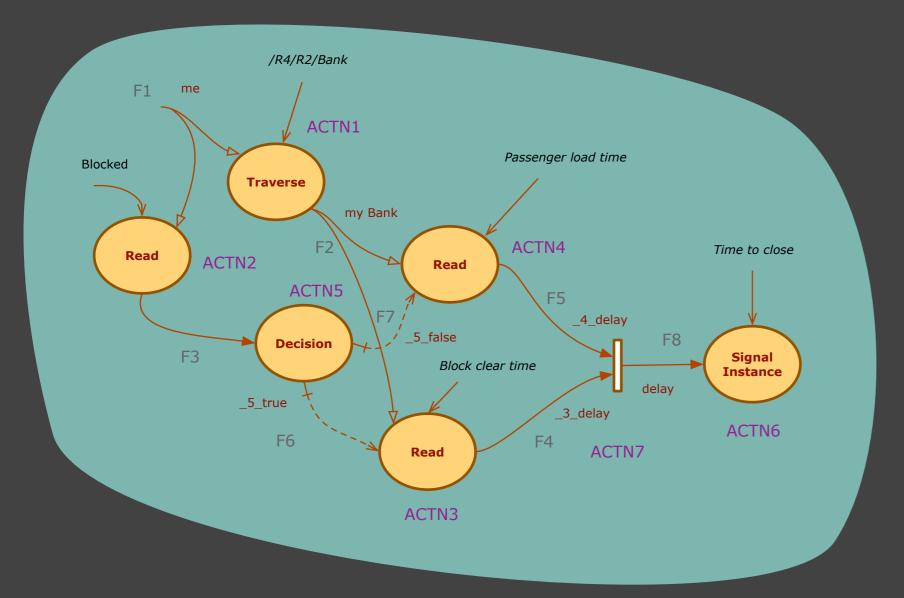
Door Data Flow Diagrams

Leon Starr 2025-10-14/ v0.8

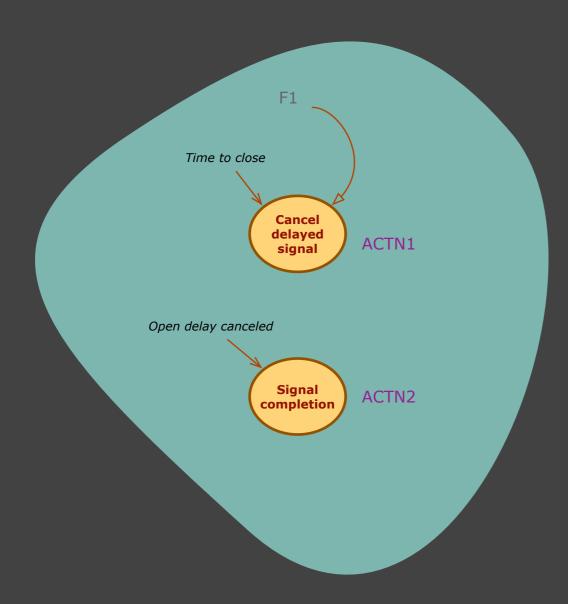
OPEN

my Bank .= /R4/R2/Bank
Blocked? delay = my Bank.Block clear time :
delay = my Bank.Passenger load time
Time to close -> me @delay

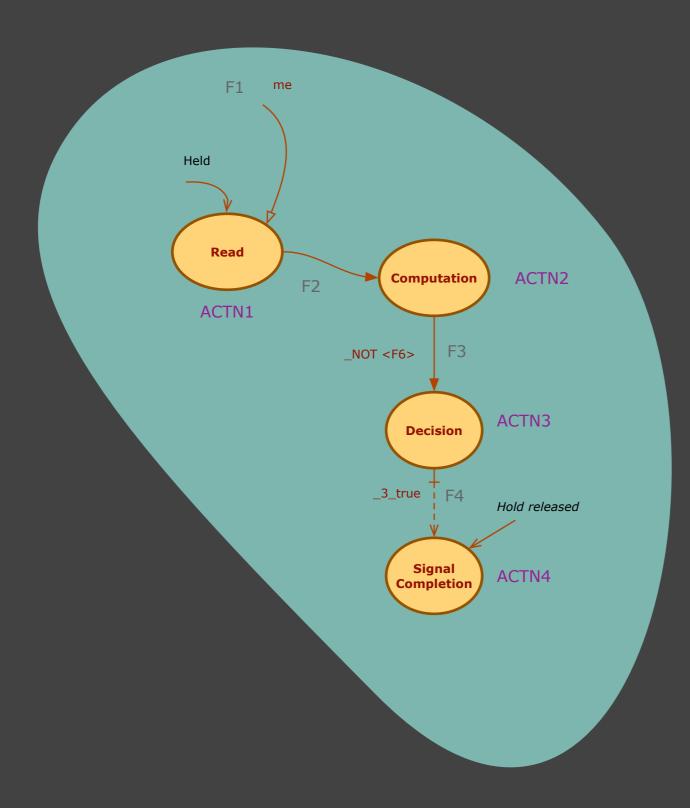


Cancel open delay

Time to close ->* me Open delay canceled -> me

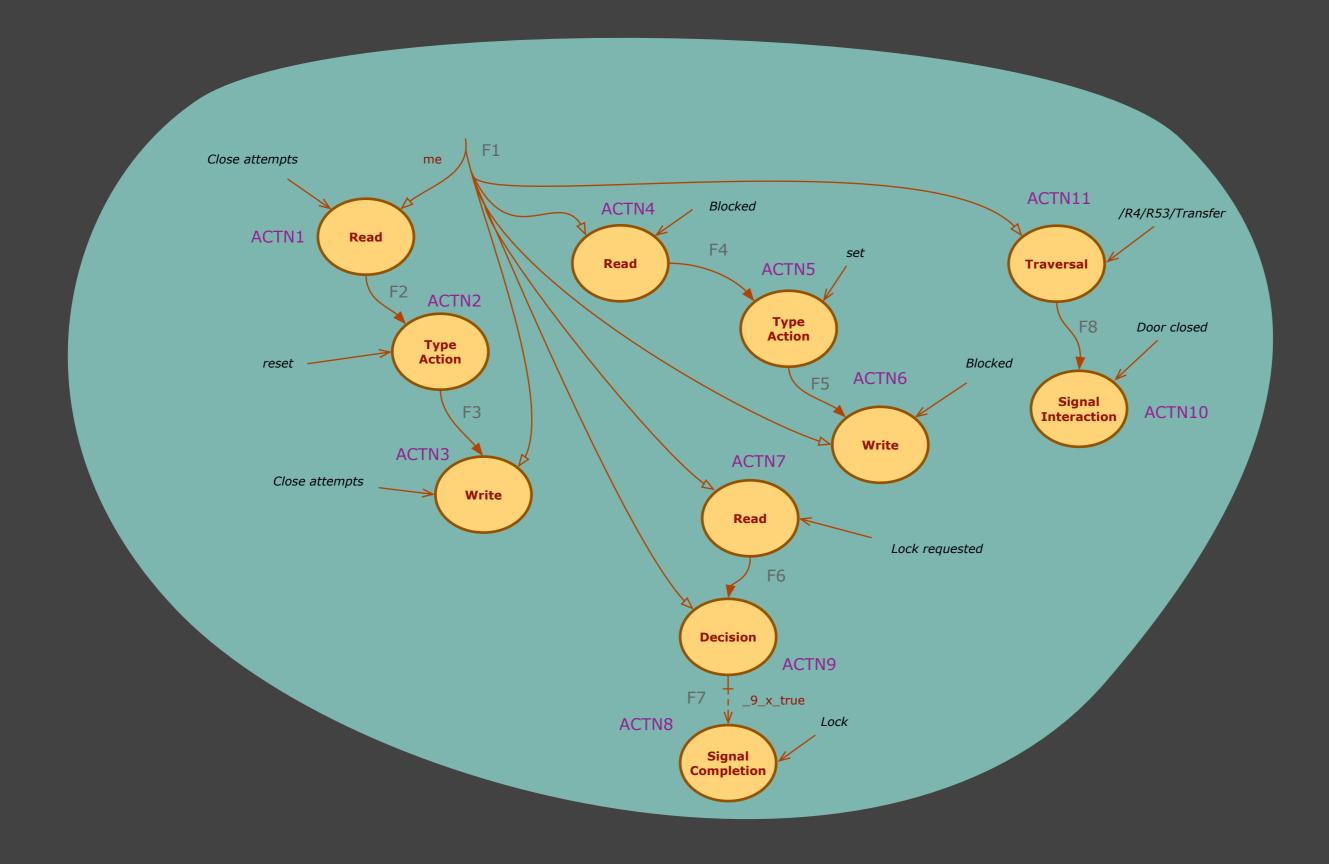


HOLDING OPEN !Held ? Hold released -> me



CLOSED

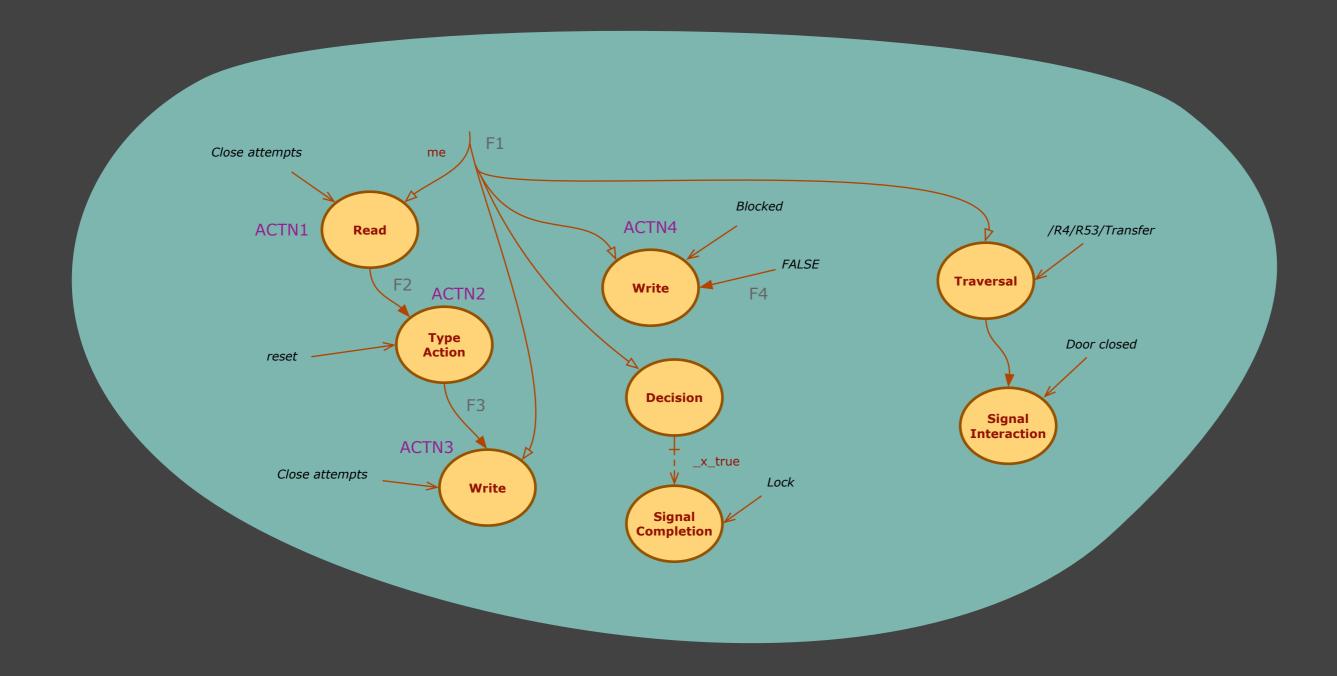
Close attempts.reset
Blocked.unset
Lock requested? Lock -> me
// May be None
Door closed -> /R4/R53/Transfer



CLOSED

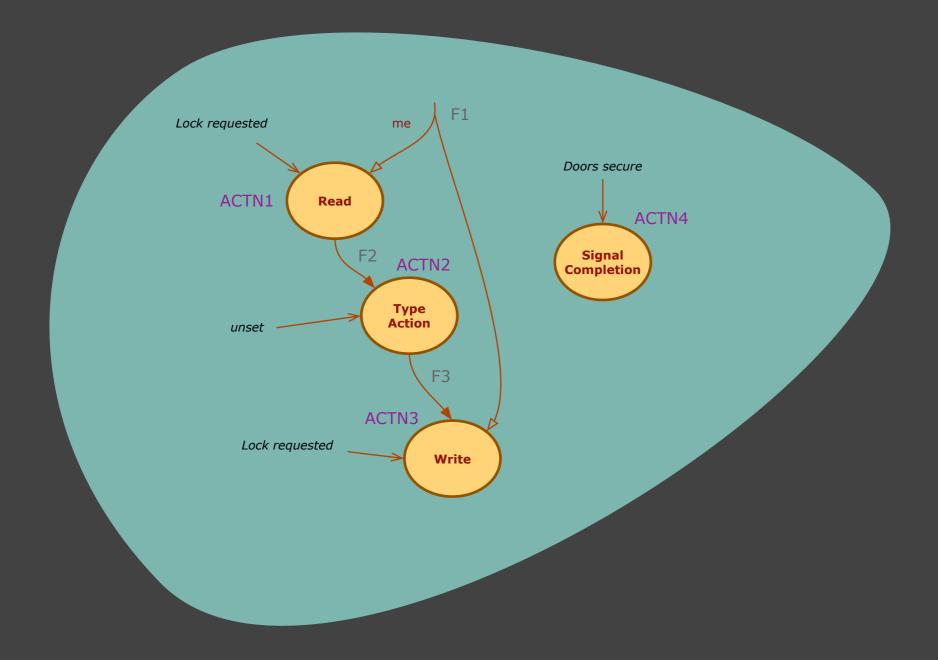
Close attempts.reset
Blocked = FALSE
Lock requested? Lock -> me
// May be None
Door closed -> /R4/R53/Transfer

Here we set Blocked to FALSE directly instead of using the boolean set type operation, which is preferred. So we save this alternate, less preferable data flow for reference. Either approach is fine, but I like to avoid explicit constants even though TRUE/FALSE are fairly safe as constants go.



LOCKED

// Tell the cabin that we're ready to go Lock requested.unset Doors secure -> /R4/Cabin



Count block

Close attempts.increment (Close attempts > /R4/R2/Bank.Max close attempts) ? Cannot close door -> me : {
Blocked.set

Keep trying -> me

