Supplementary Document

No Institute Given

Conversation. A conversation is a simple behavioral construct that effectively illustrates the general, modular, and extensible nature of parameterized behavior trees to author and simulate multi-actor interactions. Figure 1 illustrates the behavior trees used to author a multi-actor conversation. A conversation takes as parameters one or more actors that are participating in the conversation, and a destination point where the conversation must take place. The actors are first instructed to independently arrive at the meeting point. The `FocusOn` subtree instructs one actor (the subject) to perform a series of gestures while the other actors are instructed to look at the subject for its duration. The `DoConversation` subtree randomly instantiates a sequence of `FocusOn` behaviors between different combinations of actors to indicate an ongoing conversation between multiple actors. Conversations are instantiated between the villagers roaming the marketplace and are easily modified to indicate different conversation styles such as an argument, for example.

Robot Behaviors. We author a variety of robot behaviors which can be triggered due to commands issued by the player-controlled team leader, or due to other events in the environment which the robot may respond to. Figures 2 (a) and (b) illustrate simple behaviors instructing a robot to approach or follow the team leader. Figure 3 shows the behavior tree for issuing the `Follow` command, and Figure 4 illustrates a variation where the robot is instructed to follow a target (e.g., a suspicious villager). Figure 5 illustrates the command instructing the robot to pickup and object of interest and bring it back to the commander.

Villager response to Soldier-Robot Team. Figure 6 illustrates the behavior to elicit appropriate responses in the villagers to the presence of the soldier-robot team. When the soldier-robot team is not nearby, the pedestrian executes its default behavior tree or the current event that it is currently participating in (e.g., a conversation with other villagers). The presence of the robot or soldiers triggers a race condition causing the villager to either look at the passing by team, cautiously avoid them by keeping a safe distance, follow the robot out of curiosity, or even happily wave to them as they pass by. Specialized responses to the SR team can be easily defined for specific villagers who may be involved in the narrative of a mission.

Exchange Items / Shake Hands. If a haggle conversation between a vendor and the villager is successful, the vendor exchanges goods with the villager, performed using the `ExchangeItem` behavior (Figure 7). Both actors are first instructed to arrive at a common meeting point. When both actors arrive, they

are both instructed to reach for the other actors hands. When the reach is successful, the goods exchange hands and the behavior is complete. A variation of this behavior can also be used to demonstrate two actors shaking hands at the end of a polite conversation.

Wander. A simple wander behavior can be implemented using a smart waypoint system where villagers repeatedly wander to different points of interest in the marketplace, as shown in Figure 8. The waypoint system can be augmented to return specific destinations. For example, we can modify the wander behavior to create a `CrowdDispere` event where multiple villagers are instructed to disperse to locations outside the marketplace.

Chase. Figure 9 illustarates a `Chase` event where the robot is instructed to chase down a villager, who may be involved in some suspicious activity, such as the exchange of illegal contraband.
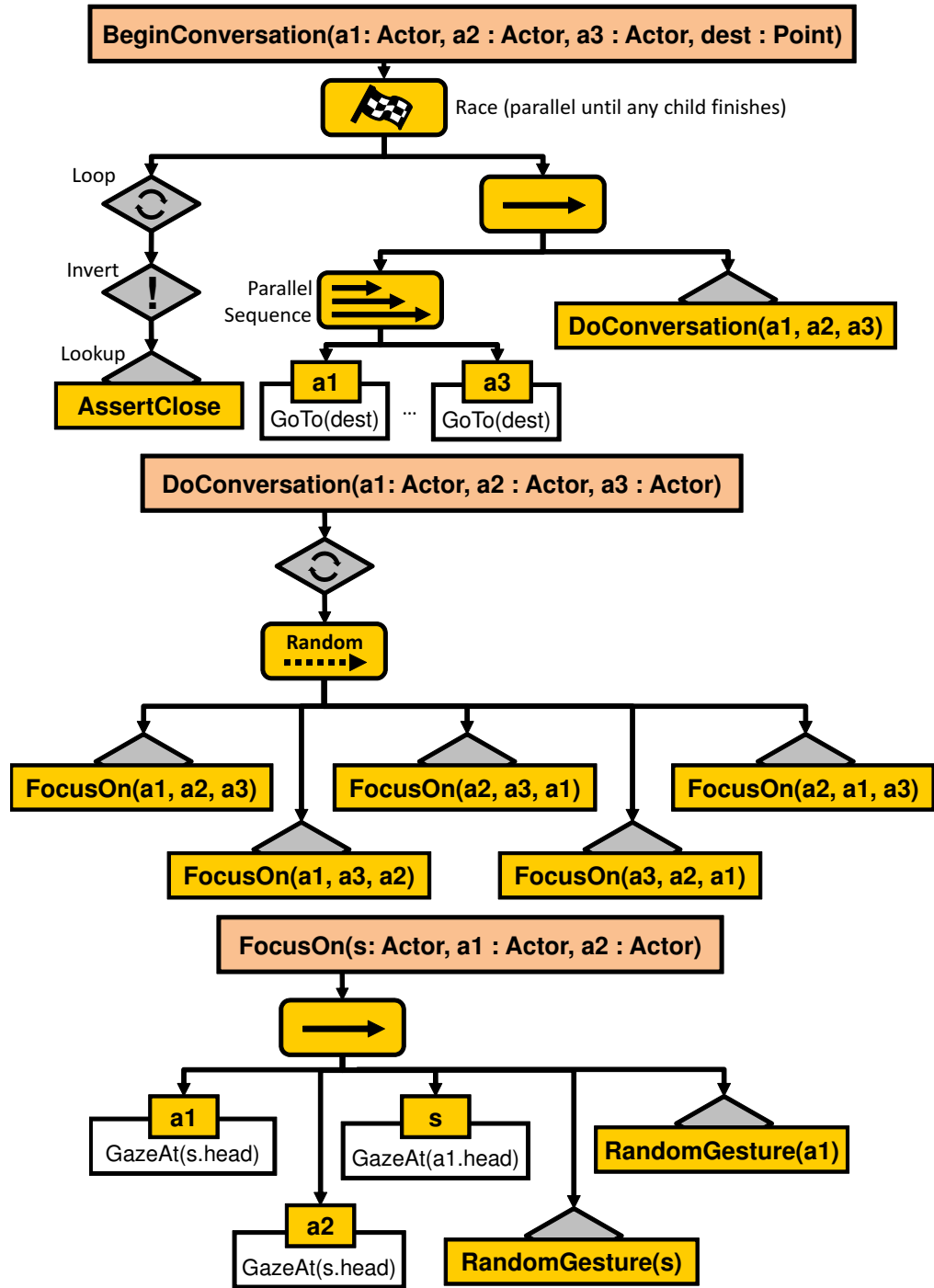
BeginConversation(a1 : Actor, a2 : Actor, a3 : Actor, dest : Point)

Race (parallel until any child finishes)

Loop

Invert

Lookup

AssertClose

Parallel Sequence

a1
GoTo(dest)

...

a3
GoTo(dest)

DoConversation(a1, a2, a3)

DoConversation(a1 : Actor, a2 : Actor, a3 : Actor)

Random

FocusOn(a1, a2, a3)

FocusOn(a2, a3, a1)

FocusOn(a2, a1, a3)

FocusOn(a1, a3, a2)

FocusOn(a3, a2, a1)

FocusOn(s : Actor, a1 : Actor, a2 : Actor)

a1
GazeAt(s.head)

s
GazeAt(a1.head)

RandomGesture(a1)

a2
GazeAt(s.head)

RandomGesture(s)

Fig. 1: Example PBT's to author and simulate a multi-actor conversation.
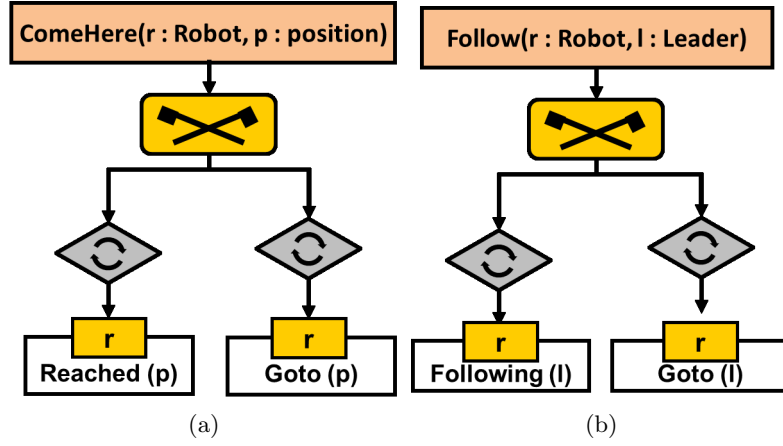
Fig. 2: Robot behaviors. (a) A behavior instructing the robot to come to the commander's location. (b) A behavior instructing the robot to follow the commander.
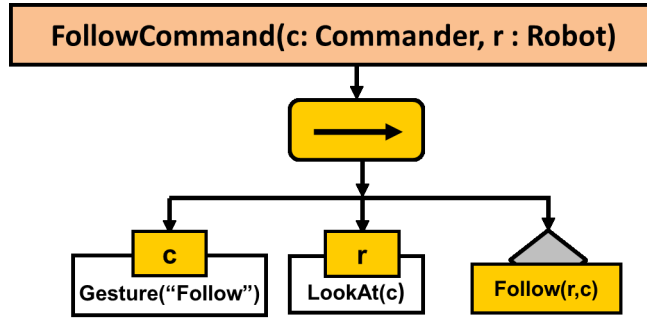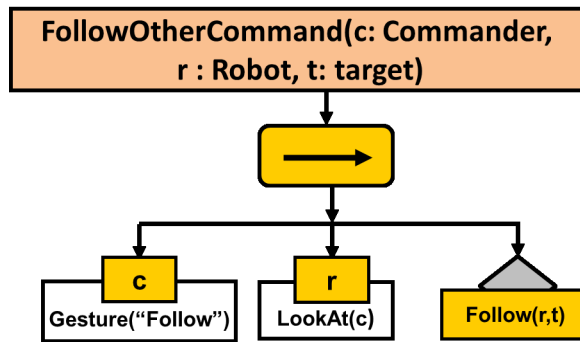

Fig. 3: Follow command behavior.
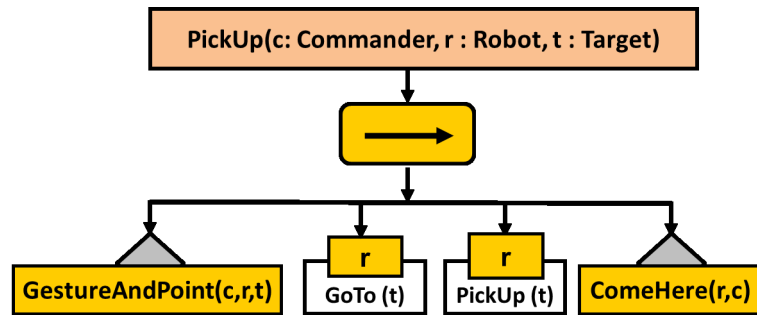

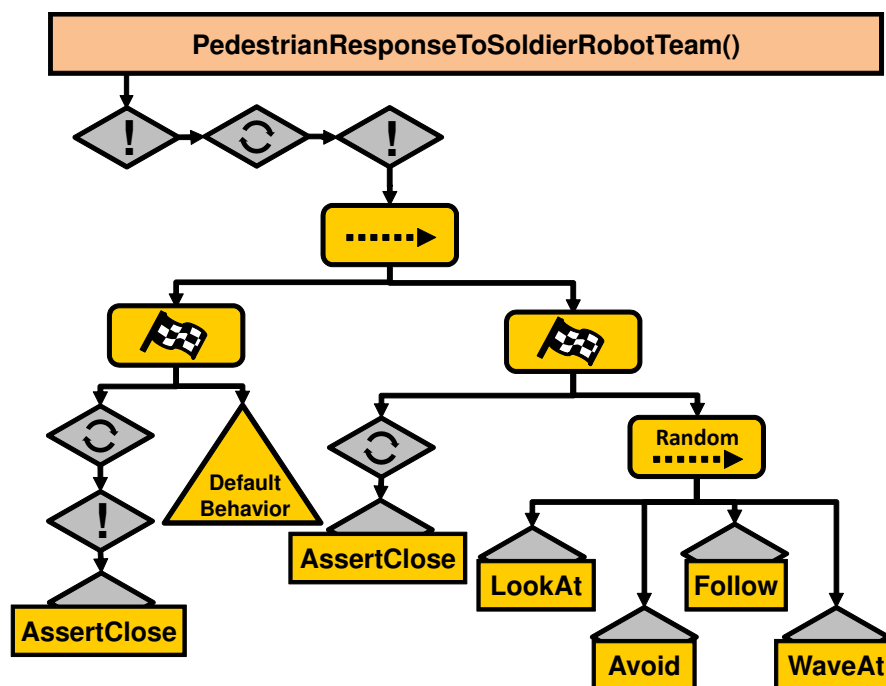Fig. 4: Follow target command.

Fig. 5: Robot pickup command.

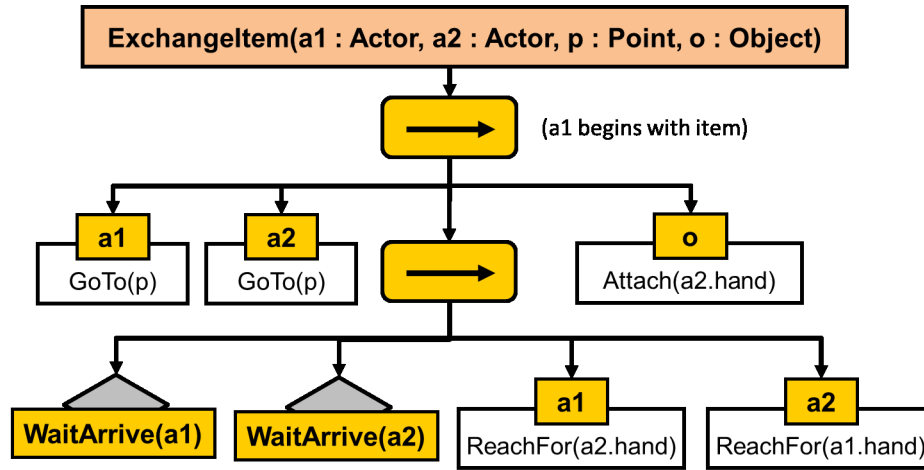Fig. 6: Villager response to soldier-robot team.
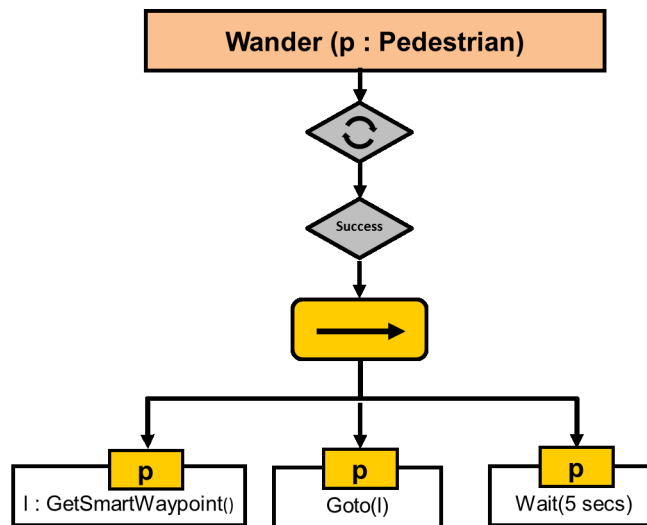
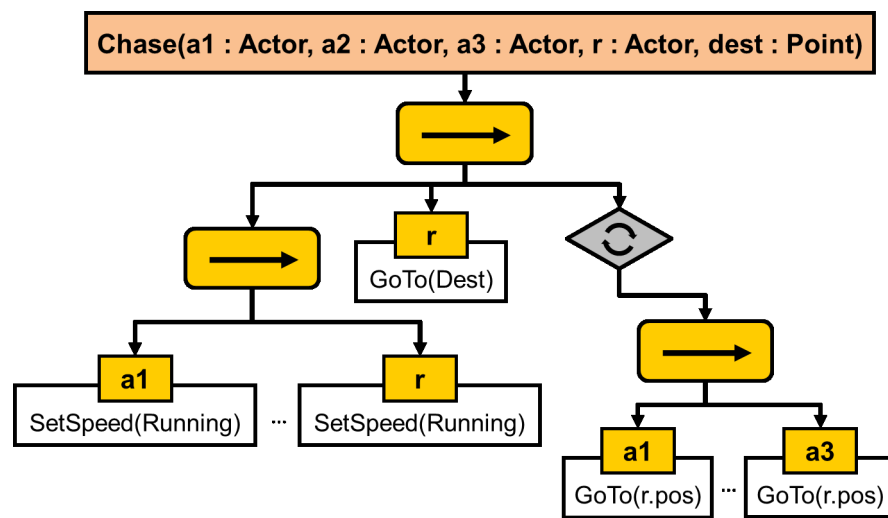Fig. 7: A behavior tree to exchange items between two actors.

Fig. 8: Wander behavior.

Fig. 9: A chase event.