MOSIM Framework

Dr. Klaus Fischer

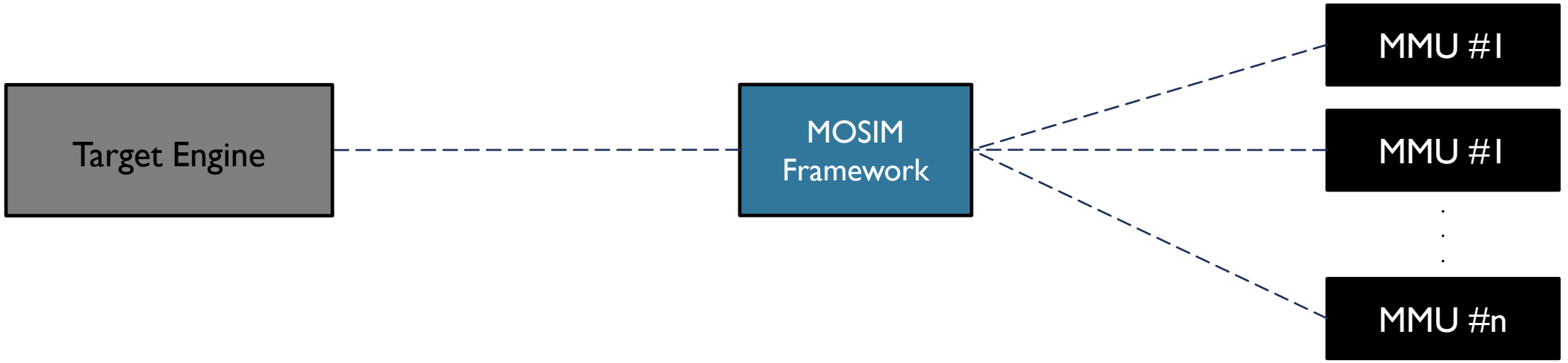
Webinar, 29th of November 2021

End-to-end Digital Integration based on Modular Simulation of Natural Human Motions

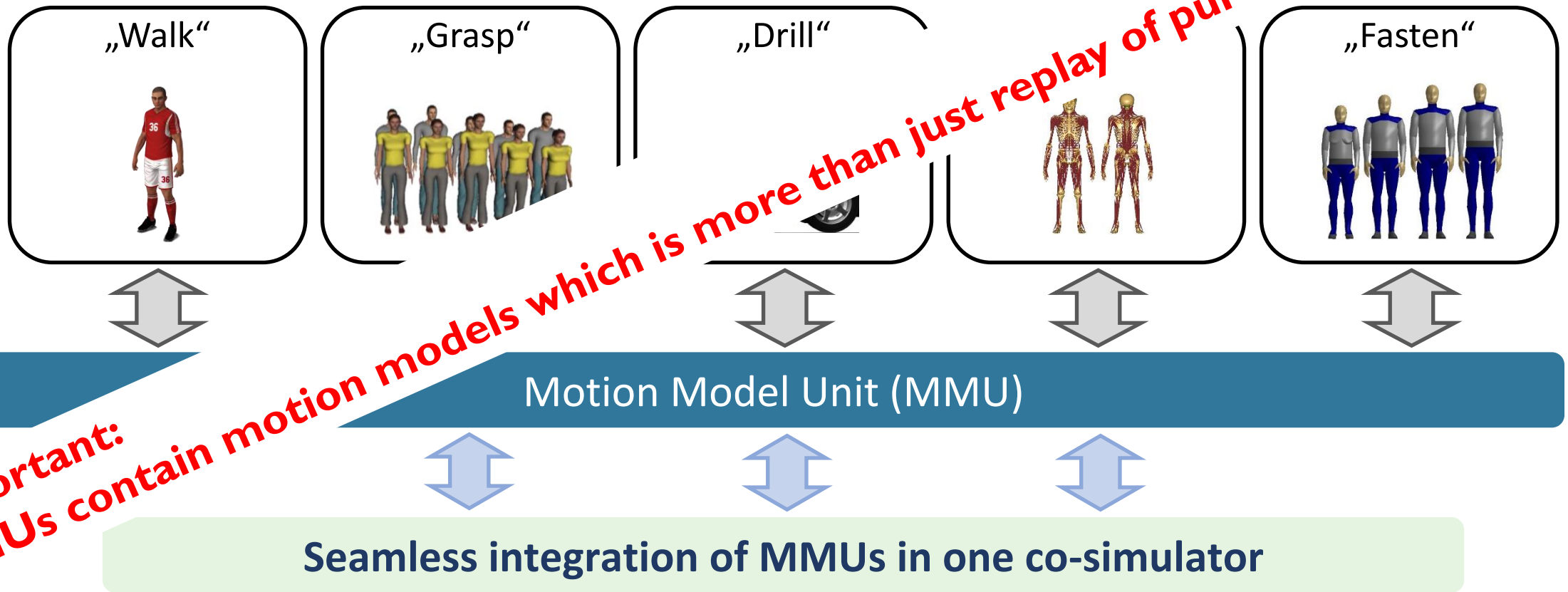
ITEA 3

Goal Description: Integration of different Motion Synthesis Approaches

- **MMU** (Modular Motion Unit): (black-box) motion synthesis approach
- **Target Engine**: (game-) engine, in which the visualization is displayed
- **Goal**: Combine multiple sequential and parallel MMUs to display a complex motion sequence in the target engine.



MOSIM will adapt the **functional mockup** concept to field of ' motion synthesis as a solution to this problem



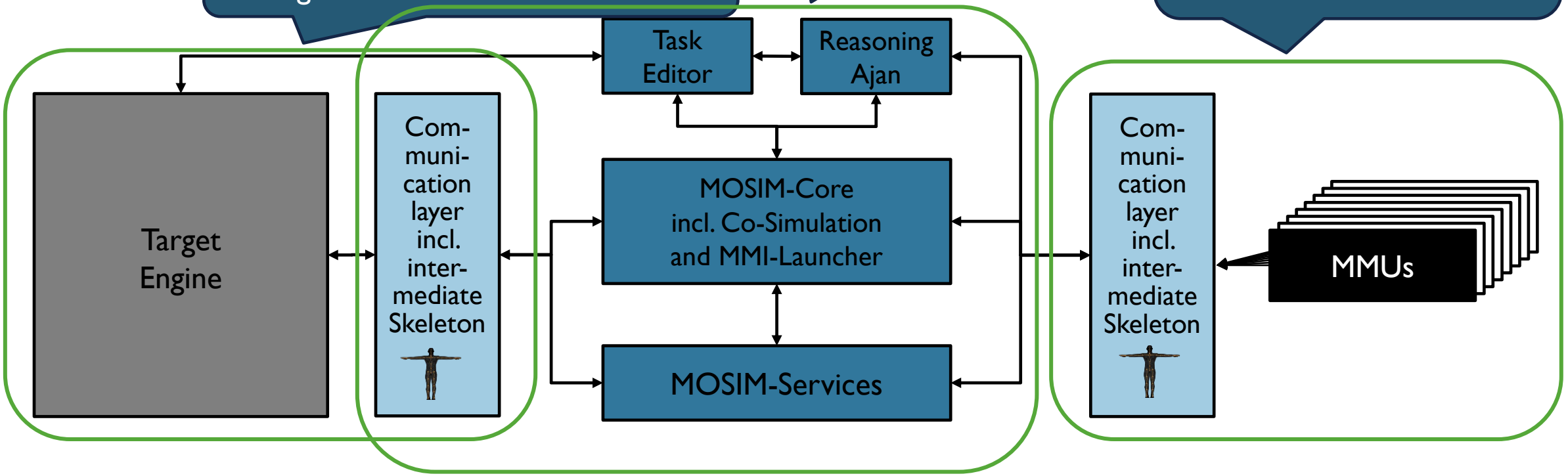
Important: MMUs contain motion models which is more than just replay of pure animations clips!

MOSIM Architecture

Assembly of Motion Model Units together into a co-simulator

Flexibility of using existing target engines via communication API's

Library of Motion Model Units for user specific applications



legend

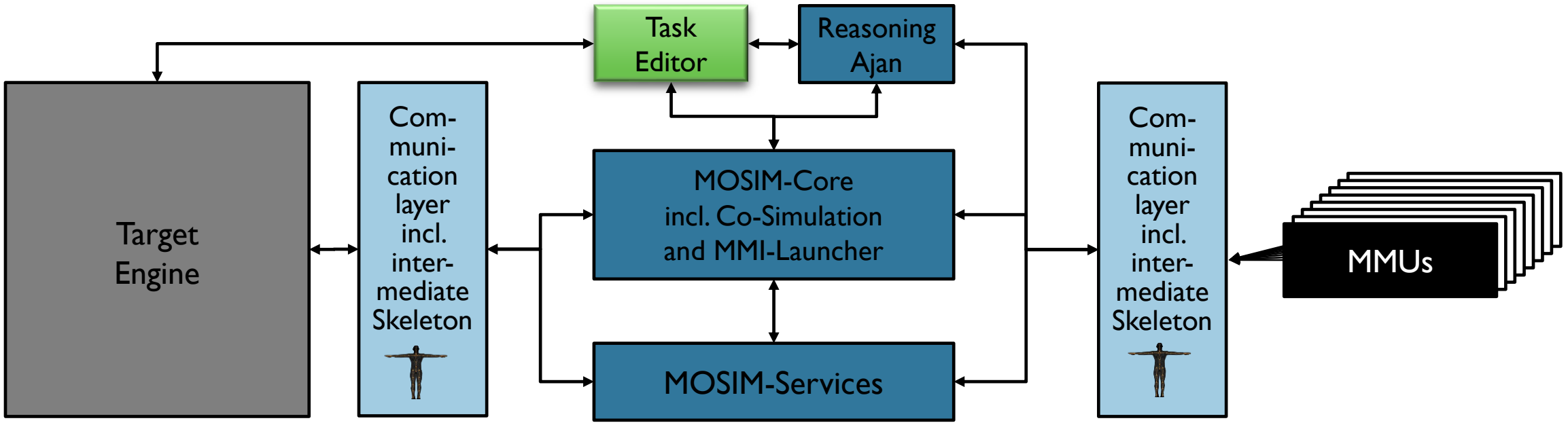
Standardized Interfaces

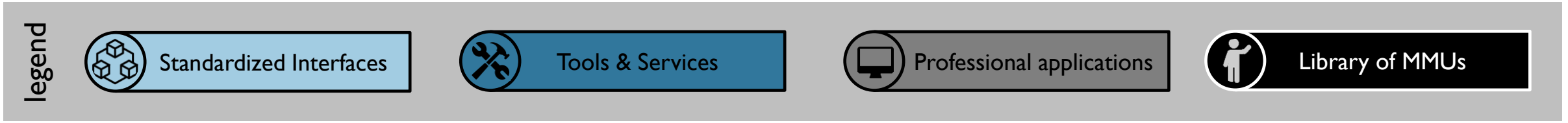
Tools & Services

Professional applications

Library of MMUs

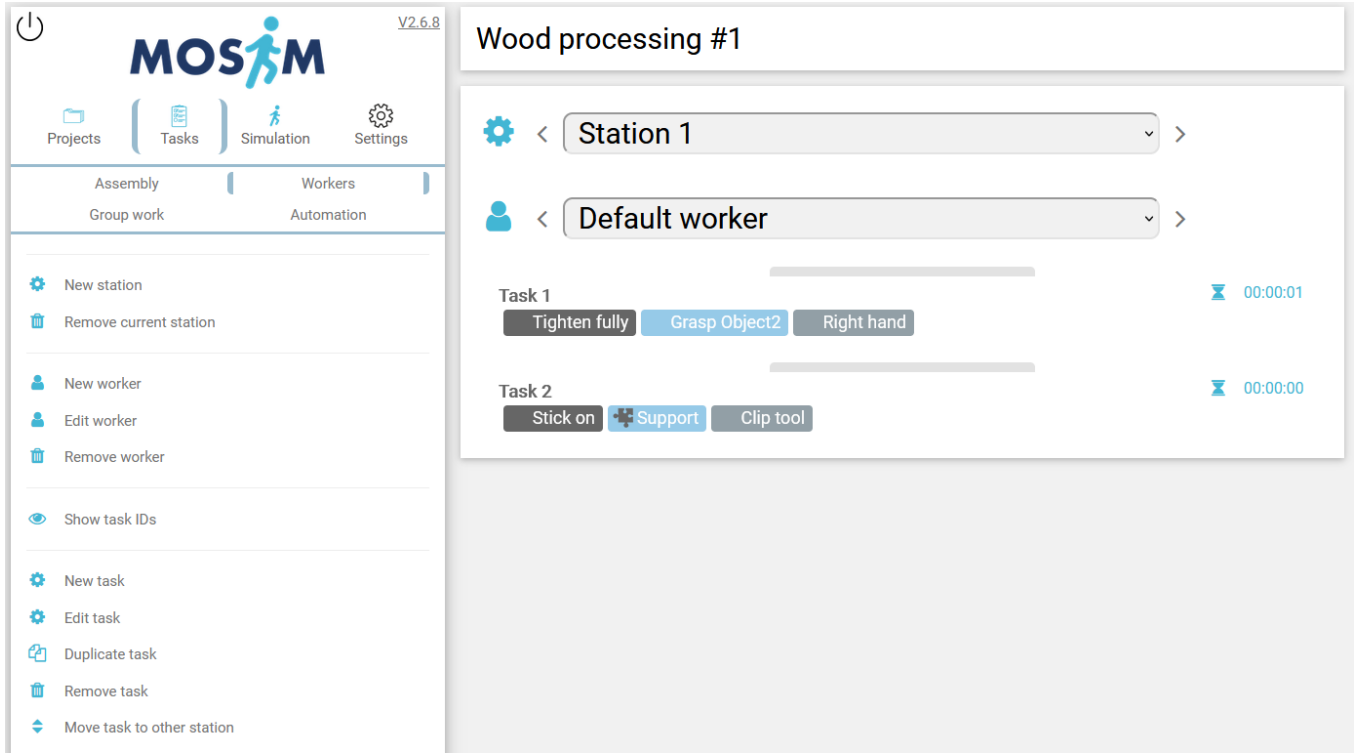
Architecture





Task editor functions

- High-level task editing (tasks are independent from the scene layout and state)
- Station/location definition
- Tool types editing
- Parts classification and grouping
- Definition of avatars and their parameters
- Project user management
- Project related MMU library (sharing MMUs within group)

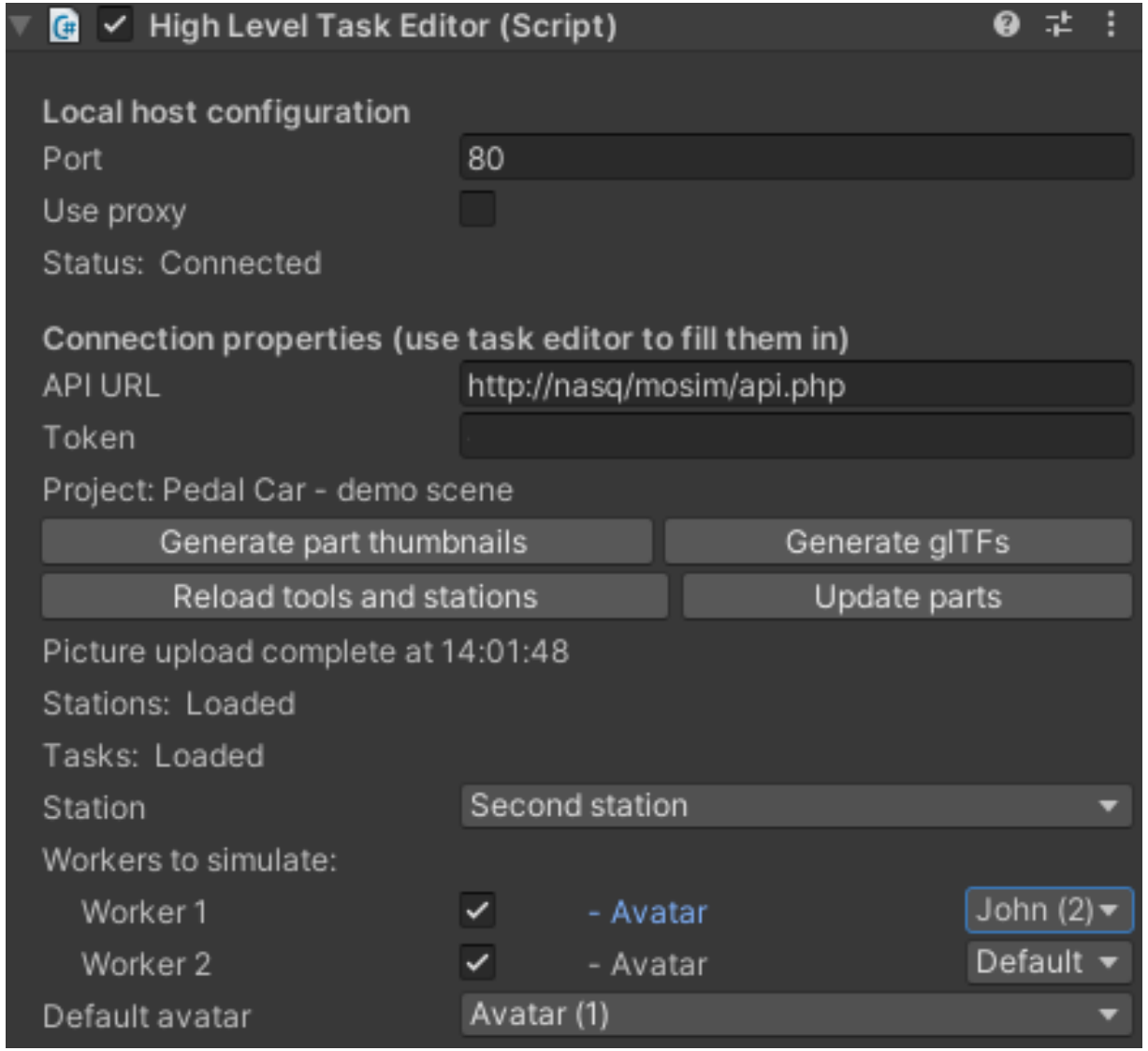


Task editor integration to Unity

- Synchronization of parts between Unity and task editor web service (names, place in hierarchy, 3D model, thumbnail)
- List of tool types sync from task editor web service
- Station synchronization
- Avatar to task list assignment
- Task list interface for reasoning agent

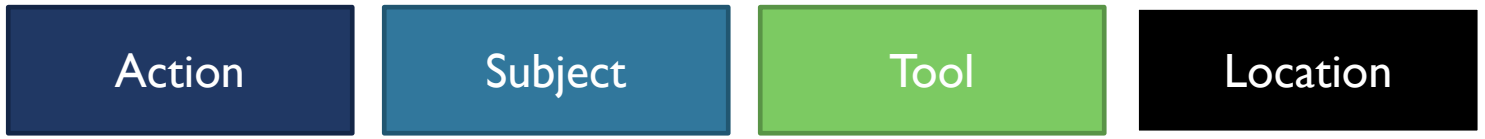
In preparation:

- Action-based camera control
- Simulation automation task list (interaction of machines, humans, camera, and events)



High-level task editing concept

- Task comprises of 4 components



- Examples:

  "Position" -> "Engine block" -> "With both hands" -> "Worktable"

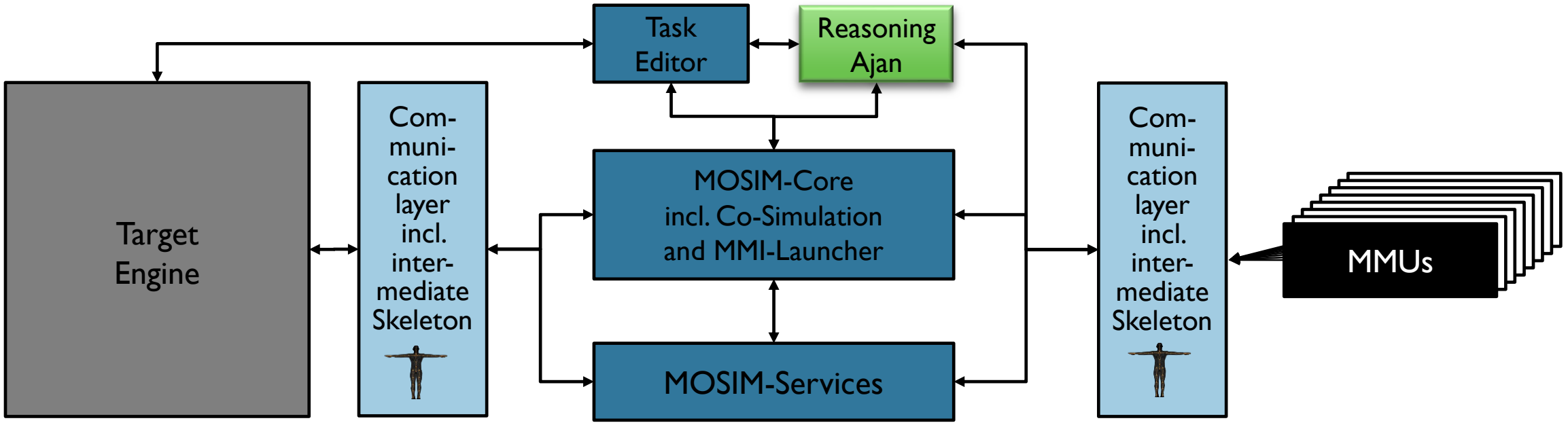
  "Clean" -> "Large cover" -> "Cloth"

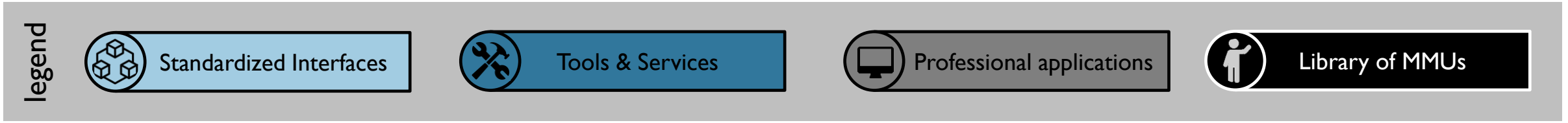
  "Tighten fully" -> "M6 screw" -> "EC screwdriver" -> "hole 6"

- Deriving MMU-level tasks (low-level)

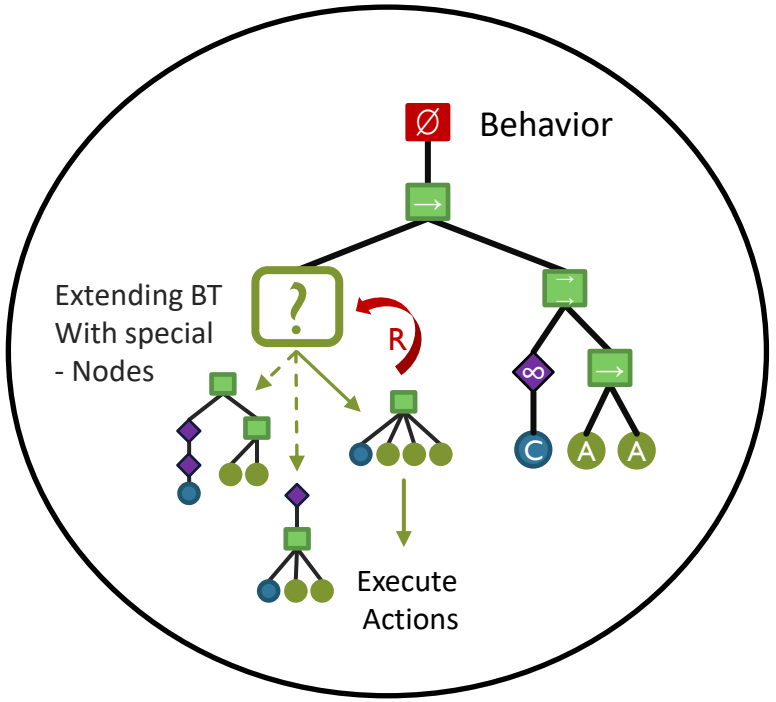
- Scene objects are referenced in high-level tasks
- Actual object locations and state are taken from the scene at the simulation execution time
- MMU-level tasks are scene-state specific, while high-level tasks remain unchanged for any scene layout (provided no part required for task has been removed from the scene)

Architecture





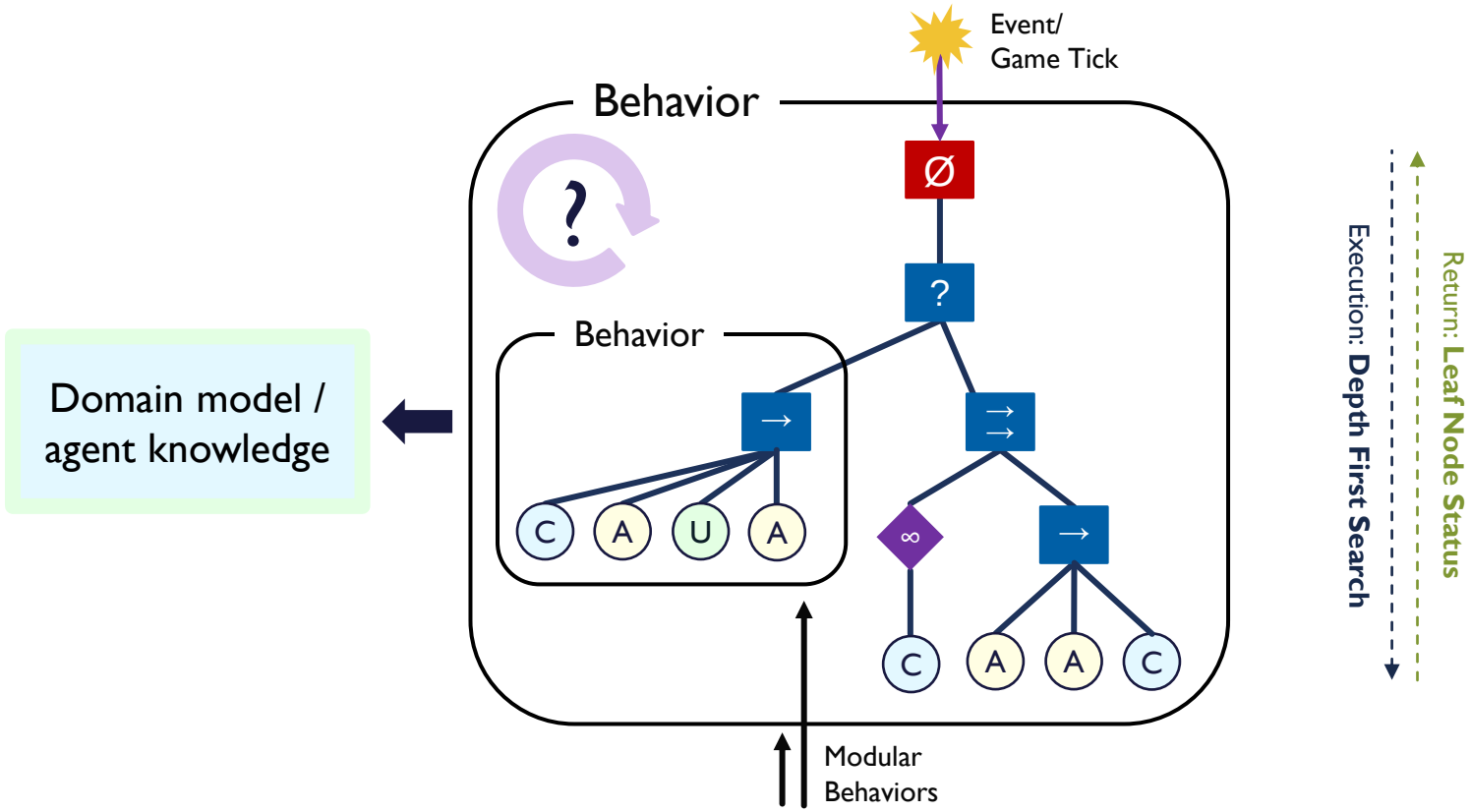
Behavior Trees

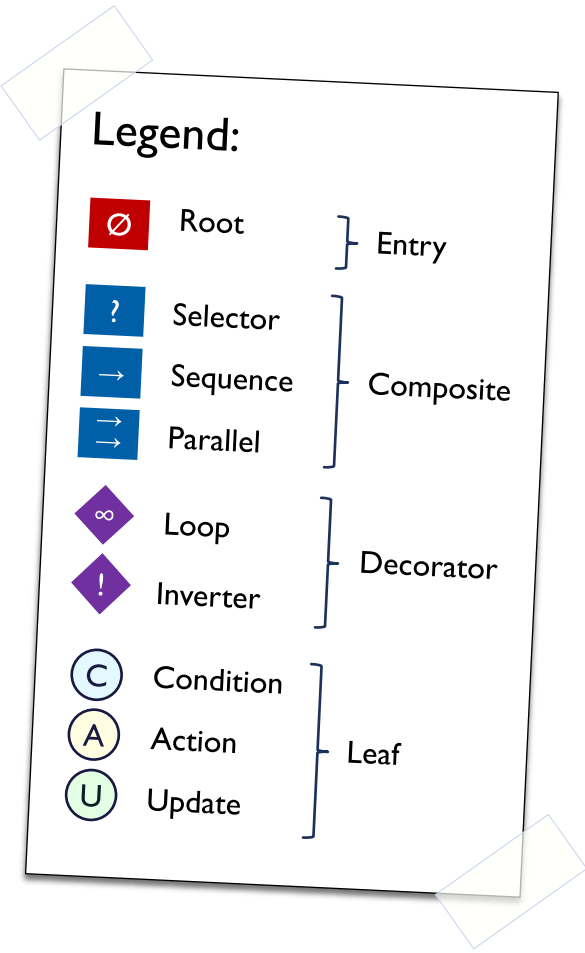


→ BTs are used in various domains

→ BTs are Modular and easy to extend

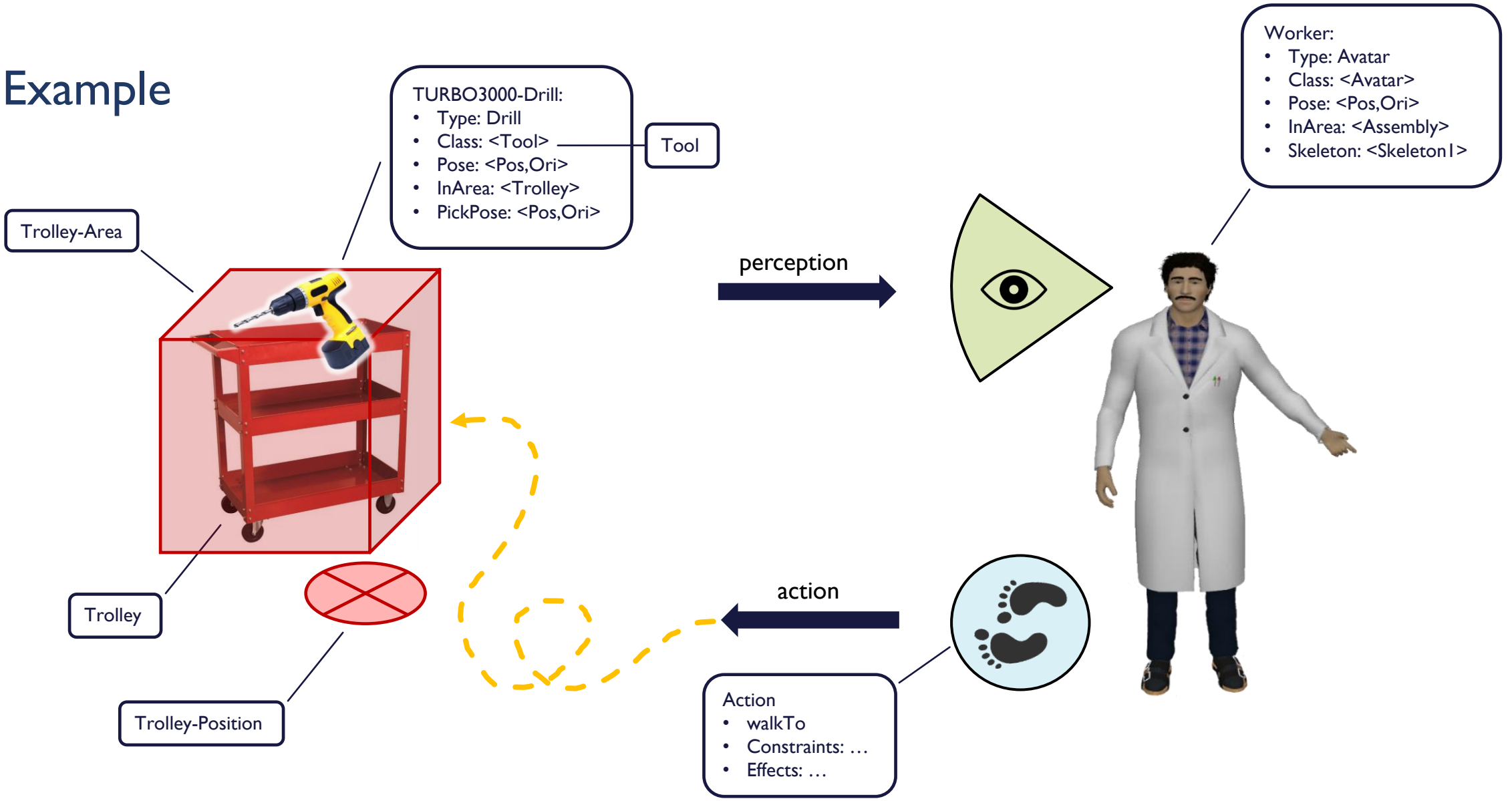
Behavior Trees





BT = HTN (DT) + FSM

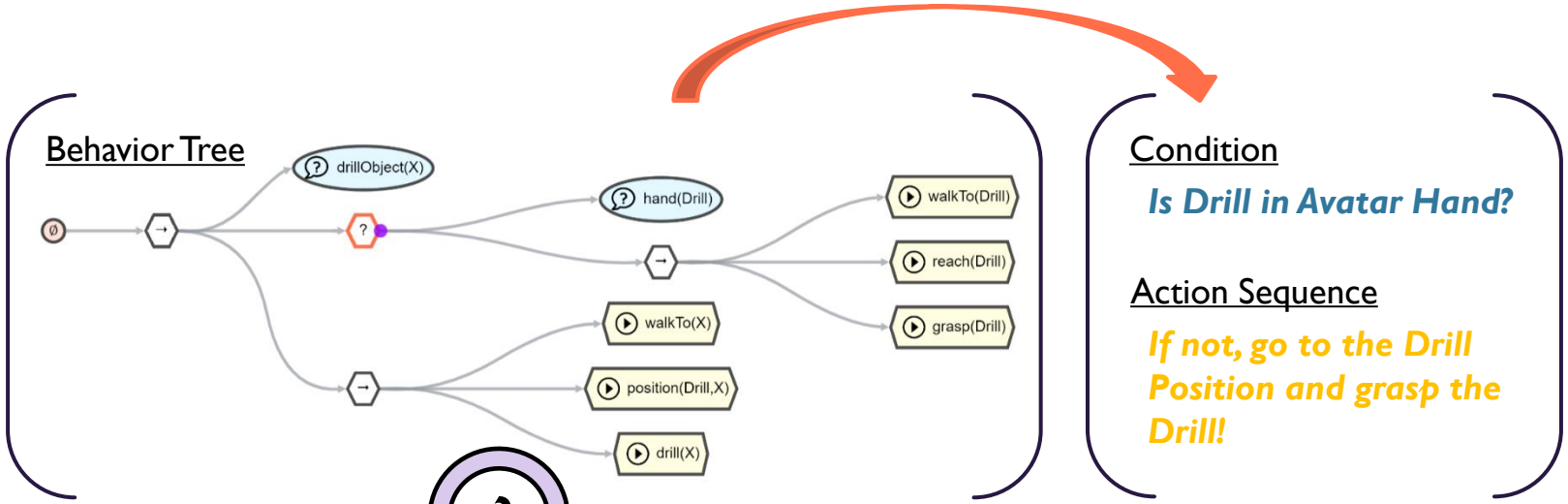
Example


TURBO3000-Drill:
• Type: Drill
• Class: <Tool>
• Pose: <Pos,Ori>
• InArea: <Trolley>
• PickPose: <Pos,Ori>
Tool
Worker:
• Type: Avatar
• Class: <Avatar>
• Pose: <Pos,Ori>
• InArea: <Assembly>
• Skeleton: <Skeleton1>
Trolley-Area
perception
Trolley
Trolley-Position
action
Action
• walkTo
• Constraints: …
• Effects: …

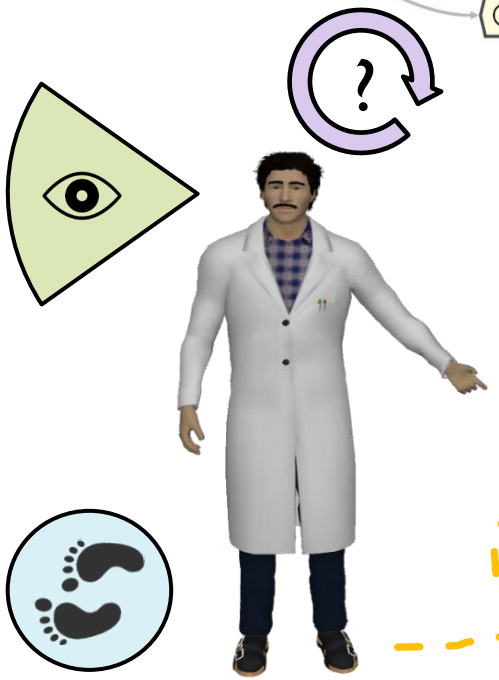
Example

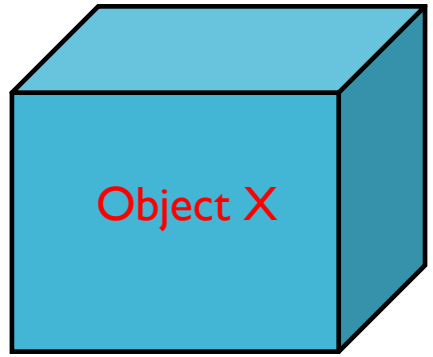
Event

Drill Object X



perception





walkTo(Drill)

action

walkTo(X) … drill(X)

Combination of MMUs in a Co-Simulation Component

- Co-Simulator:
  - Same interface as MMU
  - Central simulation components
  - Forwards Instructions to different MMUs
  - Combines poses of different MMUs