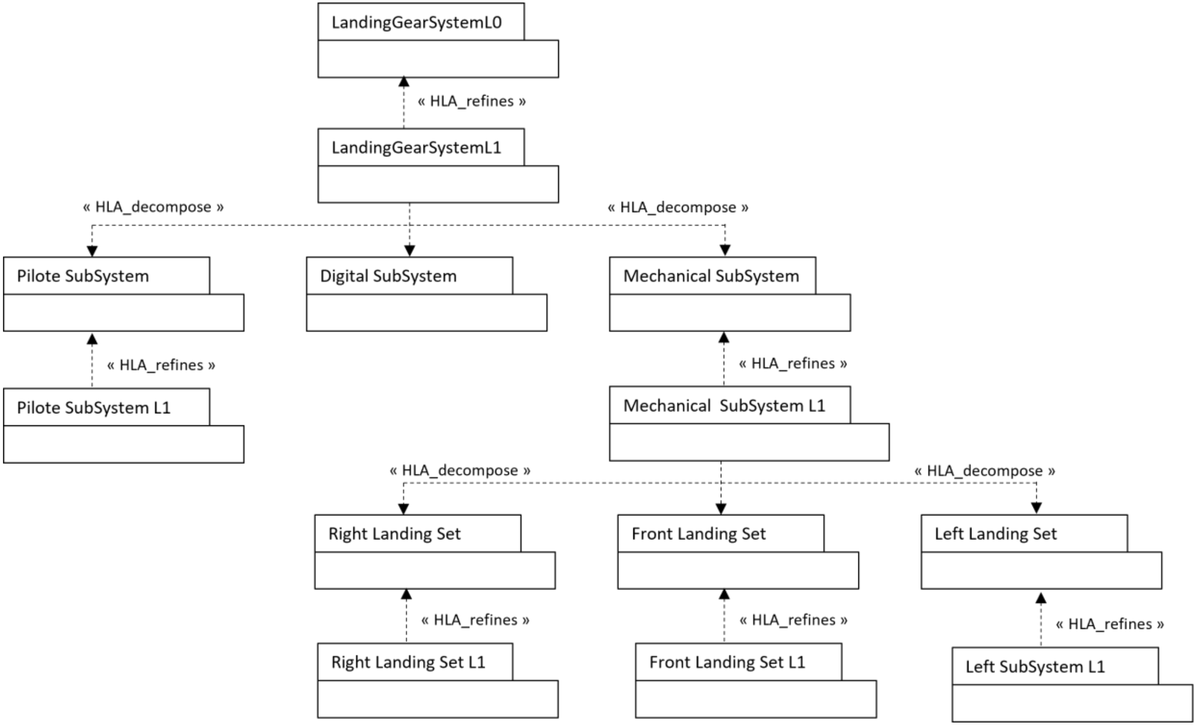
Table des matières

High-Level Architecture

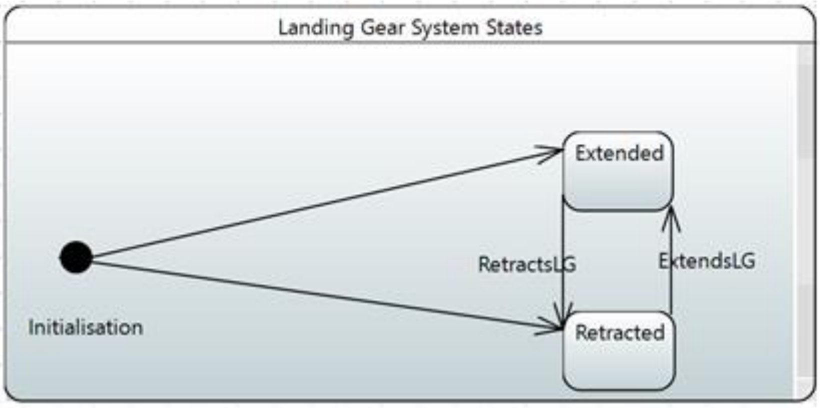
- Package Diagram :



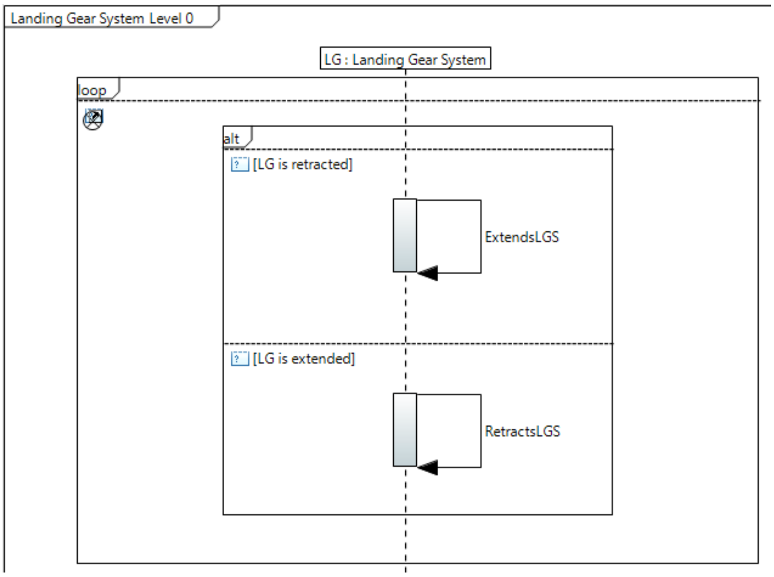
- LandingGearSystemL0 :

«Block»
Landing Gear System

Block Definition Diagram



State-Machine Diagram



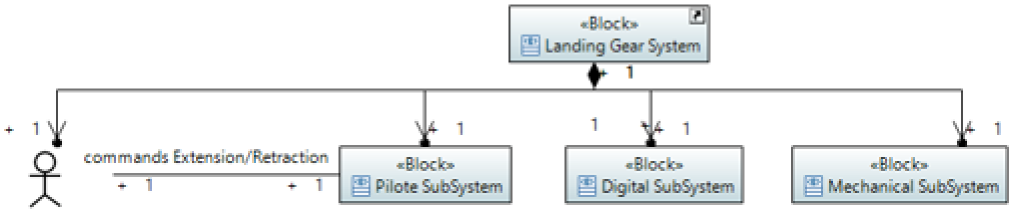
Sequence Diagram

- Event_B specification of LandingGearSystemL0:

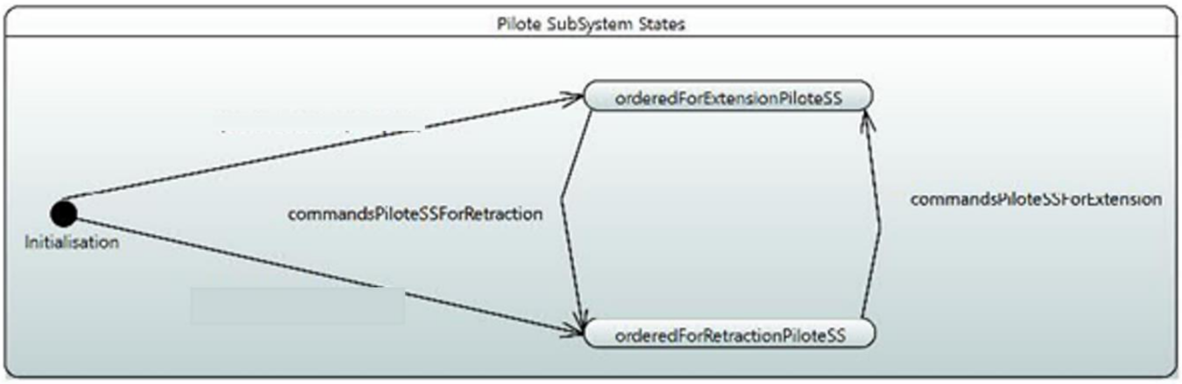
```
SYSTEM
          LandingGearSystemL0_CONT
SETS
          LandingGearSystem;
          LandingGearSystemStates
CONSTANTS
          lg,
          Extended,
          Retracted
PROPERTIES
          lg : LandingGearSystem &
          LandingGearSystem ={lg} &
          Retracted : LandingGearSystemStates &
          Extended : LandingGearSystemStates &
          Retracted /= Extended &
          LandingGearSystemStates ={Extended, Retracted}
END
```

```
SYSTEM
          LandingGearSystemL0
SEES
          LandingGearSystemL0_CONT
VARIABLES
          lgState
INVARIANT
          lgState : LandingGearSystem --> LandingGearSystemStates
INITIALISATION
          lgState :: {lg} -->LandingGearSystemStates
EVENTS
          RetractsLGS =
          SELECT
                    lgState(lg)=Extended
          THEN
          lgState(lg):=Retracted
          END;
          ExtendsLGS =
          SELECT
                    lgState(lg)=Retracted
          THEN
                    lgState(lg):=Extended
          END
END
```

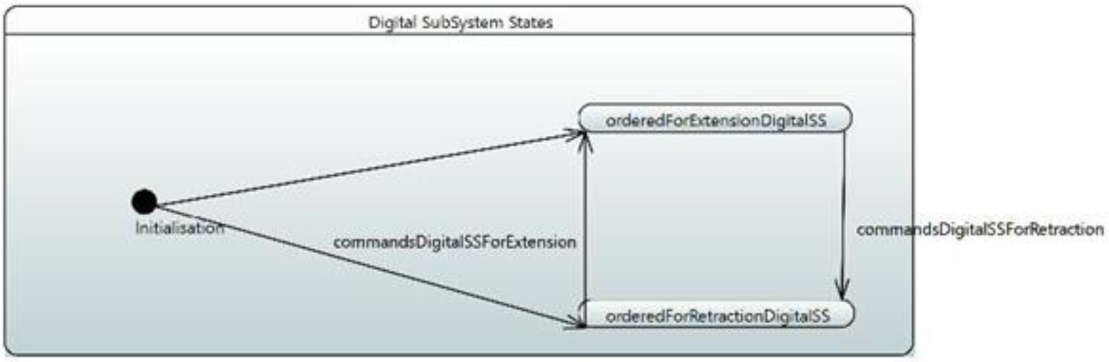
- LandingGearSystemL1:



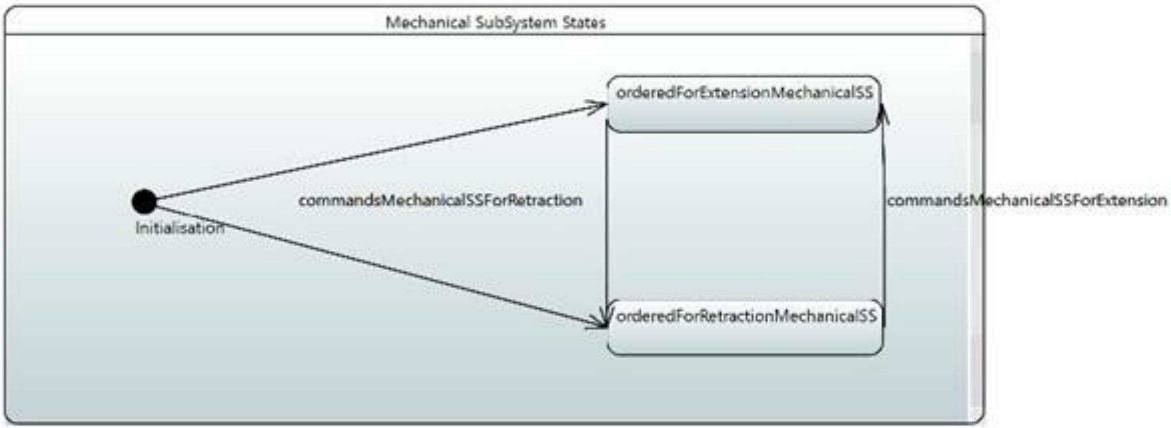
Block Definition Diagram


Pilote SubSystem States
orderedForExtensionPiloteSS
Initialisation
commandsPiloteSSForRetraction
commandsPiloteSSForExtension
orderedForRetractionPiloteSS

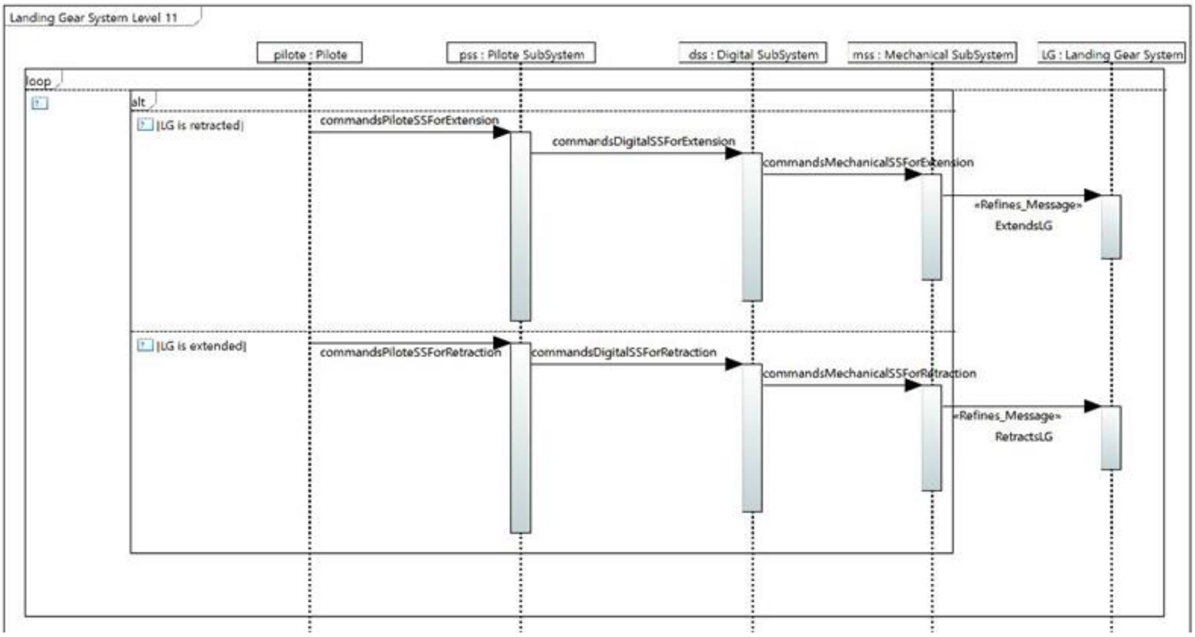
Pilote SubSystem state-machine Diagram


Digital SubSystem States
orderedForExtensionDigitalSS
Initialisation
commandsDigitalSSForExtension
commandsDigitalSSForRetraction
orderedForRetractionDigitalSS

Digital SubSystem state-machine Diagram


Mechanical SubSystem States
orderedForExtensionMechanicalSS
commandsMechanicalSSForRetraction
Initialisation
commandsMechanicalSSForExtension
orderedForRetractionMechanicalSS

Mechanical SubSystem state-machine Diagram


Landing Gear System Level 11
pilote : Pilote
pss : Pilote SubSystem
dss : Digital SubSystem
mss : Mechanical SubSystem
LG : Landing Gear System
loop
alt
[LG is retracted]
commandsPiloteSSForExtension
commandsDigitalSSForExtension
commandsMechanicalSSForExtension
«Refines_Message»
ExtendsLG
[LG is extended]
commandsPiloteSSForRetraction
commandsDigitalSSForRetraction
commandsMechanicalSSForRetraction
«Refines_Message»
RetractsLG

Sequence Diagram of LandingGearSystemL1

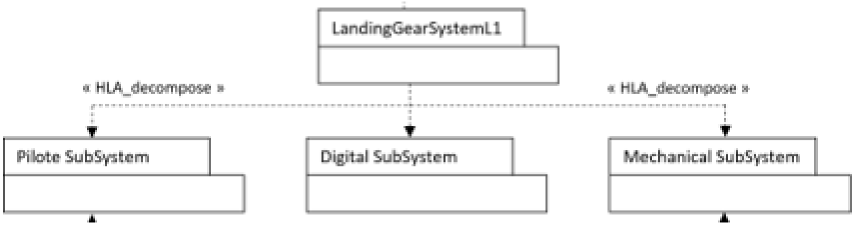
- Event_B specification of LandingGearSystemL1:

```
SYSTEM
        LandingGearSystemL1_CONT
SETS
        DigitalSubSystem;
        MechanicalSubSystem;
        PiloteSubSystem;
        DigitalSubSystemStates;
        PiloteSubSystemStates;
        MechanicalSubSystemStates;
        PILOT
CONSTANTS
        mss,
        dss,
        pss,
        pilot,
        orderedForExtensionMechanicalSS,
        orderedForRetractionPiloteSS,
        orderedForExtensionDigitalSS,
        orderedForExtensionPiloteSS,
        orderedForRetractionDigitalSS,
        orderedForRetractionMechanicalSS,
        commandsExtension,
        commandsRetraction
PROPERTIES
        mss : MechanicalSubSystem &
        dss : DigitalSubSystem &
        pss : PiloteSubSystem &
        pilot : PILOT &
        PILOT={pilot} &
        PiloteSubSystem ={pss} &
        MechanicalSubSystem ={mss} &
        DigitalSubSystem ={dss} &
        orderedForRetractionPiloteSS : PiloteSubSystemStates &
        orderedForExtensionPiloteSS : PiloteSubSystemStates &
        orderedForExtensionMechanicalSS : MechanicalSubSystemStates &
        orderedForRetractionDigitalSS : DigitalSubSystemStates &
        orderedForExtensionDigitalSS : DigitalSubSystemStates &
        orderedForRetractionMechanicalSS : MechanicalSubSystemStates &
        orderedForExtensionPiloteSS /= orderedForRetractionPiloteSS &
        orderedForExtensionDigitalSS /= orderedForRetractionDigitalSS &
        orderedForRetractionMechanicalSS /= orderedForExtensionMechanicalSS &
        DigitalSubSystemStates ={orderedForRetractionDigitalSS,
        orderedForExtensionDigitalSS} &
        PiloteSubSystemStates ={orderedForRetractionPiloteSS,
        orderedForExtensionPiloteSS} &
        MechanicalSubSystemStates ={orderedForExtensionMechanicalSS,
        orderedForRetractionMechanicalSS} &
        commandsExtension : {pilot} >-> {pss} &
        commandsRetraction : {pilot} >-> {pss}
END
```

```
REFINEMENT
        LandingGearSystemL1
REFINES
        LandingGearSystemL0
SEES
        LandingGearSystemL1_CONT,
        LandingGearSystemL0_CONT
VARIABLES
        dssState,
        mssState,
        pssState,
        lgState
INVARIANT
        dssState : DigitalSubSystem --> DigitalSubSystemStates &
        mssState : MechanicalSubSystem --> MechanicalSubSystemStates &
        pssState : PiloteSubSystem --> PiloteSubSystemStates
INITIALISATION
        dssState :: {dss} -->DigitalSubSystemStates ||
        mssState :: {mss} -->MechanicalSubSystemStates ||
        pssState :: {pss} -->PiloteSubSystemStates ||
        lgState :: {lg} -->LandingGearSystemStates
EVENTS
        commandsMechanicalSSForRetraction =
        SELECT
                dssState(dss)=orderedForRetractionDigitalSS &
                mssState(mss)=orderedForExtensionMechanicalSS
        THEN
                mssState(mss):=orderedForRetractionMechanicalSS
        END;
        ExtendsLG ref ExtendsLGS=
        SELECT
                lgState(lg)=Retracted &
                mssState(mss)=orderedForExtensionMechanicalSS
        THEN
                lgState(lg):=Extended
        END;
        commandsMechanicalSSForExtension =
        SELECT
                dssState(dss)=orderedForExtensionDigitalSS &
                mssState(mss)=orderedForRetractionMechanicalSS
        THEN
                mssState(mss):=orderedForExtensionMechanicalSS
        END;
        RetractsLG ref RetractsLGS=
        SELECT
                lgState(lg)=Extended &
                mssState(mss)=orderedForRetractionMechanicalSS
        THEN
                lgState(lg):=Retracted
        END;
        commandsDigitalSSForRetraction =
        SELECT
                pssState(pss)=orderedForRetractionPiloteSS &
                dssState(dss)=orderedForExtensionDigitalSS
        THEN
                dssState(dss):=orderedForRetractionDigitalSS
        END;
        commandsPiloteSSForExtension =
        SELECT
                lgState(lg)=Retracted &
                pssState(pss)=orderedForRetractionPiloteSS
        THEN
                pssState(pss):=orderedForExtensionPiloteSS
        END;
```

```
    commandsDigitalSSForExtension =
    SELECT
        pssState(pss)=orderedForExtensionPiloteSS &
        dssState(dss)=orderedForRetractionDigitalSS
    THEN
        dssState(dss):=orderedForExtensionDigitalSS
    END;
    commandsPiloteSSForRetraction =
    SELECT
        lgState(lg)=Extended &
        pssState(pss)=orderedForExtensionPiloteSS
    THEN
        pssState(pss):=orderedForRetractionPiloteSS
    END
END
```

- LandingGearSystemL1 decomposition:



❖ Pilote SubSystem:

- *Event_B specification of PiloteSubSystem_Interface:*

```
SYSTEM
    PiloteSubSystem_Interface
SEES
    LandingGearSystemL1_CONT,
    LandingGearSystemL0_CONT
VARIABLES
    pssState
INVARIANT
    pssState : PiloteSubSystem --> PiloteSubSystemStates
INITIALISATION
    pssState :: {pss} -->PiloteSubSystemStates
EVENTS
    commandsDigitalSSForRetraction =
    SELECT
        pssState(pss)=orderedForRetractionPiloteSS
    THEN
        skip
    END;
```

```
        commandsPiloteSSForExtension =
        SELECT
                pssState(pss)=orderedForRetractionPiloteSS
        THEN
                pssState(pss):=orderedForExtensionPiloteSS
        END;
        commandsDigitalSSForExtension =
        SELECT
                pssState(pss)=orderedForExtensionPiloteSS
        THEN
                skip
        END;
        commandsPiloteSSForRetraction =
        SELECT
                pssState(pss)=orderedForExtensionPiloteSS
        THEN
                pssState(pss):=orderedForRetractionPiloteSS
        END
END
```

- Digital SubSystem:

- *Event_B specification of DigitalSubSystem_Interface:*

```
SYSTEM
        DigitalSubSystem_Interface
SEES
        LandingGearSystemL1_CONT,
        LandingGearSystemL0_CONT
VARIABLES
        dssState
INVARIANT
        dssState : DigitalSubSystem --> DigitalSubSystemStates
INITIALISATION
        dssState :: {dss} -->DigitalSubSystemStates
EVENTS
        commandsDigitalSSForRetraction =
        SELECT
               dssState(dss)=orderedForExtensionDigitalSS
        THEN
               dssState(dss):=orderedForRetractionDigitalSS
        END ;
        commandsDigitalSSForExtension =
        SELECT
               dssState(dss)=orderedForRetractionDigitalSS
        THEN
```

```
            dssState(dss):=orderedForExtensionDigitalSS
      END ;

       commandsMechanicalSSForRetraction =
       SELECT
            dssState(dss)=orderedForRetractionDigitalSS
       THEN
            skip
       END ;
       commandsMechanicalSSForExtension =
       SELECT
            dssState(dss)=orderedForExtensionDigitalSS
       THEN
            skip
       END
END
```

- Mechanical SubSystem:

«Block»
Mechanical SubSystem

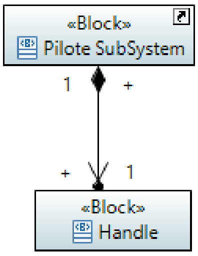
  - *Event_B specification of MechanicalSubSystem_Interface:*

```
SYSTEM
      MechanicalSubSystem_Interface
SEES
      LandingGearSystemL1_CONT,
      LandingGearSystemL0_CONT
VARIABLES
      mssState
INVARIANT
      mssState : MechanicalSubSystem --> MechanicalSubSystemStates
INITIALISATION
      mssState :: {mss} -->MechanicalSubSystemStates
EVENTS
      commandsMechanicalSSForRetraction =
      SELECT
            mssState(mss)=orderedForExtensionMechanicalSS
      THEN
            mssState(mss):=orderedForRetractionMechanicalSS
      END;
      ExtendsLG =
      SELECT
            mssState(mss)=orderedForExtensionMechanicalSS
      THEN
            skip
      END;
      commandsMechanicalSSForExtension =
        SELECT
                mssState(mss)=orderedForRetractionMechanicalSS
        THEN
                mssState(mss):=orderedForExtensionMechanicalSS
        END;
        RetractsLG =
        SELECT
                mssState(mss)=orderedForRetractionMechanicalSS
        THEN
                skip
        END
END
```

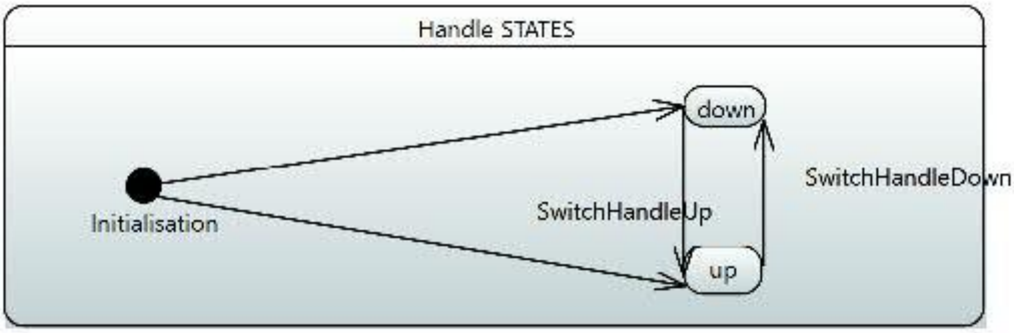
- *LandingGearSystemL1_Refinement_Interface:*

```
SYSTEM
        LandingGearSystemL1_Refinement_Interface
SEES
        LandingGearSystemL1_CONT,
        LandingGearSystemL0_CONT
VARIABLES
        lgState
INVARIANT
        lgState : LandingGearSystem --> LandingGearSystemStates
INITIALISATION
        lgState :: {lg} -->LandingGearSystemStates
EVENTS
        ExtendsLG =
        SELECT
                lgState(lg)=Retracted
        THEN
                lgState(lg):=Extended
        END;
        RetractsLG =
        SELECT
                lgState(lg)=Extended
        THEN
                lgState(lg):=Retracted
        END;
        commandsPiloteSSForExtension =
        SELECT
                lgState(lg)=Retracted
        THEN
                skip END;
        commandsPiloteSSForRetraction =
        SELECT
                lgState(lg)=Extended
        THEN
                skip END
END
```

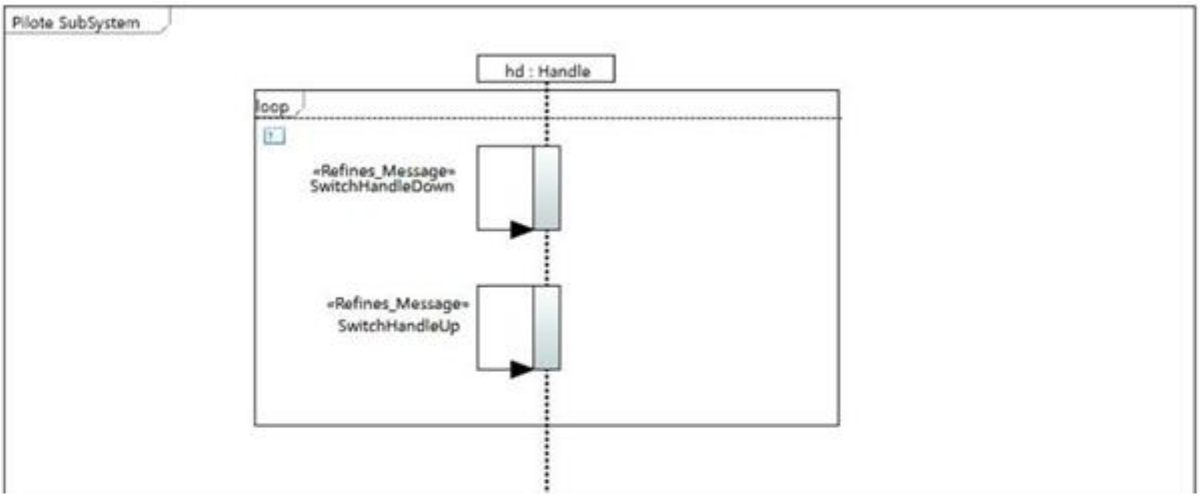
- PiloteSubSystemL1:



Block Definition Diagram of PiloteSubSystemL1



State-machine Diagram of Handle



Sequence Diagram of PiloteSubSystemL1

- Event_B specification of PiloteSubSystemL1:

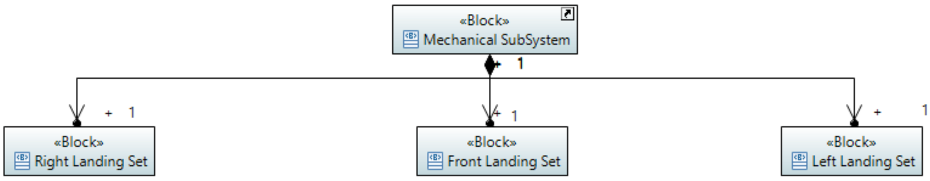
```
SYSTEM
    PiloteSubSystem_CONT
SETS
    Handle;
    HandleSTATES
CONSTANTS
    hd,
    down,
    up
PROPERTIES
    hd : Handle &
    Handle ={hd} &
    down : HandleSTATES &
    up : HandleSTATES &
    up /= down &
    HandleSTATES ={down, up}
END
```

```
REFINEMENT
    PiloteSubSystemL1
REFINES
    PiloteSubSystem_Interface
SEES
    PiloteSubSystem_CONT,
    LandingGearSystemL1_CONT,
    LandingGearSystemL0_CONT
VARIABLES
    hdState,
    pssState
INVARIANT
    hdState : Handle --> HandleSTATES
INITIALISATION
    hdState :: {hd} -->HandleSTATES ||
    pssState :: {pss} -->PiloteSubSystemStates
EVENTS
    SwitchHandleUp ref commandsPiloteSSForRetraction=
    SELECT
        hdState(hd)=down &
        pssState(pss)=orderedForExtensionPiloteSS
    THEN
        hdState(hd):=up ||
        pssState(pss):=orderedForRetractionPiloteSS
    END;
    SwitchHandleDown ref commandsPiloteSSForExtension=
    SELECT
        hdState(hd)=up &
        pssState(pss)=orderedForRetractionPiloteSS
    THEN
        hdState(hd):=down ||
        pssState(pss):=orderedForExtensionPiloteSS
```

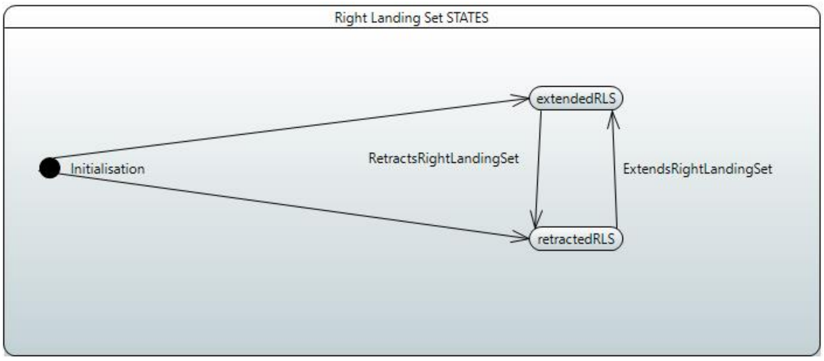
END

END

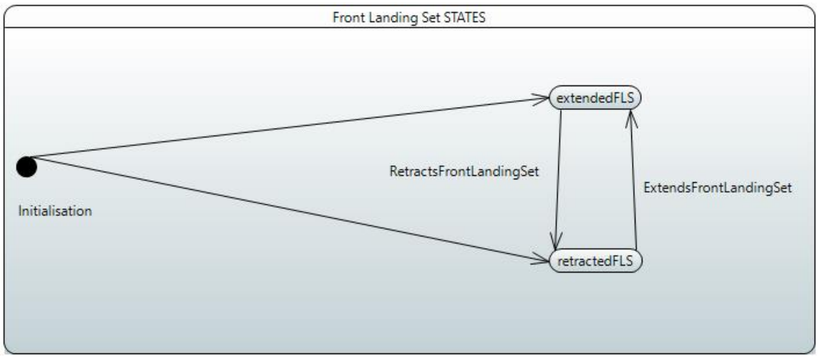
- MechanicalSubSystemL1:



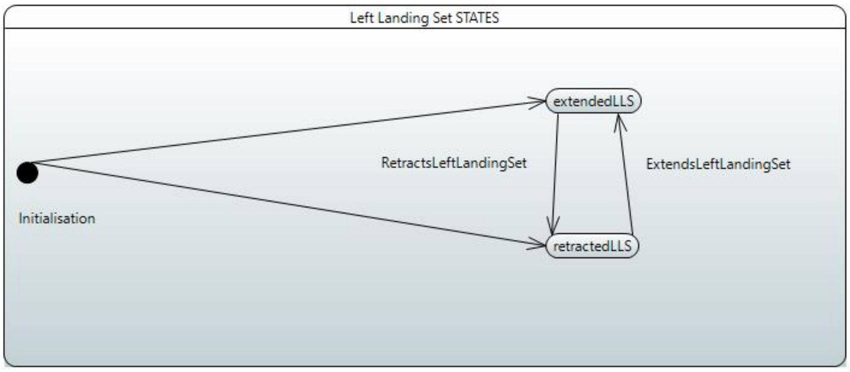
Block Definition Diagram of MechanicalSubSystemL1

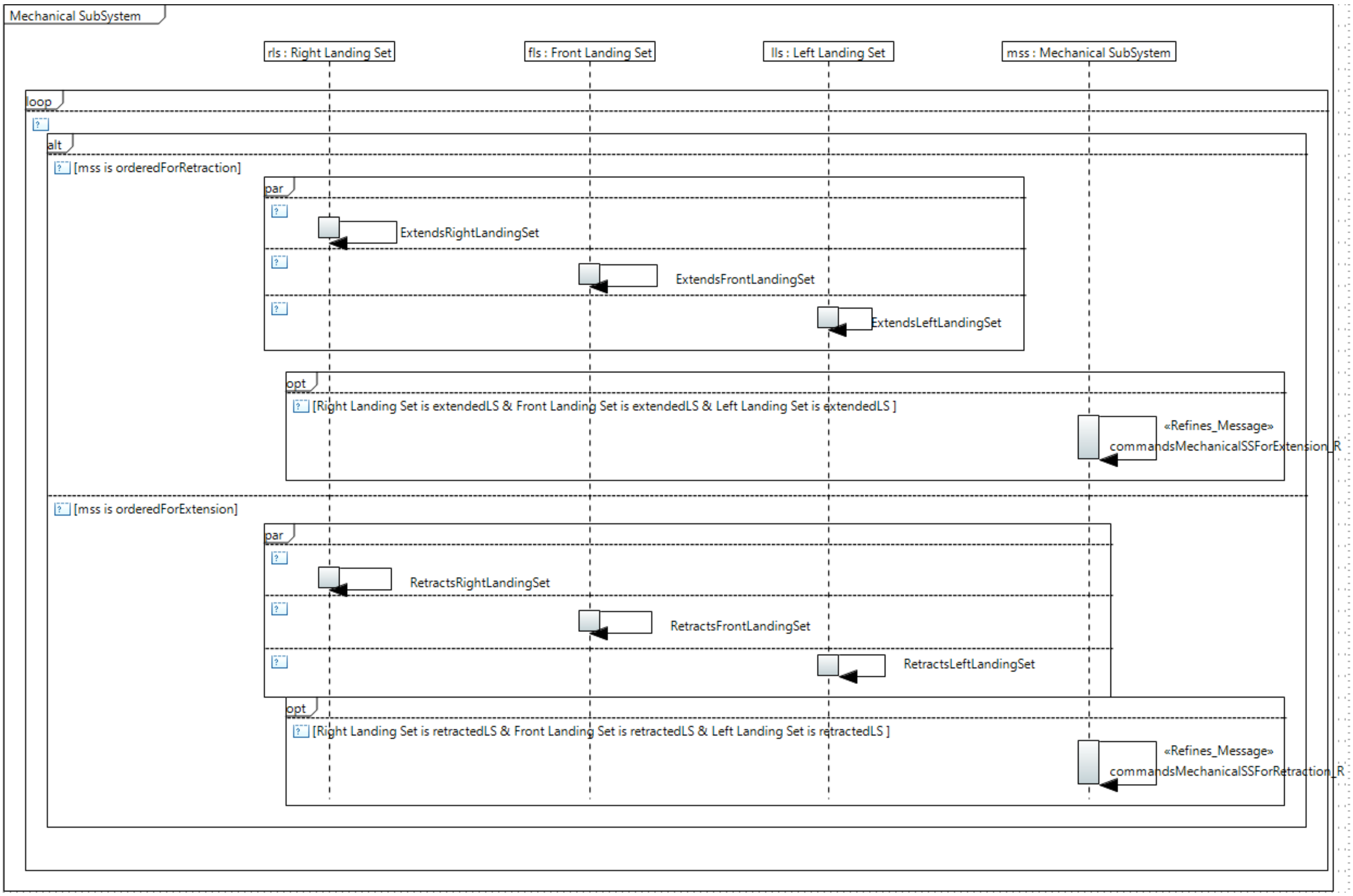


State-machine Diagram of Right Landing Set



State-machine Diagram of Front Landing Set




Mechanical SubSystem
rls : Right Landing Set
fls : Front Landing Set
lls : Left Landing Set
mss : Mechanical SubSystem
loop
alt
[mss is orderedForRetraction]
par
ExtendsRightLandingSet
ExtendsFrontLandingSet
ExtendsLeftLandingSet
opt
[Right Landing Set is extendedLS & Front Landing Set is extendedLS & Left Landing Set is extendedLS]
«Refines_Message»
commandsMechanicalSSForExtension_R
[mss is orderedForExtension]
par
RetractsRightLandingSet
RetractsFrontLandingSet
RetractsLeftLandingSet
opt
[Right Landing Set is retractedLS & Front Landing Set is retractedLS & Left Landing Set is retractedLS]
«Refines_Message»
commandsMechanicalSSForRetraction_R

Sequence Diagram of Mechanical SubSystem L1

- Event_B specification of MechanicalSubSystemL1:

SYSTEM
MechanicalSubSystem_CONT
SETS
LeftLandingSet;
FrontLandingSet;
RightLandingSet;
LeftLandingSetSTATES;
FrontLandingSetSTATES;
RightLandingSetSTATES
CONSTANTS
fls,
lls,
rls,
extendedRLS,
retractedRLS,
extendedFLS,
retractedFLS,
retractedLLS,
extendedLLS

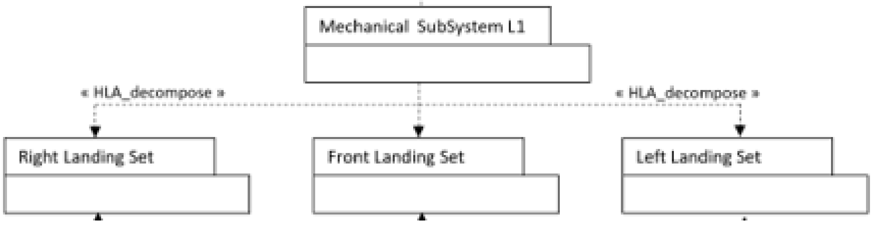
```
PROPERTIES
        rls : RightLandingSet &
        fls : FrontLandingSet &
        lls : LeftLandingSet &
        FrontLandingSet ={fls} &
        LeftLandingSet ={lls} &
        RightLandingSet ={rls} &
        extendedLLS : LeftLandingSetSTATES &
        retractedLLS : LeftLandingSetSTATES &
        extendedRLS : RightLandingSetSTATES &
        retractedFLS : FrontLandingSetSTATES &
        retractedRLS : RightLandingSetSTATES &
        extendedFLS : FrontLandingSetSTATES &
        extendedRLS /= retractedRLS &
        extendedFLS /= retractedFLS &
        extendedLLS /= retractedLLS &
        LeftLandingSetSTATES ={extendedLLS, retractedLLS} &
        FrontLandingSetSTATES ={retractedFLS, extendedFLS} &
        RightLandingSetSTATES ={retractedRLS, extendedRLS}
END
```

```
REFINEMENT
        MechanicalSubSystemL1
REFINES
        MechanicalSubSystem_Interface
SEES
        MechanicalSubSystem_CONT,
        LandingGearSystemL1_CONT,
        LandingGearSystemL0_CONT
VARIABLES
        flsState,
        llsState,
        rlsState,
        mssState
INVARIANT
        flsState : FrontLandingSet --> FrontLandingSetSTATES &
        llsState : LeftLandingSet --> LeftLandingSetSTATES &
        rlsState : RightLandingSet --> RightLandingSetSTATES
INITIALISATION
        flsState :: {fls} -->FrontLandingSetSTATES ||
        llsState :: {lls} -->LeftLandingSetSTATES ||
        rlsState :: {rls} -->RightLandingSetSTATES ||
        mssState :: {mss} -->MechanicalSubSystemStates
EVENTS
        ExtendsRightLandingSet =
        SELECT
                rlsState(rls)=retractedRLS &
                mssState(mss)= orderedForRetractionMechanicalSS
        THEN
                rlsState(rls):=extendedRLS
        END;
```

```
RetractsFrontLandingSet =
SELECT
            flsState(fls)=extendedFLS &
            mssState(mss)= orderedForExtensionMechanicalSS
THEN
            flsState(fls):=retractedFLS
END;
ExtendsLeftLandingSet =
SELECT
            llsState(lls)=retractedLLS &
            mssState(mss)= orderedForRetractionMechanicalSS
THEN
            llsState(lls):=extendedLLS
END;
CommandsMechanicalSSForExtension_R ref CommandsMechanicalSSForExtension=
SELECT
            rlsState(rls)=extendedRLS &
            flsState(fls)=extendedFLS &
            llsState(lls)=extendedLLS &
            mssState(mss)= orderedForRetractionMechanicalSS
THEN
            mssState(mss):= orderedForExtensionMechanicalSS
             END;
commandsMechanicalSSForRetraction_R ref commandsMechanicalSSForRetraction=
SELECT
            rlsState(rls)=retractedRLS &
            flsState(fls)=retractedFLS &
            llsState(lls)=retractedLLS &
            mssState(mss)= orderedForExtensionMechanicalSS
THEN
            mssState(mss):=orderedForRetractionMechanicalSS END;
ExtendsFrontLandingSet =
SELECT
            flsState(fls)=retractedFLS &
            mssState(mss)= orderedForRetractionMechanicalSS
THEN
            flsState(fls):=extendedFLS
END;
RetractsRightLandingSet =
SELECT
            rlsState(rls)=extendedRLS &
            mssState(mss)= orderedForExtensionMechanicalSS
THEN
            rlsState(rls):=retractedRLS
END;
RetractsLeftLandingSet =
SELECT
            llsState(lls)=extendedLLS &
            mssState(mss)= orderedForExtensionMechanicalSS
THEN
```

```
                llsState(lls):=retractedLLS
        END
END
```

- MechanicalSubSystemL1 decomposition:

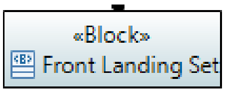


- Right Landing Set:

  - *Event_B specification of RightLandingSet_Interface:*

```
SYSTEM
        RightLandingSet_Interface
SEES
        MechanicalSubSystem_CONT,
        LandingGearSystemL1_CONT,
        LandingGearSystemL0_CONT
VARIABLES
        rlsState
INVARIANT
        rlsState : RightLandingSet --> RightLandingSetSTATES
INITIALISATION
        rlsState :: {rls} -->RightLandingSetSTATES
EVENTS
        ExtendsRightLandingSet =
        SELECT
                rlsState(rls)=retractedRLS
        THEN
                rlsState(rls):=extendedRLS
        END;
        ExtendsLGSS =
        SELECT
                rlsState(rls)=extendedRLS
        THEN
                skip END;
        RetractsLGSS =
        SELECT
                rlsState(rls)=retractedRLS
        THEN
                skip END;
        RetractsRightLandingSet =
    SELECT
        rlsState(rls)=extendedRLS
    THEN
        rlsState(rls):=retractedRLS
    END
END
```

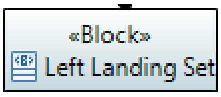
- Front Landing Set:



  - *Event_B specification of FrontLandingSet_Interface:*

```
SYSTEM
    FrontLandingSet_Interface
SEES
    MechanicalSubSystem_CONT,
    LandingGearSystemL1_CONT,
    LandingGearSystemL0_CONT
VARIABLES
    flsState
INVARIANT
    flsState : FrontLandingSet --> FrontLandingSetSTATES
INITIALISATION
    flsState :: {fls} -->FrontLandingSetSTATES
EVENTS
    RetractsFrontLandingSet =
    SELECT
        flsState(fls)=extendedFLS
    THEN
        flsState(fls):=retractedFLS
    END;
    ExtendsLG =
    SELECT
        flsState(fls)=extendedFLS
    THEN
        skip END;
    RetractsLG =
    SELECT
        flsState(fls)=retractedFLS
    THEN
        skip END;
    ExtendsFrontLandingSet =
    SELECT
        flsState(fls)=retractedFLS
    THEN
        flsState(fls):=extendedFLS
    END
END
```

- Left Landing Set:



  - *Event_B specification of LeftLandingSet_Interface:*

```
SYSTEM
        LeftLandingSet_Interface
SEES
        MechanicalSubSystem_CONT,
        LandingGearSystemL1_CONT,
        LandingGearSystemL0_CONT
VARIABLES
        llsState
INVARIANT
        llsState : LeftLandingSet --> LeftLandingSetSTATES
INITIALISATION
        llsState :: {lls} -->LeftLandingSetSTATES
EVENTS
        ExtendsLeftLandingSet =
        SELECT
                llsState(lls)=retractedLLS
        THEN
                llsState(lls):=extendedLLS
        END;
        ExtendsLG =
        SELECT
                llsState(lls)=extendedLLS
        THEN
                skip END;
        RetractsLG =
        SELECT
                llsState(lls)=retractedLLS
        THEN
                skip END;
        RetractsLeftLandingSet =
        SELECT
                llsState(lls)=extendedLLS
        THEN
                llsState(lls):=retractedLLS
        END
END
```

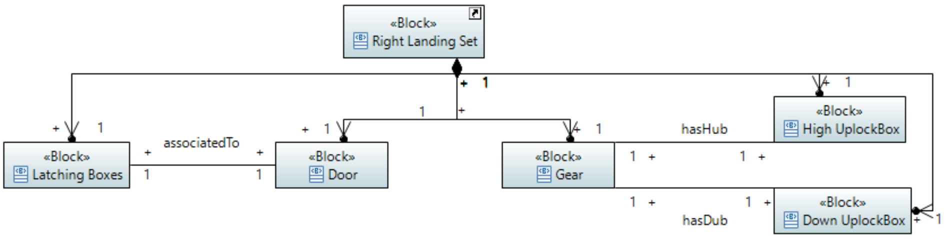
  - *MechanicalSubSystem_Refinement_Interface:*

```
SYSTEM
        MechanicalSubSystem_Refinement_Interface
SEES
        MechanicalSubSystem_CONT,
        LandingGearSystemL1_CONT,
        LandingGearSystemL0_CONT
VARIABLES
        mssState
```

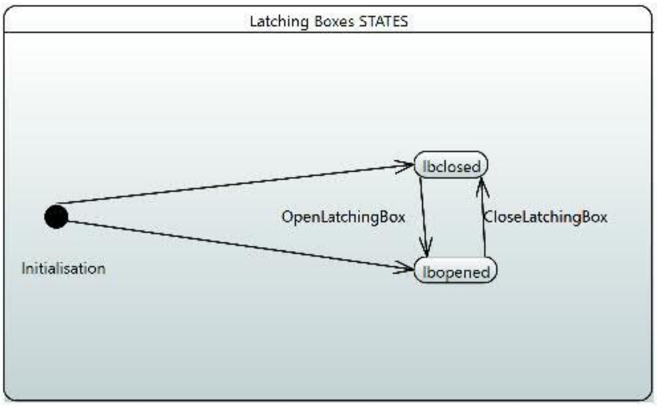
```
INVARIANT
        mssState : MechanicalSubSystem --> MechanicalSubSystemStates
INITIALISATION
        mssState :: {mss} -->MechanicalSubSystemStates
EVENTS
        ExtendsRightLandingSet =
        SELECT
                mssState(mss)=orderedForExtensionMechanicalSS
        THEN
                skip END;
        RetractsFrontLandingSet =
        SELECT
                mssState(mss)=orderedForRetractionMechanicalSS
        THEN
                skip END;
        ExtendsLeftLandingSet =
        SELECT
                mssState(mss)=orderedForExtensionMechanicalSS
        THEN
                skip END;
        ExtendsLG =
        SELECT
                mssState(mss)=orderedForExtensionMechanicalSS
        THEN
                skip END;
        RetractsLG =
        SELECT
                mssState(mss)=orderedForRetractionMechanicalSS
        THEN
                skip END;
        ExtendsFrontLandingSet =
        SELECT
                mssState(mss)=orderedForExtensionMechanicalSS
        THEN
                skip END;
        RetractsRightLandingSet =
        SELECT
                mssState(mss)=orderedForRetractionMechanicalSS
        THEN
                skip END;

        RetractsLeftLandingSet =
        SELECT
                mssState(mss)=orderedForRetractionMechanicalSS
        THEN
                skip END
END
```

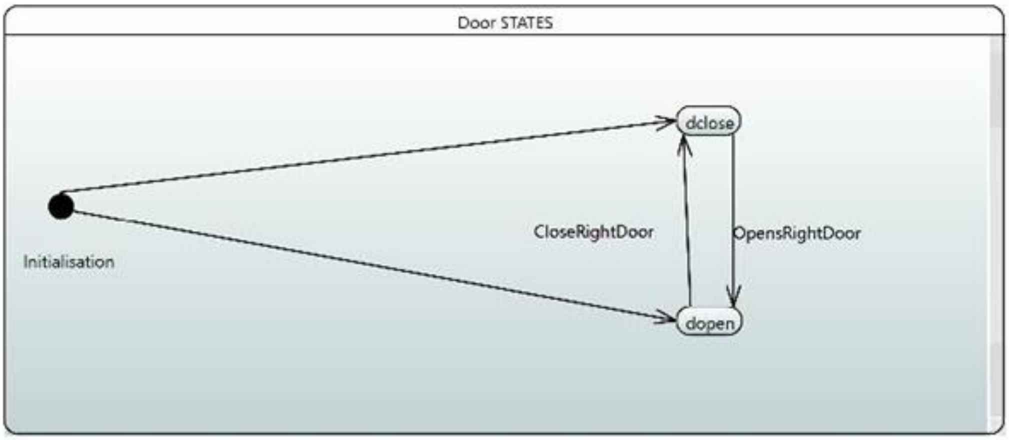
- Right Landing Set L1:



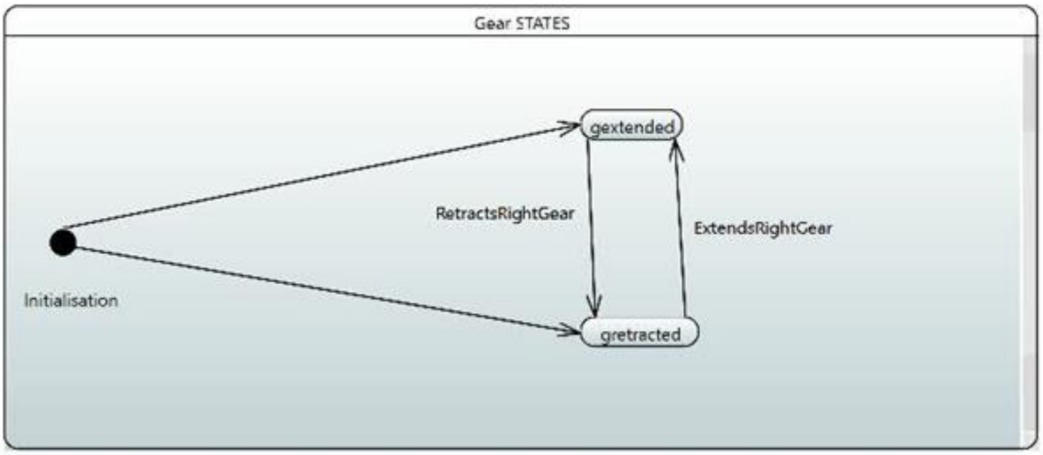
Block Definition Diagram of RightLandingSetL1

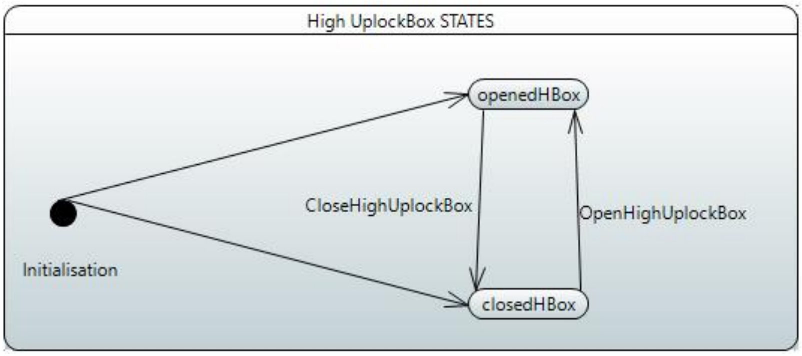


State-machine Diagram of Latching Boxes

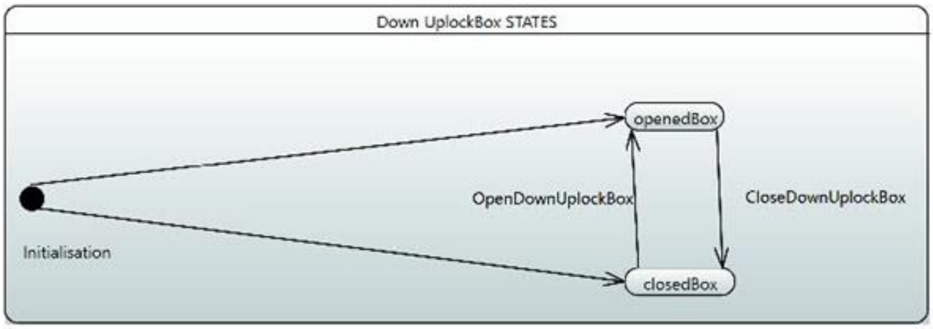


State-machine Diagram of Right Door

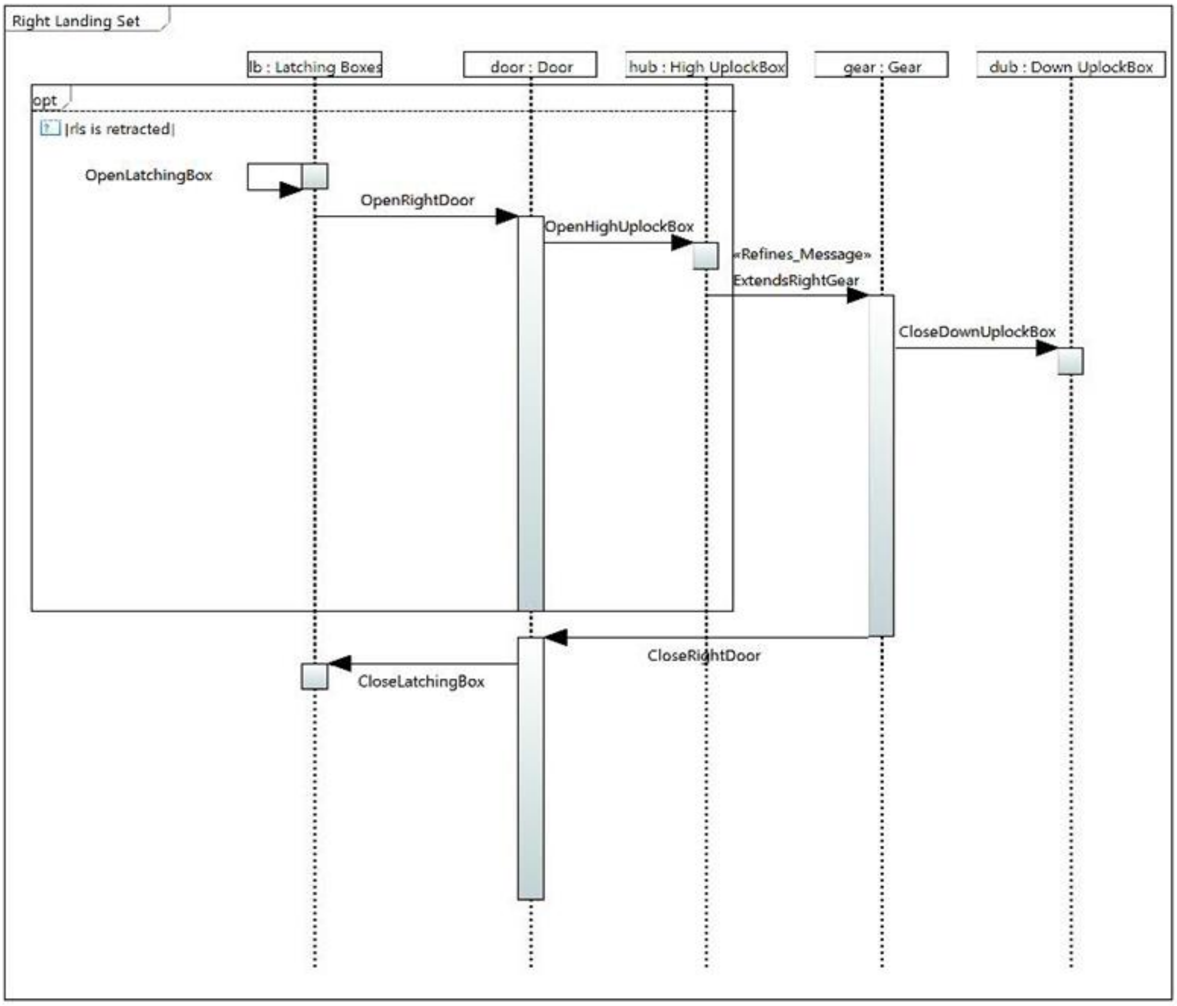




State-machine Diagram of High UplockBox



State-machine Diagram of Down UplockBox



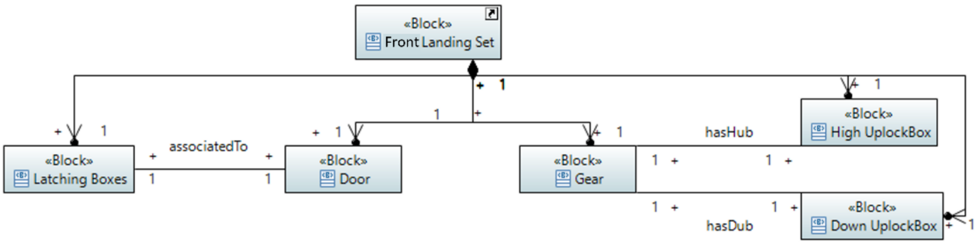
Sequence Diagram of RightLandingSetL1

- Event_B specification of RightLandingSetL1:

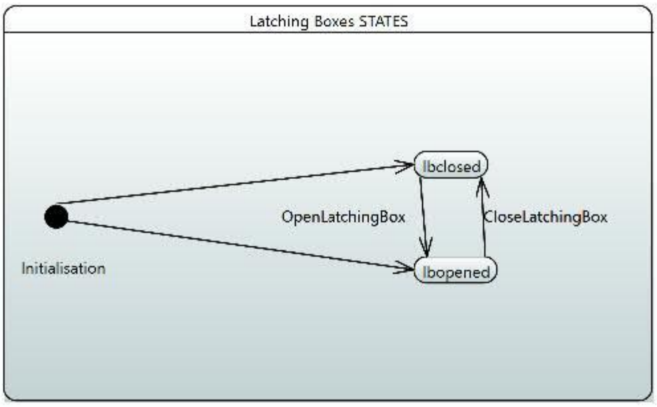
```
SYSTEM
        RightLandingSet_CONT
SETS
        HighUplockBox;
        DownUplockBox;
        Gear;
        LatchingBoxes;
        Door;
        DoorSTATES;
        HighUplockBoxSTATES;
        DownUplockBoxSTATES;
        GearSTATES;
        LatchingBoxesSTATES
CONSTANTS
        gear,
        lb,
        hub,
        dub,
        door,
        openedHBox,
        openedDBox,
        closedHBox,
        dclose,
        gextended,
        lbclosed,
        dopen,
        gretracted,
        lbopened,
        closedDBox,
        hasHub,
        hasDub,
        associatedTo
PROPERTIES
        dub : DownUplockBox &
        lb : LatchingBoxes &
        hub : HighUplockBox &
        door : Door &
        gear : Gear &
        HighUplockBox ={hub} &
        DownUplockBox ={dub} &
        Gear ={gear} &
        Door ={door} &
        LatchingBoxes ={lb} &
        closedHBox : HighUplockBoxSTATES &
        dopen : DoorSTATES &
        openedDBox : DownUplockBoxSTATES &
        openedHBox : HighUplockBoxSTATES &
        lbopened : LatchingBoxesSTATES &
        closedDBox : DownUplockBoxSTATES &
        gretracted : GearSTATES &
        lbclosed : LatchingBoxesSTATES &
        gextended : GearSTATES &
        dclose : DoorSTATES &
        gretracted /= gextended &
        dclose /= dopen &
        lbopened /= lbclosed &
        openedHBox /= closedHBox &
        openedDBox /= closedDBox &
        GearSTATES ={gextended, gretracted} &
        DownUplockBoxSTATES ={closedDBox, openedDBox} &
        LatchingBoxesSTATES ={lbclosed, lbopened} &
        HighUplockBoxSTATES ={closedHBox, openedHBox} &
        DoorSTATES ={dopen, dclose} &
        hasHub : {gear} >-> {hub} &
        hasDub : {gear} >->{dub} &
        associatedTo : {lb} >-> {door}
END
```

```
REFINEMENT
        RightLandingSetL1
REFINES
        RightLandingSet_Interface
SEES
        RightLandingSet_CONT,
        MechanicalSubSystem_CONT,
        LandingGearSystemL1_CONT,
        LandingGearSystemL0_CONT
VARIABLES
        doorState,
        dubState,
        gearState,
        hubState,
        lbState,
        rlsState
INVARIANT
        doorState : Door --> DoorSTATES &
        dubState : DownUplockBox --> DownUplockBoxSTATES &
        gearState : Gear --> GearSTATES &
        hubState : HighUplockBox --> HighUplockBoxSTATES &
        lbState : LatchingBoxes --> LatchingBoxesSTATES
INITIALISATION
        doorState :: {door} -->DoorSTATES ||
        dubState :: {dub} -->DownUplockBoxSTATES ||
        gearState :: {gear} -->GearSTATES ||
        hubState :: {hub} -->HighUplockBoxSTATES ||
        lbState :: {lb} -->LatchingBoxesSTATES ||
        rlsState :: {rls} -->RightLandingSetSTATES
EVENTS
        OpenRightDoor =
        SELECT
            lbState(lb)=lbopened &
            doorState(door)=dclose
        THEN
            doorState(door):=dopen
        END;
        CloseLatchingBox =
        SELECT
            doorState(door)=dclose &
            lbState(lb)=lbopened
        THEN
            lbState(lb):=lbclosed
        END;
        ExtendsRightGear ref ExtendsRightLandingSet=
        SELECT
            hubState(hub)=openedHBox &
            gearState(gear)=gretracted &
            rlsState(rls)=retractedRLS
        THEN
            gearState(gear):=gextended ||
            rlsState(rls):=extendedRLS
        END;
        OpenHighUplockBox =
        SELECT
            doorState(door)=dopen &
            hubState(hub)=closedHBox
        THEN
            hubState(hub):=openedHBox
        END;
        OpenLatchingBox =
        SELECT
            lbState(lb)=lbclosed
        THEN
            lbState(lb):=lbopened
        END;
        CloseRightDoor =
        SELECT
            dubState(dub)=closedDBox &
            doorState(door)=dopen
        THEN
            doorState(door):=dclose
        END;
        CloseDownUplockBox =
        SELECT
            gearState(gear)=gextended &
            dubState(dub)=openedDBox
        THEN
            dubState(dub):=closedDBox
        END
END
```

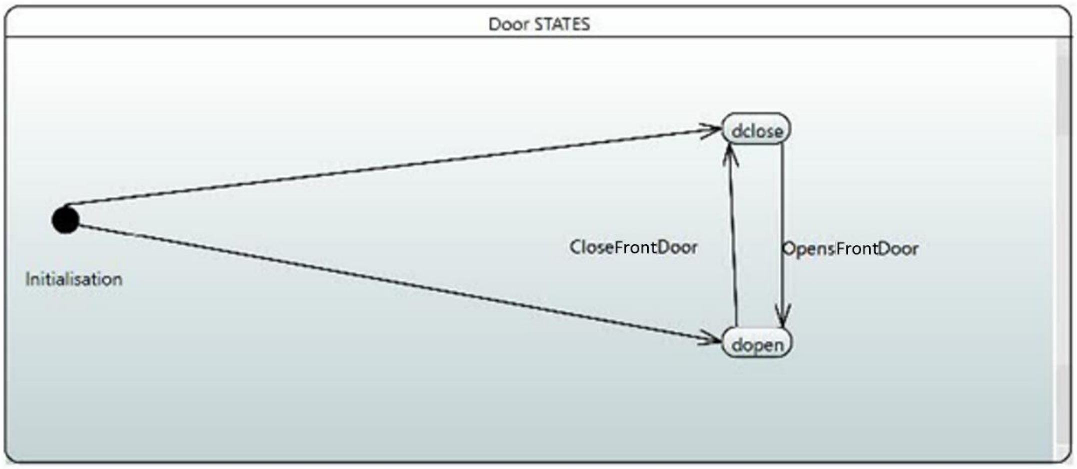
- Front Landing Set L1:



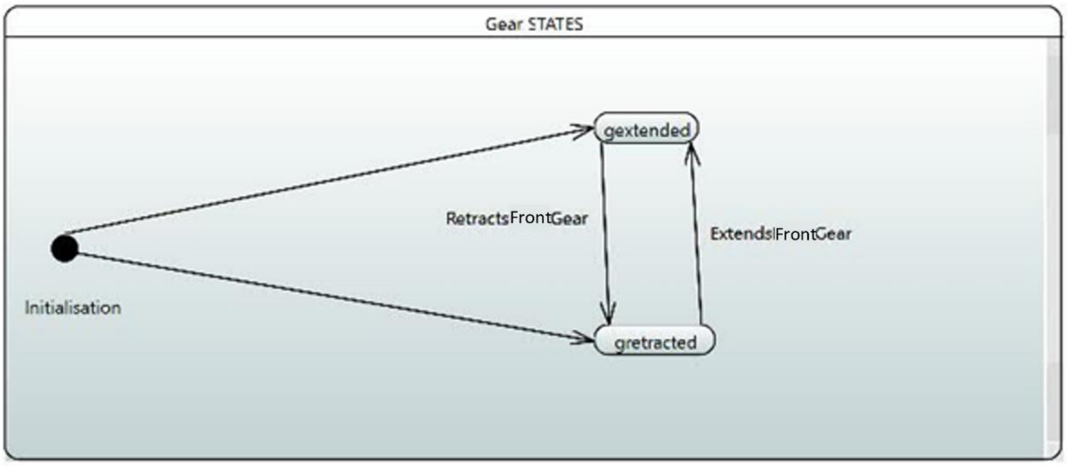
Block Definition Diagram of FrontLandingSetL1



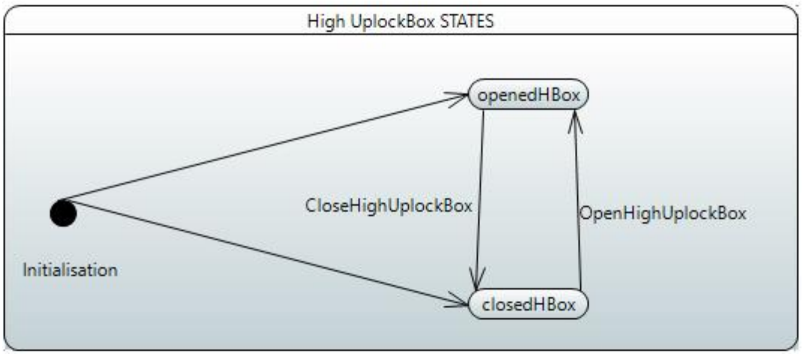
State-machine Diagram of Latching Boxes



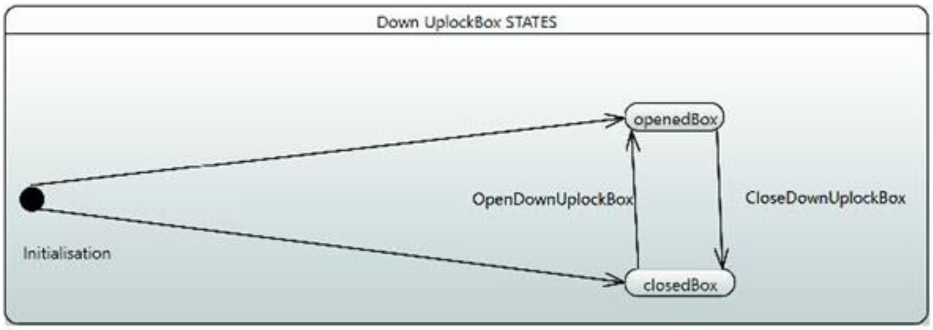
State-machine Diagram of Front Door



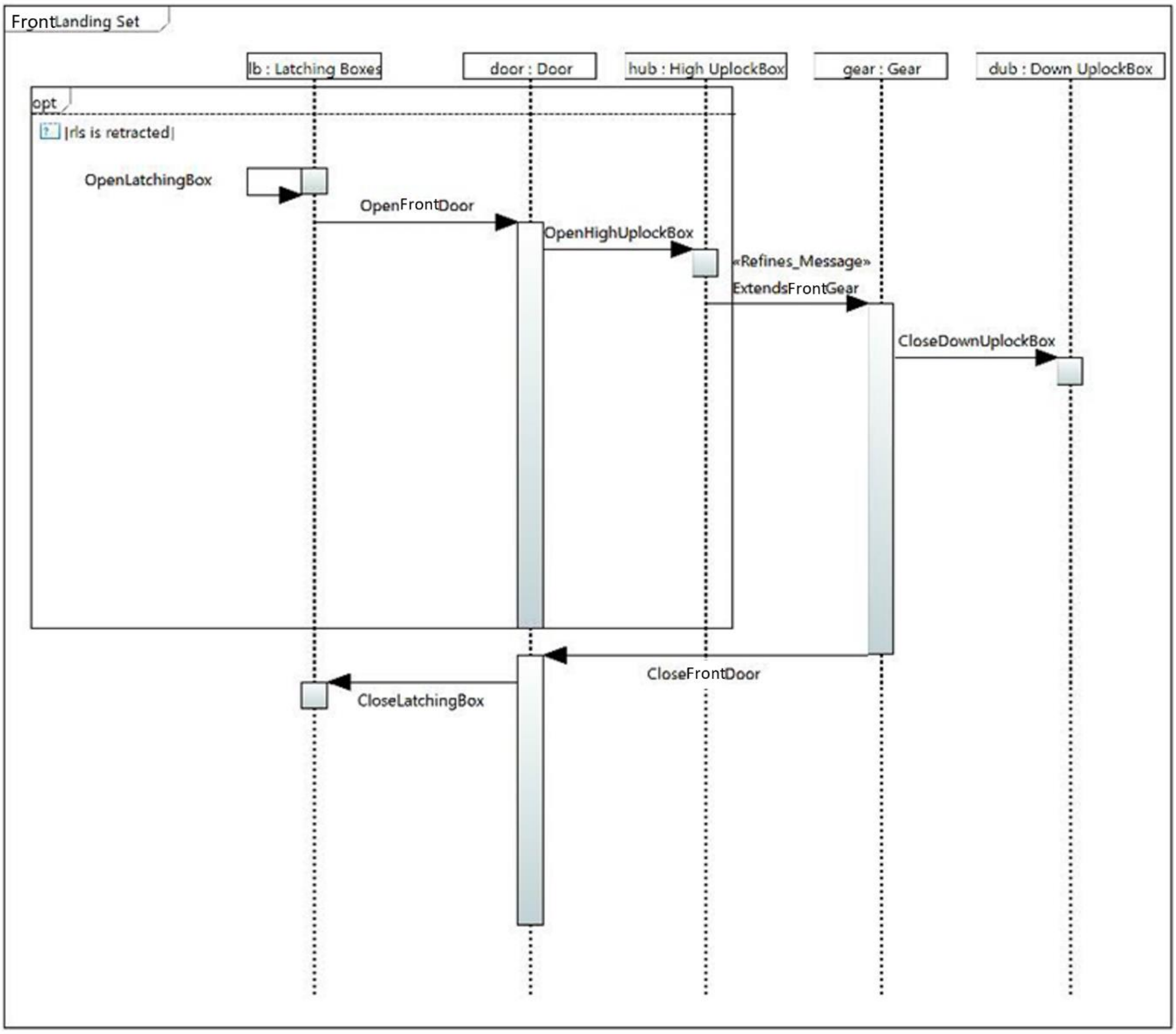
State-machine Diagram of Front Gear



State-machine Diagram of High UplockBox



State-machine Diagram of Down UplockBox



Sequence Diagram of FrontLandingSetL1

- Event_B specification of FrontLandingSetL1:

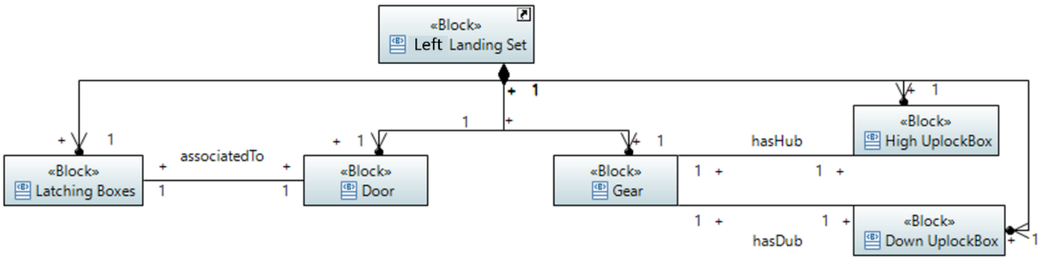
```
SYSTEM
        FrontLandingSet_CONT
SETS
        HighUplockBox;
        LatchingBoxes;
        Gear;
        Door;
        DownUplockBox;
        GearSTATES;
        LatchingBoxesSTATES;
        HighUplockBoxSTATES;
        DownUplockBoxSTATES;
        DoorSTATES
CONSTANTS
        hub,
        door,
        lb,
        dub,
        gear,
        openedHBox,
        openedDBox,
        closedHBox,
        dclose,
        gextended,
        lbopened,
        gretracted,
        lbclosed,
        dopen,
        closedDBox,
        hasHub,
        hasDub,
        associatedTo
PROPERTIES
        dub : DownUplockBox &
        lb : LatchingBoxes &
        hub : HighUplockBox &
        door : Door &
        gear : Gear &
        HighUplockBox ={hub} &
        DownUplockBox ={dub} &
        Gear ={gear} &
        Door ={door} &
        LatchingBoxes ={lb} &
        closedHBox : HighUplockBoxSTATES &
        dopen : DoorSTATES &
        openedDBox : DownUplockBoxSTATES &
        openedHBox : HighUplockBoxSTATES &
        lbopened : LatchingBoxesSTATES &
        closedDBox : DownUplockBoxSTATES &
        gretracted : GearSTATES &
        lbclosed : LatchingBoxesSTATES &
        gextended : GearSTATES &
        dclose : DoorSTATES &
        gretracted /= gextended &
        dclose /= dopen &
        lbopened /= lbclosed &
        openedHBox /= closedHBox &
        openedDBox /= closedDBox &
        GearSTATES ={gextended, gretracted} &
```

```
    DownUplockBoxSTATES ={closedDBox, openedDBox} &
    LatchingBoxesSTATES ={lbclosed, lbopened} &
    HighUplockBoxSTATES ={closedHBox, openedHBox} &
    DoorSTATES ={dopen, dclose} &
    hasHub : {gear} >-> {hub} &
    hasDub : {gear} >->{dub} &
    associatedTo : {lb} >-> {door}
END
```

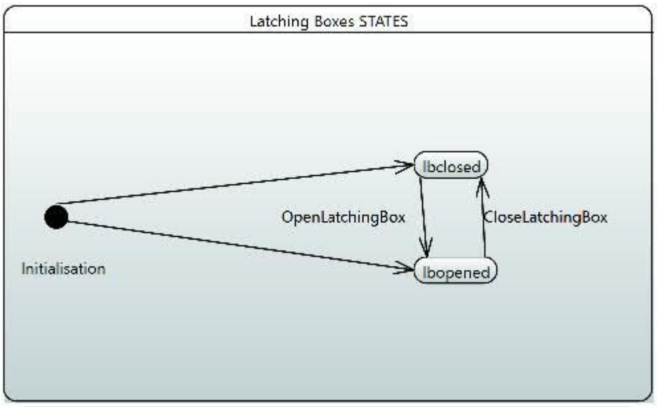
```
REFINEMENT
    FrontLandingSetL1
REFINES
    FrontLandingSet_Interface
SEES
    FrontLandingSet_CONT,
    MechanicalSubSystem_CONT,
    LandingGearSystemL1_CONT,
    LandingGearSystemL0_CONT
VARIABLES
    doorState,
    dubState,
    gearState,
    hubState,
    lbState,
    flsState
INVARIANT
    doorState : Door --> DoorSTATES &
    dubState : DownUplockBox --> DownUplockBoxSTATES &
    gearState : Gear --> GearSTATES &
    hubState : HighUplockBox --> HighUplockBoxSTATES &
    lbState : LatchingBoxes --> LatchingBoxesSTATES
INITIALISATION
    doorState :: {door} -->DoorSTATES ||
    dubState :: {dub} -->DownUplockBoxSTATES ||
    gearState :: {gear} -->GearSTATES ||
    hubState :: {hub} -->HighUplockBoxSTATES ||
    lbState :: {lb} -->LatchingBoxesSTATES ||
    flsState :: {fls} -->FrontLandingSetSTATES
EVENTS
    CloseDownUplockBox =
    SELECT
        gearState(gear)=gextended &
        dubState(dub)=openedDBox
    THEN
        dubState(dub):=closedDBox
    END;
    CloseFrontDoor =
    SELECT
        dubState(dub)=closedDBox &
        doorState(door)=dopen
    THEN
```

```
                doorState(door):=dclose
        END;
        OpenFrontDoor =
        SELECT
                lbState(lb)=lbopened &
                doorState(door)=dclose
        THEN
                doorState(door):=dopen
        END;
        CloseLatchingBox =
        SELECT
                doorState(door)=dclose &
                lbState(lb)=lbopened
        THEN
                lbState(lb):=lbclosed
        END;
        OpenLatchingBox =
        SELECT
                lbState(lb)=lbclosed
        THEN
                lbState(lb):=lbopened
        END;
        OpenHighUplockBox =
        SELECT
                doorState(door)=dopen &
                hubState(hub)=closedHBox
        THEN
                hubState(hub):=openedHBox
        END;
        ExtendsFrontGear ref ExtendsFrontLandingSet=
        SELECT
                hubState(hub)=openedHBox &
                gearState(gear)=gretracted &
                flsState(fls)=retractedFLS
        THEN
                gearState(gear):=gextended ||
                flsState(fls):=extendedFLS
        END
END
```

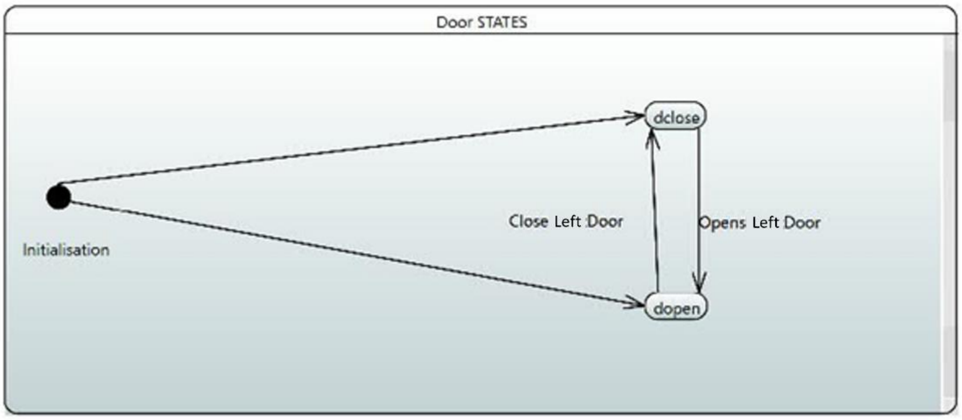
- Left Landing Set L1:



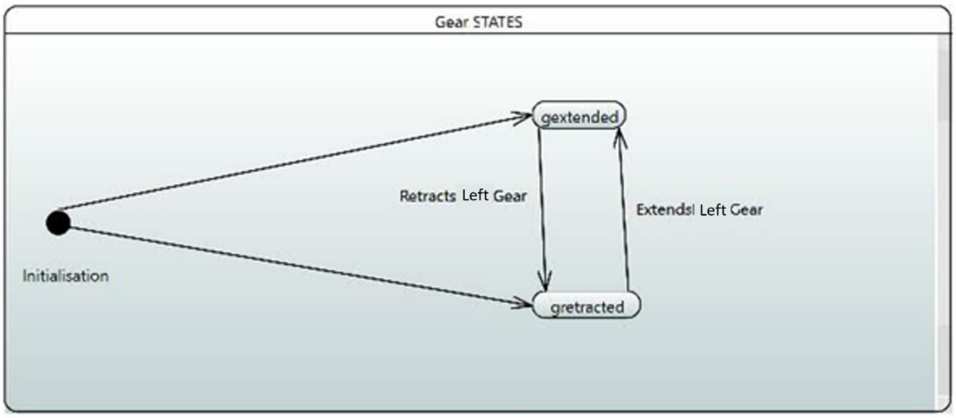
Block Definition Diagram of LeftLandingSetL1



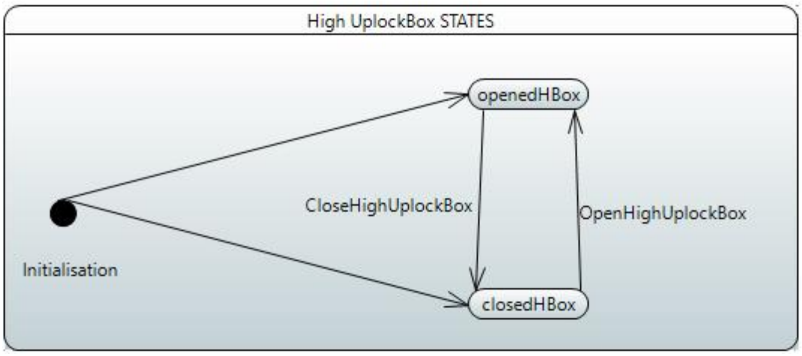
State-machine Diagram of Latching Boxes



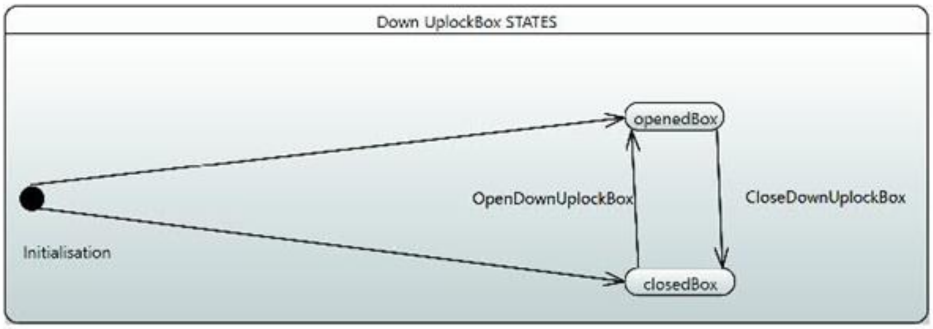
State-machine Diagram of Left Door



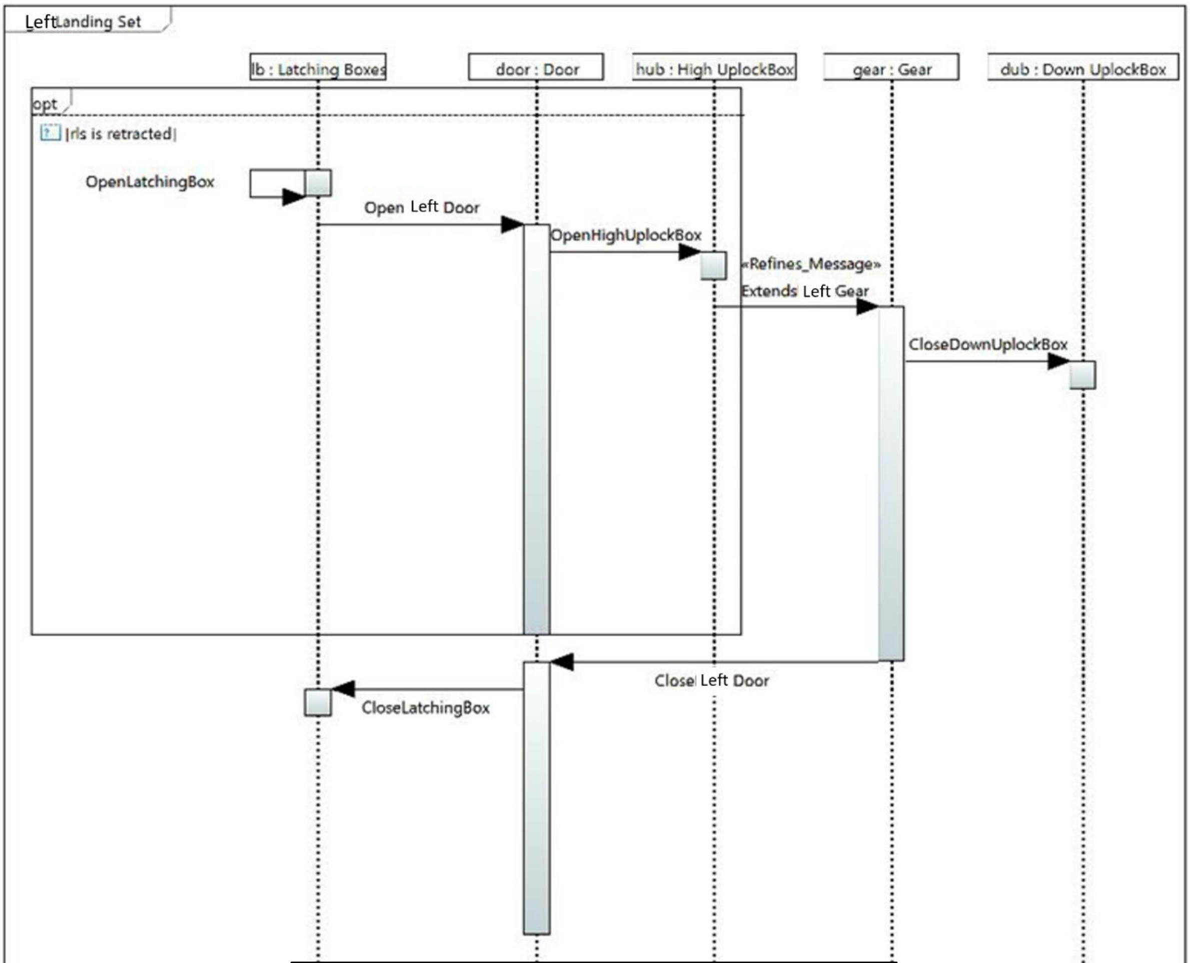
State-machine Diagram of Left Gear



State-machine Diagram of High UplockBox



State-machine Diagram of Down UplockBox



Sequence Diagram of LeftLandingSetL1

- Event_B specification of LeftLandingSetL1:

```
SYSTEM
        LeftLandingSet_CONT
SETS
        Gear;
        LatchingBoxes;
        Door;
        HighUplockBox;
        DownUplockBox;
        HighUplockBoxSTATES;
        DoorSTATES;
        DownUplockBoxSTATES;
        LatchingBoxesSTATES;
        GearSTATES
CONSTANTS
        hub,
        door,
        lb,
        dub,
        gear,
        openedHBox,
        openedDBox,
        closedHBox,
        dclose,
        gextended,
        lbopened,
        gretracted,
        lbclosed,
        dopen,
        closedDBox,
        hasHub,
        hasDub,
        associatedTo
PROPERTIES
        dub : DownUplockBox &
        lb : LatchingBoxes &
        hub : HighUplockBox &
        door : Door &
        gear : Gear &
        HighUplockBox ={hub} &
        DownUplockBox ={dub} &
        Gear ={gear} &
        Door ={door} &
        LatchingBoxes ={lb} &
        closedHBox : HighUplockBoxSTATES &
        dopen : DoorSTATES &
        openedDBox : DownUplockBoxSTATES &
        openedHBox : HighUplockBoxSTATES &
        lbopened : LatchingBoxesSTATES &
        closedDBox : DownUplockBoxSTATES &
        gretracted : GearSTATES &
        lbclosed : LatchingBoxesSTATES &
        gextended : GearSTATES &
        dclose : DoorSTATES &
        gretracted /= gextended &
        dclose /= dopen &
        lbopened /= lbclosed &
        openedHBox /= closedHBox &
        openedDBox /= closedDBox &
        GearSTATES ={gextended, gretracted} &
        DownUplockBoxSTATES ={closedDBox, openedDBox} &
        LatchingBoxesSTATES ={lbclosed, lbopened} &
        HighUplockBoxSTATES ={closedHBox, openedHBox} &
        DoorSTATES ={dopen, dclose} &
        hasHub : {gear} >-> {hub} &
        hasDub : {gear} >->{dub} &
        associatedTo : {lb} >-> {door}
END
```

```
REFINEMENT
        LeftLandingSetL1
REFINES
        LeftLandingSet_Interface
SEES
        LeftLandingSet_CONT,
        MechanicalSubSystem_CONT,
        LandingGearSystemL1_CONT,
        LandingGearSystemL0_CONT
VARIABLES
        doorState,
        dubState,
        gearState,
        hubState,
        lbState,
        llsState
INVARIANT
        doorState : Door --> DoorSTATES &
        dubState : DownUplockBox --> DownUplockBoxSTATES &
        gearState : Gear --> GearSTATES &
        hubState : HighUplockBox --> HighUplockBoxSTATES &
        lbState : LatchingBoxes --> LatchingBoxesSTATES
INITIALISATION
        doorState :: {door} -->DoorSTATES ||
        dubState :: {dub} -->DownUplockBoxSTATES ||
        gearState :: {gear} -->GearSTATES ||
        hubState :: {hub} -->HighUplockBoxSTATES ||
        lbState :: {lb} -->LatchingBoxesSTATES ||
        llsState :: {lls} -->LeftLandingSetSTATES

        EVENTS
```

```
    OpenLatchingBox =
    SELECT
        lbState(lb)=lbclosed
    THEN
        lbState(lb):=lbopened END;
    CloseLeftDoor =
    SELECT
        dubState(dub)=closedDBox &
        doorState(door)=dopen
    THEN
        doorState(door):=dclose END;
    OpenHighUplockBox =
    SELECT
        doorState(door)=dopen &
        hubState(hub)=closedHBox
    THEN
        hubState(hub):=openedHBox END;
    ExtendsLeftGear ref ExtendsLeftLandingSet=
    SELECT
        hubState(hub)=openedHBox &
        gearState(gear)=gretracted &
        llsState(lls)=retractedLLS
    THEN
        gearState(gear):=gextended ||
        llsState(lls):=extendedLLS
    END;
    OpenLeftDoor =
    SELECT
        lbState(lb)=lbopened &
        doorState(door)=dclose
    THEN
        doorState(door):=dopen
    END;
    CloseLatchingBox =
    SELECT
        doorState(door)=dclose &
        lbState(lb)=lbopened
    THEN
        lbState(lb):=lbclosed
    END;
    CloseDownUplockBox =
    SELECT
        gearState(gear)=gextended &
        dubState(dub)=openedDBox
    THEN
        dubState(dub):=closedDBox
    END
END
```