

Can you show me the room?

Taking the utterance in isolation and eliminating the question: *Show me the room.*

From our perspective I see three possibilities:

1. *Show me the room.* (video feed)
2. *Show me the room on the map.*
3. *Show me to the room.*

Taking each in turn...

1. *Show me the room.*

Field	Value	Comment
name	show	
agent	jr	This would have been added to his queue
participants	commander and room obj0 = commander obj1 = room	
applicability conditions	<i>have</i> (agent, Camera)	Note that if obj0 and obj1 are not found in the environment, the PAR would have failed.
preparatory specification	<i>in</i> (agent, obj1)-> <i>locomote</i> (agent, obj1) and <i>connection</i> (Video, agent, obj0) -> <i>connect</i> (agent, Video, obj0)	If the agent isn't in the room, then go to the room. If there isn't a video connection, establish one. Note: objects including agents have a <i>location</i> field. <i>in</i> is implemented by checking to see if the location of agent is obj1.
termination conditions	<i>covered</i> (agent, obj1)	Note: I'm not entirely sure of how to check to see if the agent covered the entire room when sending the video feed. I imagine the motion generator (or in this case possibly LTL) will handle it.
post assertions	<i>seen</i> (obj0, obj1)	Note: this might change depending on the scenario.
during conditions	<i>in</i> (agent, obj1) and <i>videoing</i> (agent, obj1) and <i>sending</i> (agent, Video, obj0) and <i>seeing</i> (obj0, Video)	We haven't used during conditions very often.
purpose	<i>see</i> (obj0, obj1)	Purpose is most useful for replanning. If the commander needs to see the room and a video link can't be established, jr might go and lead him to the room or just send images.
subactions	<i>turnOn</i> (agent, VideoCamera) then	These are all also PARs.

	scan(agent, obj1) then turnoff(agent, VideoCamera)	If the camera is already on, that action will just return. We may not turn the camera off, I'm not sure.
manner	quickly, carefully, fully, etc	We could add these modifiers

2. *Show me the room on the map.*

Field	Value	Comment
name	show	
agent	jr	This would have been added to his queue
participants	commander, room, and map obj0 = commander obj1 = room obj2 = map	
applicability conditions	<i>have</i> (agent, obj2) and <i>have</i> (obj0, obj2)	Note that if obj0, obj1, and obj2 are not found in the environment, the PAR would have failed. This is weird. The map isn't a physical thing, so if they don't have the map they can't go and get it? They need to have the same map built in?
preparatory specification	<i>know</i> (agent, position(obj1)) -> <i>searchFor</i> (obj1)	If the agent doesn't know where the room is, then they have to search for it. Note: objects including agents have a position field, which is different than location. Position is the 3D coordinates of the object, and location is the surrounding PAR object. We currently don't have a memory model, so <i>know</i> is a bit odd. We kind of assume that agents are omniscient for the most part.
termination conditions	duration?	Once the room is shown on the map, it should be left there for a certain time? Perhaps it doesn't terminate until the commander sends another command?
post assertions	<i>know</i> (obj0, position(obj1))	commander knows the position of the room?
during conditions	<i>shownOn</i> (obj1, obj2)	The room is on the map

purpose	<i>know(obj0, position(obj1))</i>	
subactions	add(agent, obj1, obj2)	Jr. will add the room to the map. Might also send the map to the commander, but here I assumed the map is always accessible to both.
manner	detailed, general, etc	We could add these modifiers

3. Show me to the room.

Field	Value	Comment
name	show	
agent	jr	This would have been added to his queue
participants	commander, room, and map obj0 = commander obj1 = room	
applicability conditions	?	Note that if obj0 and obj1 are not found in the environment, the PAR would have failed.
preparatory specification	<i>know(agent, position(obj1)) -> searchFor(obj1)</i> and <i>with(agent, obj0) -> locomote(agent, obj0)</i>	If jr isn't with the commander, then he has to go to the commander. This might result in him searching for the commander.
termination conditions	<i>in(obj0, obj1)</i>	The commander is in the room.
post assertions	<i>in(obj0, obj1), know(agent, position(obj0)), in(agent, obj0)</i>	
during conditions	<i>with(agent, obj0)</i> and <i>with(obj0, agent)</i>	jr and the commander are together The various positions, locations, etc are also updated as they move through the environment.
purpose	<i>in(obj0, obj1)</i>	the commander is in the room
subactions	follow(obj0, agent) while locomote(agent, obj1)	While jr is locomoting to the room, the commander should follow.
manner	quickly, directly, carefully, avoiding x,	We could add these modifiers

The predicates (in italics) need to be evaluated in the preparatory specifications and termination conditions. For example, *in(obj0, obj1)* checks to see if obj0 is currently located in obj1. Predicates in post assertions and during conditions, are true during those times.