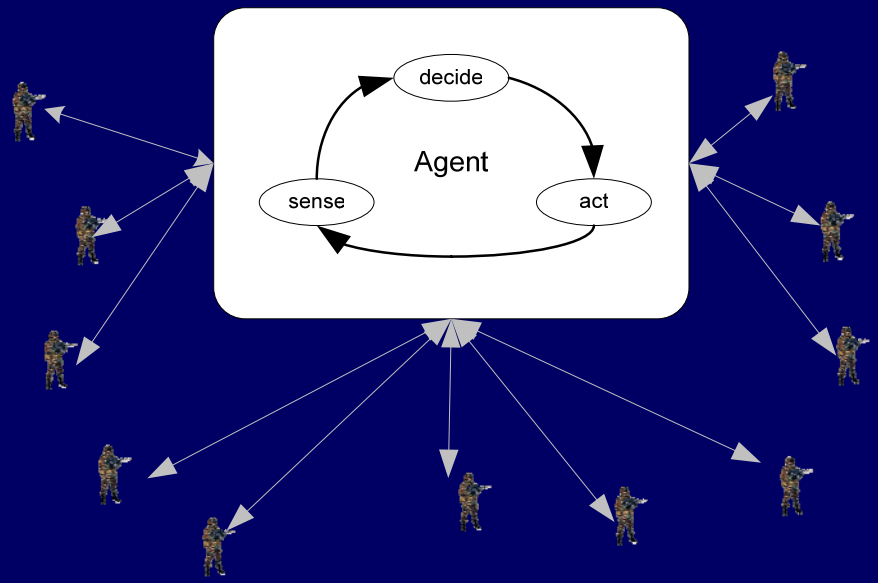


- Single agent instance controls small number of entities
- Strengths
  - Best case: scale to *n*X entities, where *n* is number of entities/unit
  - Simplifies coordination between entities within a unit
- Limitations
  - Need to support multiple, parallel goals (not well suited to Soar constraints -- but see Randy's talk)
  - Additional KE requirements (master control)
  - Additional implementation complexity
- Soar Tech: (initial) RAID architecture
- ICT: Full Spectrum Command

Game player controllers

PoV: Commander
Grain-size: Individual
Technology: Heterogeneous

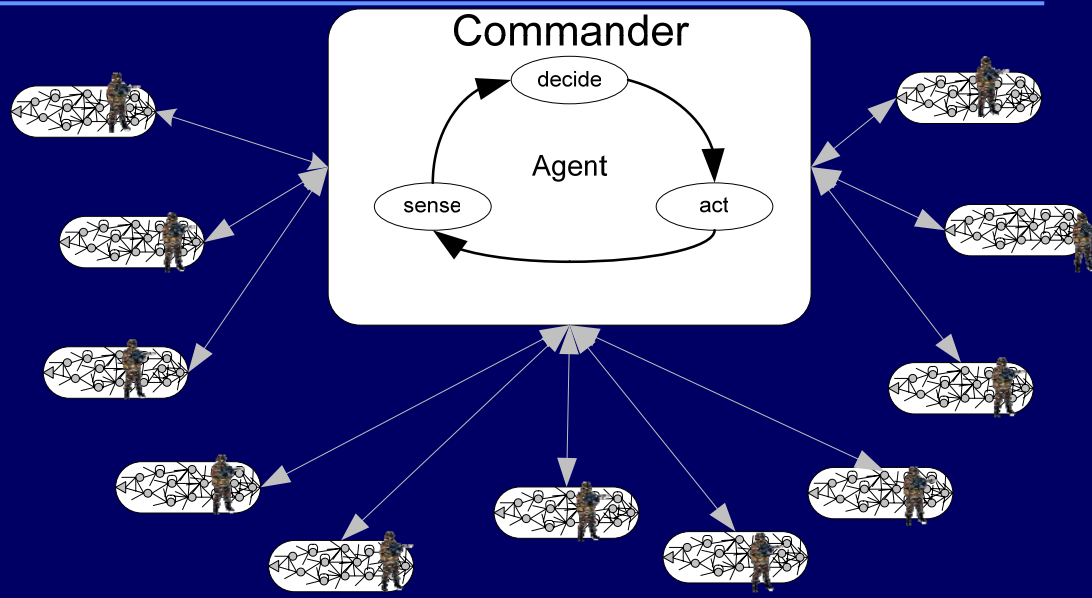
- Commander agent directly controls all the entities in the simulation
  - Agent1 (move-to 10,7)
  - Agent2 (shoot, Threat6)
  - Agent3 (go prone)
- Feasible for chess, likely not feasible for dynamic games
- Soar: Many of same limitations as Multiple entities/controller



PoV: Commander & Entity
Grain-size: Individual
Technology: Heterogeneous

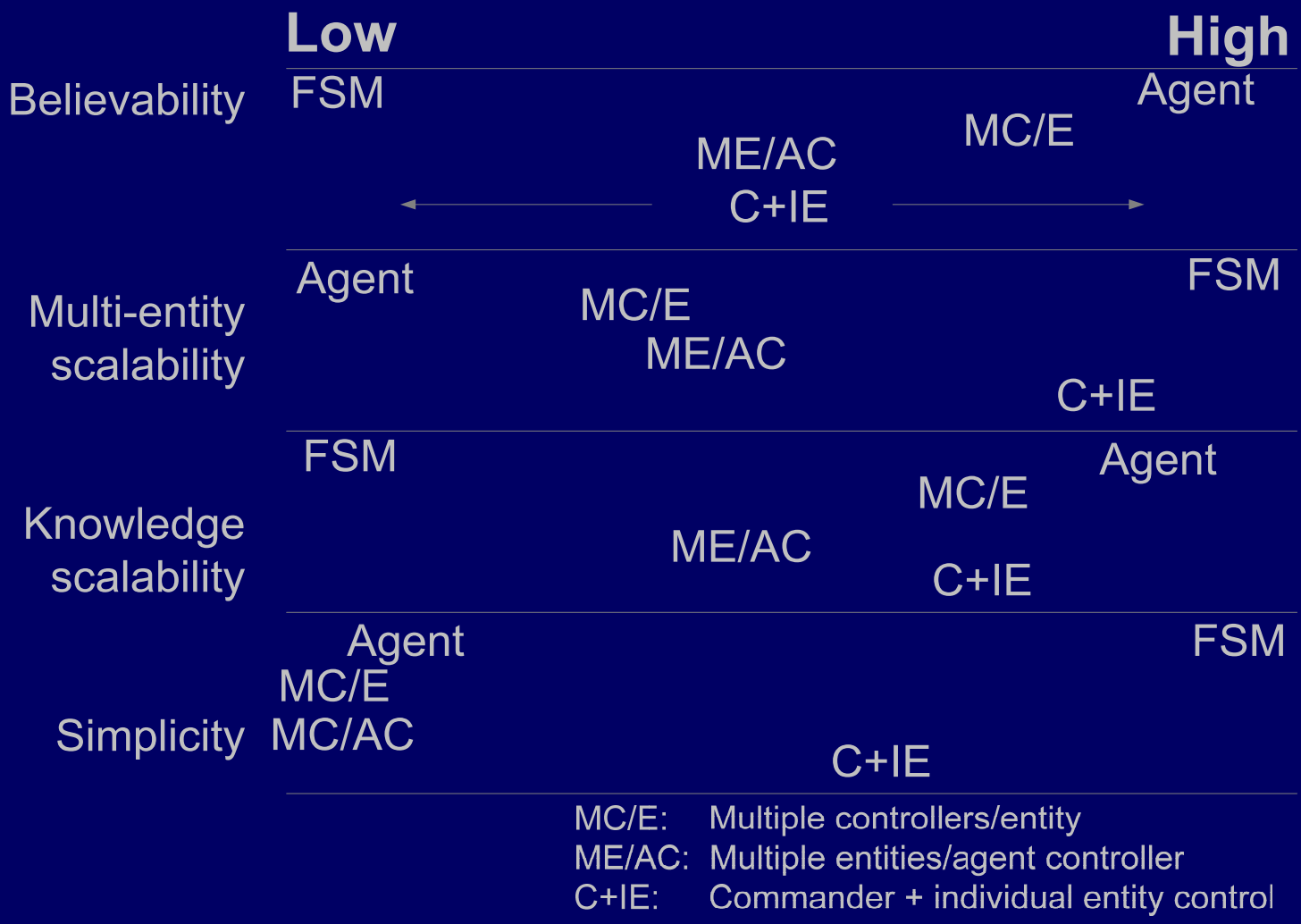
Commander + entity control

- Combine agent commander with entity level (FSM) controllers

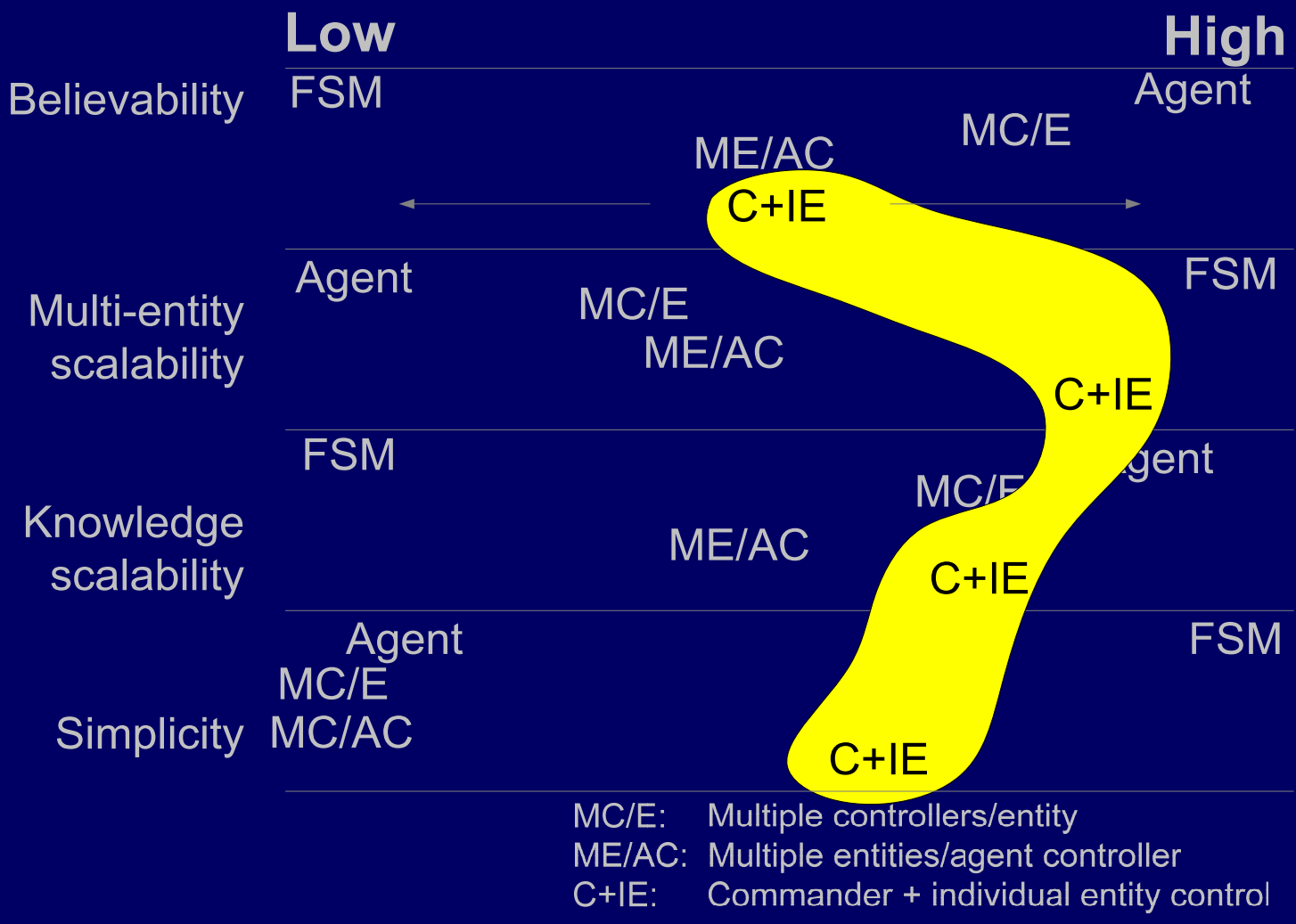


- Assumption: Global perspective of commander can increase realism (e.g., increased coordination) at the entity level
- Significantly increases communication bandwidth
- Soar Tech: CIANC3

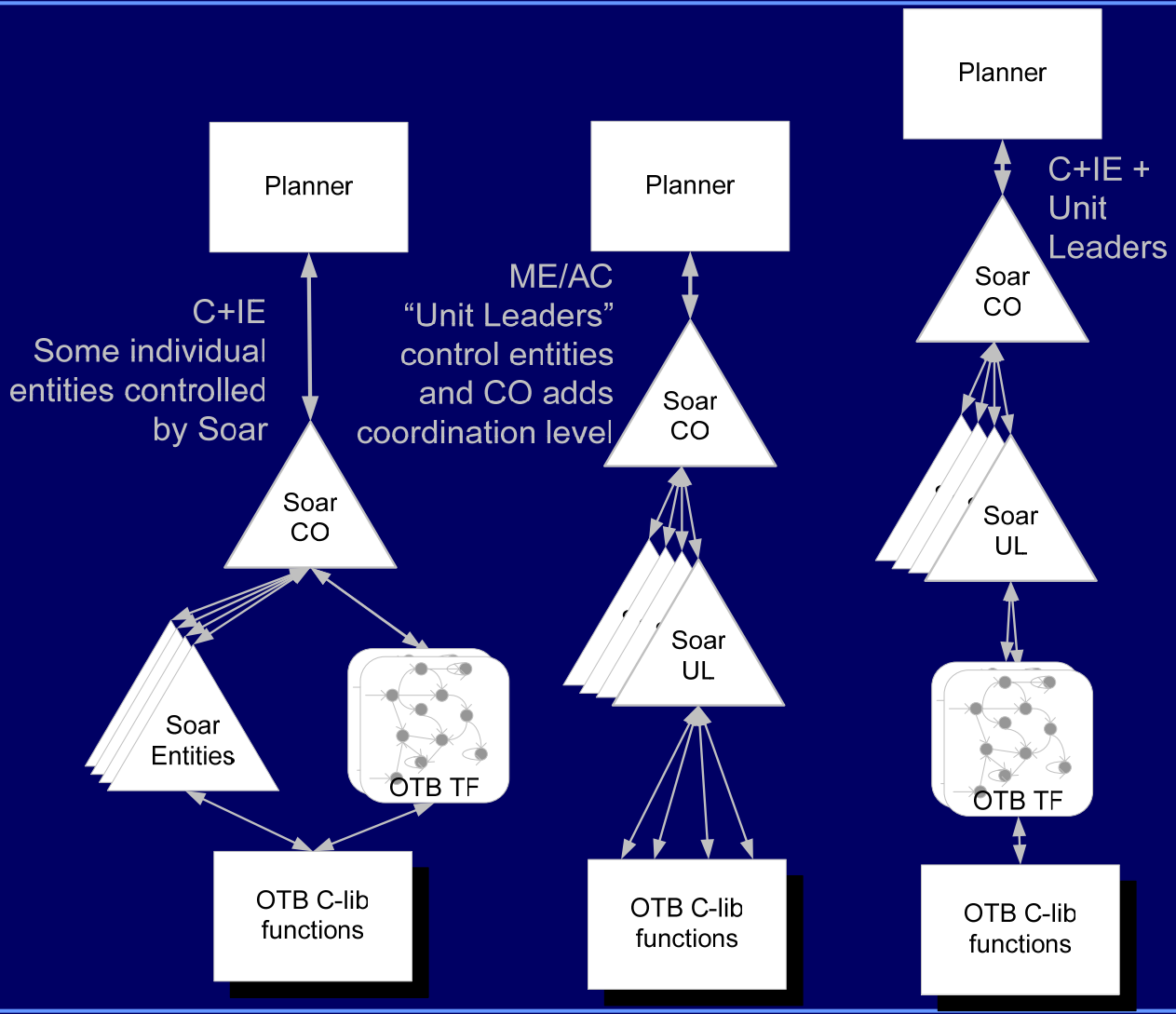
Summary of Initial Analysis



Summary of Initial Analysis



Architecture Considerations for JFETS



Future Work, Nuggets & Coal

- Future work
  - Explore options empirically (JFETS project)
  - Develop (better) metrics for evaluation dimensions
  - Are these the right dimensions?
  - Are there additional/alternative dimensions for other applications? (e.g., training)
- Nuggets
  - Hybrid approaches look promising for addressing some scalability limitations of Soar
  - Potential research areas (unit leaders, commander, control switching)
- Coal
  - Soar not the right tool for 100+ entities on one PC
  - Operator bottleneck limits consideration of some options