vector

neighbourDetected: nat

goalSeen

objectSeen: nat

Pusher

Ⓡ IMotionControl
Ⓡ INeighbourWatchOps
Ⓡ IGoalWatchOps
Ⓡ IObjectWatchOps
π close: nat, Ta: nat, Tb: nat, Tc: nat, Td: nat, Te: nat, S: nat
✖ neighbours: nat = 2
✖ distance: nat, d: nat, n: nat
✖ goal: boolean
⏱ T
⏱ N
⏱ C
ⓘ INeighbourWatch
ⓘ IGoalWatch
ⓘ IObjectWatch

IObjectWatchOps
- Ⓞ enableObjectWatch()
- Ⓞ disableObjectWatch()

IObjectWatch
- ⚡ objectSeen: nat

INeighbourWatchOps
- Ⓞ enableNeighbourDetection()
- Ⓞ disableNeighbourDetection()

IGoalWatchOps
- Ⓞ enableGoalFinding()
- Ⓞ disableGoalFinding()

IMotionControl
- Ⓞ searchObject()
- Ⓞ moveToObject()
- Ⓞ closeInOnObject()
- Ⓞ pushObject()
- Ⓞ moveAroundObject()
- Ⓞ scanAndAlign()
- Ⓞ evade()

IGoalWatch
- ⚡ goalSeen

INeighbourWatch
- ⚡ neighbourDetected: nat

Searching
entry enableObjectWatch()
during searchObject()

objectSeen?distance #T

[since(T)>=Ta]

MovingToObject
during moveToObject()

Watch

objectSeen?d #T[distance>=close/\since(T)<Ta]/distance = d

[sinceEntry(Evading)>=Te]

Evading
during evade()

[since(T)>=Ta]

[distance<close/\since(T)<Ta]

ClosingInOnObject
during closeInOnObject()

Watch

objectSeen?d #T[distance>0/\since(T)<Ta]/distance = d

[sinceEntry(MovingAround)>=Td]

MovingAround
during moveAroundObject()

[goal/\sinceEntry(Scanning)>=S]

[distance==0/\since(T)<Ta]/disableObjectWatch()

[distance>0/\since(C)>=Tc]

Pushing
during pushObject()
entry enableObjectWatch(); enableNeighbourDetection()
exit disableObjectWatch(); disableNeighbourDetection()

Watch

- objectSeen?d[distance==0/\d==0]
- objectSeen?d[distance==0/\d>0]/#C; distance = d
- objectSeen?d[distance>0/\d==0]/distance = d
- objectSeen?d[distance>0/\d>0]
- neighbourDetected?n[neighbours<2/\n<2]
- neighbourDetected?n[neighbours>=2/\n<2]/#N; neighbours = n
- neighbourDetected?n[neighbours<2/\n>=2]/neighbours = n
- neighbourDetected?n[neighbours>=2/\n>=2]

Scanning
entry goal = false; enableGoalFinding()
during scanAndAlign()
exit disableGoalFinding()

goalSeen/goal = true

Watch

[neighbours<2/\since(N)>=Tb]

[not goal/\sinceEntry(Scanning)>=S]/neighbours = 2; #C; #N