Vars not used from table !

visionObstacles : IP Pos

safeLocation : (x,y), x ∈ ℕ, y ∈ ℕ*

return2NormalOperation : bool

prioritisedLocus : IP Pos

Chargers : IP Pos

invalidMap : Message

noMoreViablePlans : Message

failureMode Invoked : Message

noPlan : Bool

systemState : (failure, noPlan, messages)

plan2C : seq Pos

plan2O : seq Pos

recheck : Bool

communicationData : (messages, data followed, from Who) ← used this

GSObstacles : IP Pos

failed2Reconfigure : Bool

completed : Message

identifiedFault : Message

Command : Message

Deliverables:

[Scenario1, Scenario2,
SC1, SC2, SC3, SC4,
U2,
G6, G7,
HI1, HI2
CR3]

Assumptions:

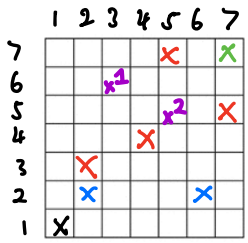
1. Side Rovers will never fail
2. Current Side rover will come
3. User provides a goal that is reachable
4. Charge to full when battery = 1
5. (Side rover and help position) is indicated as points on the grid.

Types

Pos == 0...7 × 0...7

Max_Battery == 5

Message is a String



RoverState

currentPosition	: Pos	
obstacles	: ℙ Pos	(V2)
failure	: Bool	
failureReboot	: ℙ Pos	(new variable)
failureHelp	: Pos ⇸ helperID	(new variable)
goal	: ℙ Pos	
batteryLevel	: ℕ	
chargingComplete	: Bool	
recharge	: Bool	
atGoal	: Bool	
dataCollected	: Message	(part of communication data)
helperID	: ℕ	

(failureReboot ⊆ obstacles) ∧ (dom(failureHelp) ⊆ obstacles)

∧ (failureReboot ∩ dom(failureHelp) = ∅)

batteryLevel ∈ 1..Max_Battery (SC1)

chargingComplete <=> batteryLevel = Max_Battery

currentPosition ∉ obstacles (SC4)

goal ∉ obstacles (SC2) (V2)

helperID ∈ {0,1,2}, 0 means no help (or main rover ID) (SC3)

dom(failureHelp) ⊆ Pos

ran(failureHelp) ⊆ helperID

RoverStore INIT

currentPosition	:	(1,1)
obstacles	:	{(2,3), (4,4), (5,7), (7,5)}
failure	:	False
failureReboot	:	{(2,2), (6,2)}
failureHelp	:	{(2,6) ⟼ 1, (5,5) ⟼ 2}
goal	:	(7,7)
batteryLevel	:	5
chargingComplete	:	True
recharge	:	False
atGoal	:	False
dataCollected	:	"This is mock data"
helperID	:	0

This means at (5,5) the rover with id of 2 will come to help

N.B is variable not stated, implicitly means retain original value

Scenario 1

move

ΔRoverState
next? : Pos

failure = False
recharge = False (H12)
batteryLevel > 1 (SL1)
next? ∉ obstacles (SL2) (SL4)

currentPosition' = next?
batteryLevel' = batteryLevel − 1
atGoal' = True ⟷ (currentPosition' = goal)

notifyComplete (G6)

Ξ RoverState
dataEmit! : Message

atGoal = True
dataEmit! = dataCollected

setRecharge (HL1)

ΔRoverState

failure = False
recharge = False
batteryLevel = 1

recharge' = True

charge (H22)

ΔRoverState

failure = False
recharge = True
batteryLevel < 5

batteryLevel' = batteryLevel + 1

finishCharge (H32)

ΔRoverState

failure = False
recharge = True
batteryLevel = 5

recharge' = False
chargingComplete = True

Scenario 2

triggerRebootFailure (SL3)

ΔRoverState

failure = False
currentPosition ∈ failureReboot

failure' = True (G7)
helperID' = 0

rebootRecover (SL3)

ΔRoverState

failure = True
currentPosition ∈ failureReboot

failure' = False
helperID' = 0

triggerHelpFailure (SL3) (CR2)

ΔRoverState

failure = False

(currPosition ∈ dom(failureHelp))

failure' = True (G7)

helperID = 0

requestHelp (SL3) (G7) (CR3)

=_RoverState

failure = True

(currPosition ∈ dom(failureHelp))

recieveHelperID (SL3) (CR3)

ΔRoverState

failure = True

(currPosition ∈ dom(failureHelp))

helperID' = failureHelp(currPosition)

authenticateAndRecover (SL3) (CR3)

ΔRoverState

arrivingRoverID? : ℕ

failure = True

helperID ≠ 0

arrivingRoverID = helperID

failure = False

helperID = 0