Jr, I want you to search the room you are in first and when you go to leave head for the North end of the building, continue around East and we will meet up at the end of a long hallway. Also, don't forget to notify me the minute you see a bomb.

To me there is an implicit action of searching rooms for bombs, but that isn't clear from this command alone. Further information specifies the route that should be taken (where Jr. is first, then toward the North end of the building, then around East). After the search action is completed, Jr. is to meet the commander at the end of a long hallway.
So starting with the *search* action. There would be a search PAR. The question is how to include the necessary arguments. I would propose sequential *searches* with the areas as arguments:
searchFor(Jr., location(Jr.), Bombs) //search the obj argument
searchForTo(Jr., north(building), Bombs) // search from the current location to the obj argument. I guess north would be function that returns the northern most room of the building?
searchForTo(Jr., east(building), Bombs)
meet(Jr., commander, end(Hallway.property = long)) // general Hallway object?

If we were animating this in a virtual environment, it would just be searching the environment for bombs with the path to be searched explicitly in the path field of the PAR. This path would be generated by a routine that examines the building and parameters and generates a likely path. Here I don't think we can assume prior knowledge of the environment.

Need a way to send through a generic object with various properties.
Big question: Where and when do ambiguities about the environment get resolved?

1. *Search the room you are in for bombs.*

Field	**Value**	**Comment**
name	searchFor	
agent	jr	This would have been added to his queue
participants	obj0 = location(Jr.) obj1 = Bombs	The location function returns the room that Jr. is in when the function is called. Bombs would need to be the generic parent of bombs in the object hierarchy. I will have to make sure that the code doesn't crash when the object isn't instantiated.
applicability conditions		Basically if Jr. can move and sense, he should be able to search? He also has to have a representation of bombs and need to recognize them.
preparatory specification		Jr. is already in the room, so he doesn't need to go to it.
termination conditions	*covered*(agent, obj0)	Note: I'm not entirely sure of how to check to see if the agent covered the entire room when searching for items. I assume this will come

		from LTL.
post assertions	*seen*(agent, obj0)	Note: this would be a part of the memory model that Jr. observed this room at this time. We would assume that Jr. has perfect memory.
during conditions	*in*(agent, obj0)	We haven't used during conditions very often.
purpose	*find*(agent, obj0)	
subactions	notify(agent, commander, obj1) while scan(agent, obj0)	These are all also PARs. I'm not sure of what to do about notifying. **Notifying** might be a separate action implemented as a standing order. The current PAR system isn't set up to do this, but could be.
manner	quickly, carefully, fully, etc	We could add these modifiers

2. *Search for bombs to the North end of the building.*.

Field	**Value**	**Comment**
name	searchForTo	
agent	jr	This would have been added to his queue
participants	obj0 = Bombs obj1 = north(building)	Bombs would need to be the generic parent of bombs in the object hierarchy. I will have to make sure that the code doesn't crash when the object isn't instantiated. north would return the northern most room of the building?
applicability conditions		Basically if Jr. can move and sense, he should be able to search? He also has to have a representation of bombs and need to recognize them.
preparatory specification		Assumes Jr searches from his current location.
termination conditions	*covered*(agent, obj1)	Note: I'm not entirely sure of how to check to see if the agent covered the entire room when searching for items. I assume this will come from LTL.

post assertions	*seen*(agent, obj1)	Note: this would be a part of the memory model that Jr. observed this room at this time. We would assume that Jr. has perfect memory.
during conditions	*in*(agent, locations)	Jr.'s location would change as he entered rooms and hallways during the search
purpose	*find*(agent, obj0)	
subactions	notify(agent, commander, obj0) while searching	These are all also PARs. I'm not sure of what to do about notifying. **Notifying** might be a separate action implemented as a standing order. The current PAR system isn't set up to do this, but could be.
manner	quickly, carefully, fully, etc	We could add these modifiers

3. *Search for bombs to the East side of the building..*

Field	**Value**	**Comment**
name	searchForTo	
agent	jr	This would have been added to his queue
participants	obj0 = Bombs obj1 = east(building)	Bombs would need to be the generic parent of bombs in the object hierarchy. I will have to make sure that the code doesn't crash when the object isn't instantiated. east would return the eastern most room of the building?
applicability conditions		Basically if Jr. can move and sense, he should be able to search? He also has to have a representation of bombs and need to recognize them.
preparatory specification		Assumes Jr searches from his current location.
termination conditions	*covered*(agent, obj1)	Note: I'm not entirely sure of how to check to see if the agent covered the entire room when searching for items. I

		assume this will come from LTL.
post assertions	*seen*(agent, obj1)	Note: this would be a part of the memory model that Jr. observed this room at this time. We would assume that Jr. has perfect memory.
during conditions	*in*(agent, locations)	Jr.'s location would change as he entered rooms and hallways during the search
purpose	*find*(agent, obj0)	
subactions	notify(agent, commander, obj0) while searching	These are all also PARs. I'm not sure of what to do about notifying. **Notifying** might be a separate action implemented as a standing order. The current PAR system isn't set up to do this, but could be.
manner	quickly, carefully, fully, etc	We could add these modifiers

4. *Meet the commander at the end of a long hallway.*

Field	**Value**	**Comment**
name	meet	
agent	jr	This would have been added to his queue
participants	obj0 = commander obj1 = end(Hallway) where the Hallway PAR has a property length which is either numerically specified or given a qualitative labeling for *long*.	When animating, Hallway would be disambiguated when the command was given, but for our purposes, that probably isn't the case. We will need a way to represent unspecified objects with properties. end function would return the end of the hallway. **Unfortunately, hallways have two ends, so how do we know which end?**
applicability conditions		Assumes Jr. can locomote and sense
preparatory specification		We make sure that Jr. is at the end of the hallway as a prep. spec, but it might result in behavior where he passes the commander in the

		hallway on the way to the end and just keeps going.
termination conditions	*with*(agent, obj0)	with could be implemented as a distance function
post assertions		Because of the termination condition, you can't really assume that Jr. is at the end of the hallway. If he meets the commander on the way to the end of the hallway, he will end the meet action before he gets to the end.
during conditions	*in*(agent, Hallway)	
purpose	*with*(agent, obj0)	
subactions	locomote(agent, obj1)	This is really the meat of the action.
manner	quickly, carefully, fully, etc	We could add these modifiers

Mark that room as clear and make certain to check all the rooms in the hall.

This is two sequential actions. The first sets a property of a room. One question is which room is being referred to; the room that Jr. is currently in or a room that is being discussed. From the dialogue it is the room that Jr. was just in, which means that we have to have a model of where he was. Here I think the room could be disambiguated at the time the command is issued.
The next action is to make certain to check all the rooms in the hall. He already has a command to search rooms to the north. This may or may not include all of the rooms in the hall where he currently is. If we can assume that we know the environment, then we can generate a path that will visit all of the rooms. If we can't, then the information needs to be passed to LTL. It would likely be translated into a search as specified about. I'm not sure of how to specify ***all the rooms in the hall***.

Wait. Shouldn't you be going into the hallway? I told you to go there.

The only command here is to wait which would result in pausing the execution of all actions and suspending the action queue operation, perhaps even emptying the action queue.

Jr, go back to the first room and wait for me there, I think there is something hidden that you missed.

Here we might only care about the going back to the first room and waiting. I'm not sure if the reason would be represented as a purpose for the action?

5. *Go back to the first room and wait for me there.*

Field	**Value**	**Comment**
name	wait	
agent	jr	This would have been added to his queue
participants	obj0 = Commander obj1 = Room	Jr is to wait for the commander. The room referenced will have to

		be obtained from Jr.'s memory.
applicability conditions		Assumes Jr. can locomote and sense
preparatory specification	in(agent, obj1)->locomote(action, obj1)	If the agent isn't in the room, he should go there.
termination conditions	*with*(agent, obj0)	with could be implemented as a distance function
post assertions	in(agent, obj1)	Here we can assume that Jr. is in the room when he meets the commander.
during conditions		
purpose	*with*(agent, obj0)	
subactions		Given the prep. spec. There isn't really an action.
manner	quickly, carefully, fully, etc	We could add these modifiers