Analysis of Multi-agent Scenarios

- Several tasks have to be solved synchronously
- Tasks may be composed from subtasks
- Tasks may be accomplished by different methods
- Methods have quality, duration, agents
- Methods may enable, facilitate, hinder other methods
- Agents plan local method schedules in coordination with other agents
- Schedules have to adapt to changes (environment, enemy activity, failure)

Currently, schedulers are evaluated through *simulation*

**Reasoning tools can analyze real-world effects of scenarios**

Use logic of events to specify tasks, methods, dependencies, timing…

Prove that *every instance of an overall scenario*, under certain assumptions about possible changes to the world, implies a certain goal property at its end

Reasoning task requires next generation tools to complete the proof in “real” time
Formal Analysis of multi-agent scenarios

Example formalizations

Specifications in the logic of events can be generated from C-TAEMS task structures

• Events e (beginning) and e’ (end) will satisfy task CTask3bE with quality q if they satisfy (belong to) one of the six methods

• Events e and e’ satisfy method Engage3bA with quality q if they are triggered by agent3, describe a duration between 7 and 13 and if q is between 7 and 13.

Formalization in logic of events:

Subtasks: CTask3bE(e,e',q) ⇔ Engage3bA(e,e',q) ∨ Engage3bB(e,e',q) ∨ ... ∨ Engage3bF(e,e',q)

Methods: Engage3bA(e,e',q) ⇔ loc(e)=loc(e')=agent3 ∧ q ∈ {7..13} ∧ time(e')−time(e) ∈ {7..13}

... Engage3bF(e,e',q) ⇔ ...
Formal Analysis of multi-agent scenarios

Example formalizations

- Windows2 is composed from CTask2a and CTask2b: e2 and e2’ satisfy Window2 with quality q2 if there are events between e2 and e2’ that satisfy CTask2a and CTask2b with quality q2a and q2b and q2 is the sum of these qualities

- CTask1 enables CTask2:
  If e2b and e2b’ satisfy CTask2b, then there must have been events e1, e1’ before e2b (the beginning event) that satisfy CTask1

\[
\text{Subtasks: } \text{Window}_2 (e_z, e'_z, q_z) \iff \exists e_{2a}, e_{2a}', q_{2a}, e_{2b}, e_{2b}', q_{2b}.
\]
\[
\quad \text{CTask}_{2a} (e_{2a}, e_{2a}', q_{2a}) \land \text{CTask}_{2b} (e_{2b}, e_{2b}', q_{2b})
\]
\[
\quad \land e_z \leq \{e_{2a}, e_{2a}', e_{2b}, e_{2b}'\} \leq e_z'
\]
\[
\quad \land q_z = q_{2a} + q_{2b}
\]

\[
\text{Enable: } \text{CTask}_{2b} (e_{2b}, e_{2b}', q_{2b}) \Rightarrow \exists e_1, e_1', q_1. \text{CTask}_1 (e_1, e_1', q_1) \land e_1' \leq e_{2b}
\]
Formal Analysis of multi-agent scenarios

Example statement

Objective: Prove that certain desired properties hold at the end of every instance of a scenario.

- If e and e’ satisfy Root-Task with a quality q that is at least q-min and the initial state of the world at e satisfies Init-Assumptions and for all events between the beginning and the end of the scenario a Change-Assumption holds, then the state of the world at the end e’ satisfies Goal-property.

\[
\forall e, e'. \forall q \geq q\text{-min.} \quad \text{Root-Task}(e, e', q) \land \text{Init-Assumptions}(e) \\
\land \forall e'' \in [e, e']. \text{ Change-assumption}(e'') \Rightarrow \text{Goal-property} (e')
\]

- Proof provides deeper insights but requires refinements of a scenario
  - Axioms about relation between methods and their effect on the state of real world
  - Assumptions about possible changes (enemy actions, environment changes, …)