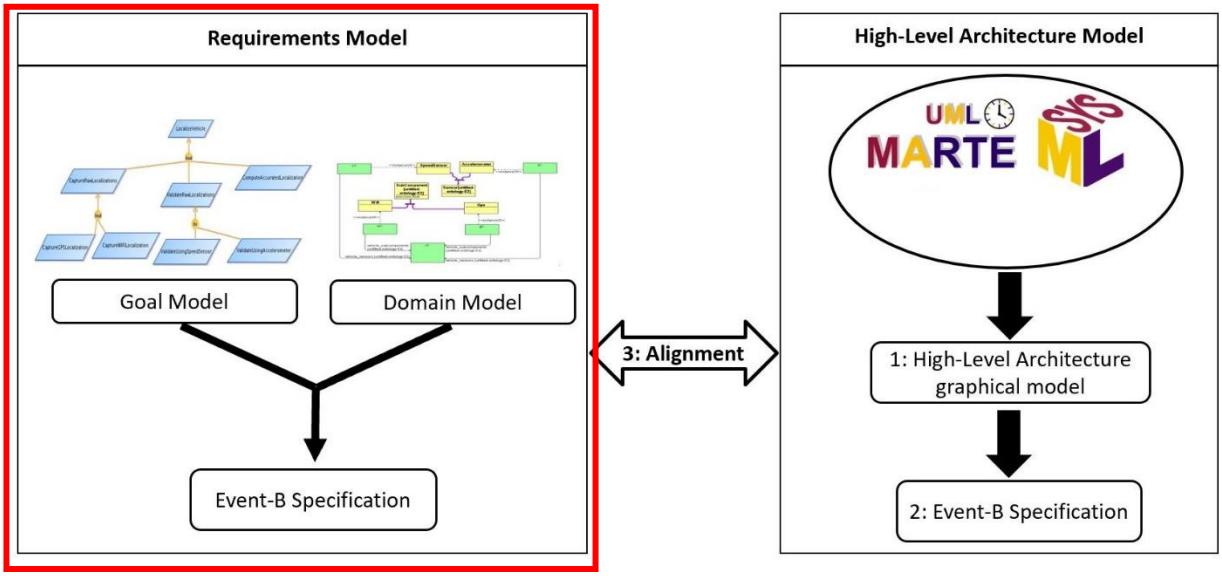
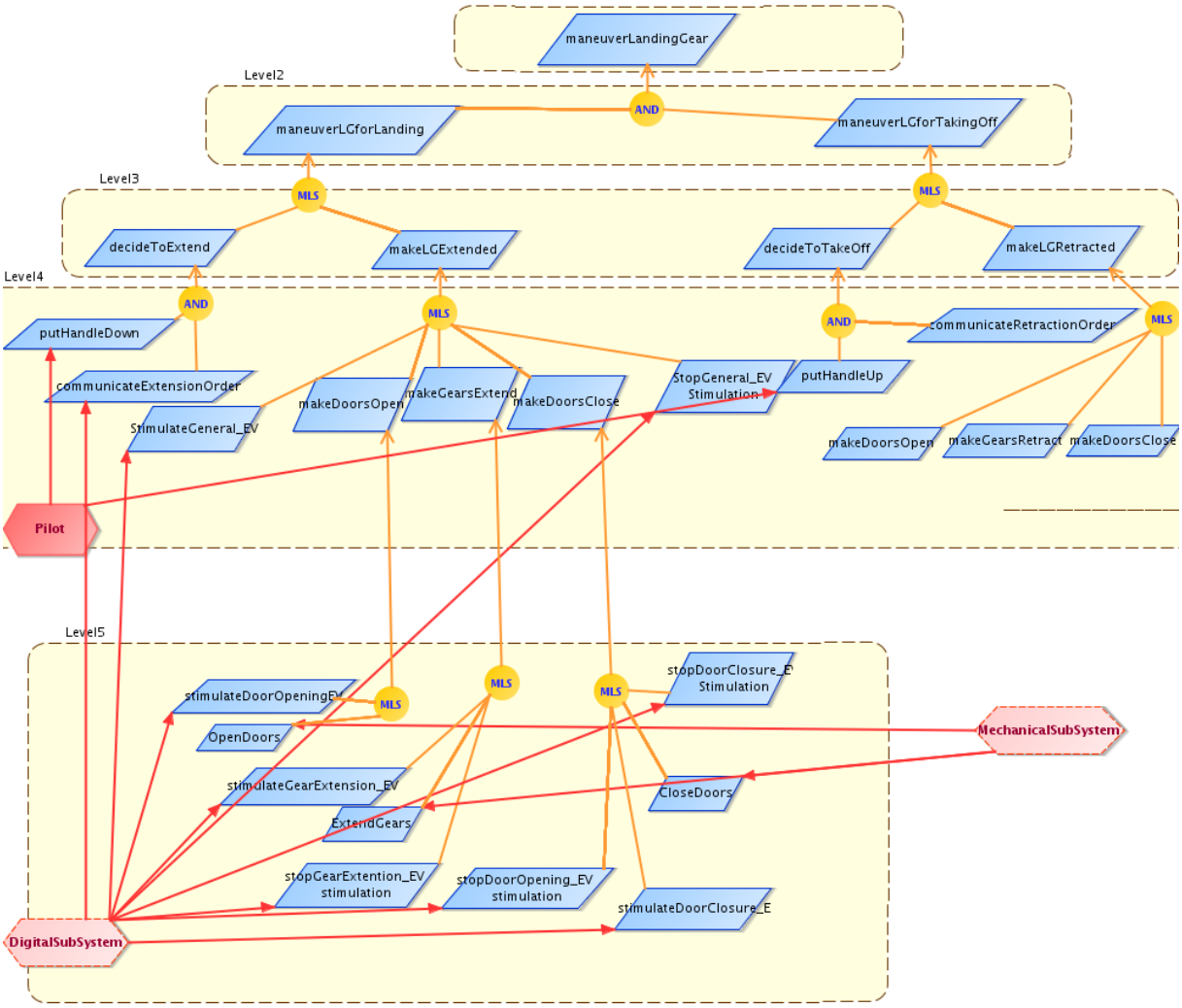
Ce document présente la 1ere partie de la méthodologie de la thèse :



- **Modèle SysML/KAOS abstrait du « Landing Gear System » composé de 5 niveaux :**

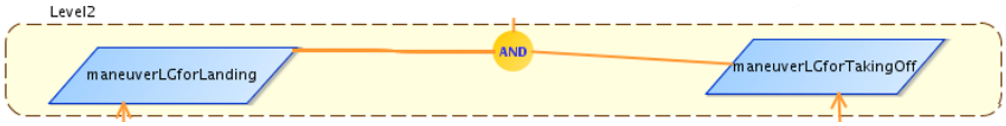


- 1er niveau :

```
SYSTEM
    LandingGearSystem_CONTEXT
SETS
    LandingGearSystem;
    LG_STATES
CONSTANTS
    LG,
    lg_extended,
    lg_retracted
PROPERTIES
    LG : LandingGearSystem
    & LandingGearSystem={LG}
    & lg_extended : LG_STATES
    & lg_retracted : LG_STATES
    & LG_STATES={lg_extended,lg_retracted}
    & lg_extended /= lg_retracted
END
```


```
SYSTEM
    Landing_Gear_System
SEES
    LandingGearSystem_CONTEXT
VARIABLES
    lgState
INVARIANT
    lgState: LandingGearSystem --> LG_STATES
INITIALISATION
    lgState::LandingGearSystem --> LG_STATES
EVENTS
    maneuverLandingGear=
    ANY state  WHERE
    state : LG_STATES
    THEN

    lgState(LG) := state
END

END
```

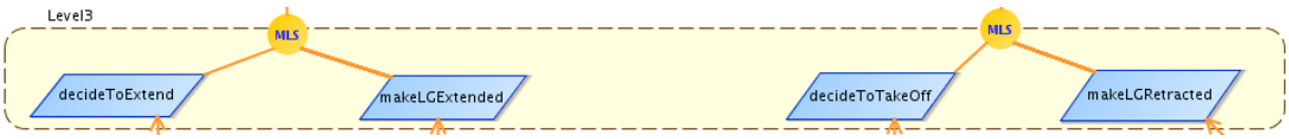
- 2éme niveau :




```
SYSTEM
    LandingGearSystem_CONTEXT_L1
SETS
    MANEUVER_TYPES
CONSTANTS
    taking_off, landing
PROPERTIES
    taking_off : MANEUVER_TYPES
    & landing : MANEUVER_TYPES
    & MANEUVER_TYPES={taking_off,landing}
    & taking_off /= landing
END
```

```
REFINEMENT Landing_Gear_System_L1
REFINES Landing_Gear_System

SEES
     LandingGearSystem_CONTEXT,
    LandingGearSystem_CONTEXT_L1
VARIABLES
    currentManeuver,lgState
INVARIANT
    currentManeuver : LandingGearSystem --> MANEUVER_TYPES
INITIALISATION
    lgState::LandingGearSystem --> LG_STATES
    || currentManeuver::LandingGearSystem --> MANEUVER_TYPES
EVENTS

    maneuverLGforLanding ref_and maneuverLandingGear=
    BEGIN
    lgState(LG):=lg_extended
    || currentManeuver(LG):=landing
    END;

    maneuverLGforTakingOff ref_and  maneuverLandingGear=
   BEGIN
    lgState(LG):=lg_retracted
    || currentManeuver(LG):=taking_off
    END

END
```

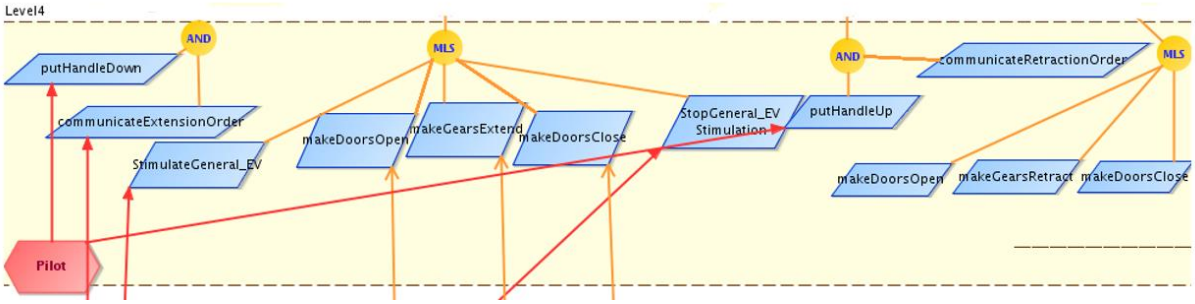
- 3éme niveau :



```
SYSTEM
    LandingGearSystem_CONTEXT_L2
SETS
    Actions
CONSTANTS
    Extension,
    Retraction
PROPERTIES
    Extension : Actions &
    Retraction: Actions &
    Extension /= Retraction &
    Actions={Extension, Retraction}
END
```

```
REFINEMENT Landing_Gear_System_L2
REFINES Landing_Gear_System_L1

SEES
    LandingGearSystem_CONTEXT,
    LandingGearSystem_CONTEXT_L1,
    LandingGearSystem_CONTEXT_L2

ABSTRACT_VARIABLES
    currentManeuver, lgState, decision
INVARIANT
 decision : LandingGearSystem --> Actions
INITIALISATION
 lgState::LandingGearSystem --> LG_STATES
 || currentManeuver::LandingGearSystem --> MANEUVER_TYPES
 || decision :: LandingGearSystem --> Actions

EVENTS
    decideToExtend ref_milestone maneuverLGforLanding=
    BEGIN
        decision(LG):=Extension
    END;

    makeLGExtended ref_milestone maneuverLGforLanding=
    SELECT decision(LG)=Extension THEN
       lgState(LG):=lg_extended
    || currentManeuver(LG):=landing

    END
END
```

- 4éme niveau :



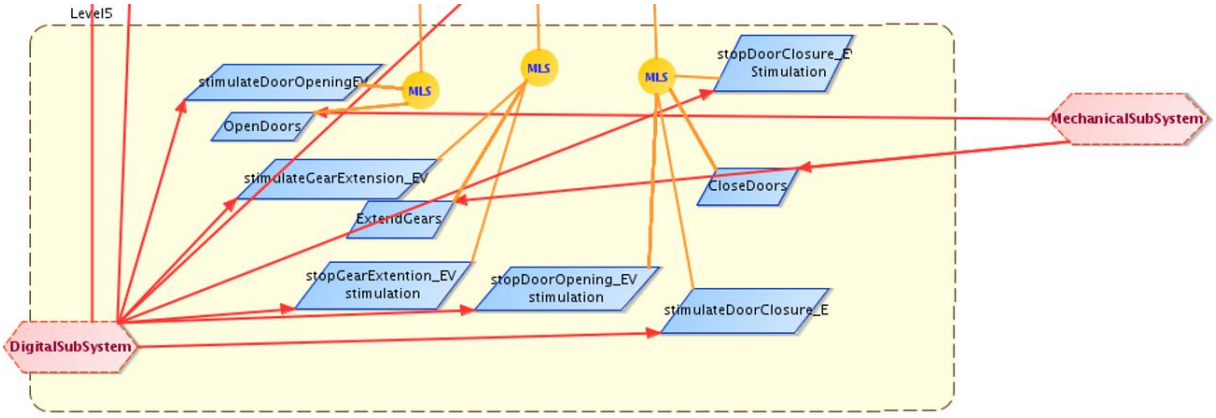

```
SYSTEM
        LandingGearSystem_CONTEXT_L3
SEES
        LandingGearSystem_CONTEXT
SETS
        Handle; HANDLE_STATES; ELECTRO_VALVES; EV_States; Doors; Doors_States; Gears;
        Gears_States
CONSTANTS
        LgOfHd, HD, down, up,
        stimulated, stopped,
        lgOfG_EV, G_EV,
        lgOfDRS, open,close , DRS,
        lgOfGRS, extended, retracted, GRS
PROPERTIES
        HD : Handle
        & Handle = {HD}
        & up : HANDLE_STATES
        & down : HANDLE_STATES
        & HANDLE_STATES={up,down}
        & LgOfHd : Handle >->> LandingGearSystem
        & LgOfHd = {HD|->LG}
        & up /= down
        & DRS:Doors
        & Doors={DRS}
        & open: Doors_States
        & close : Doors_States
        & open /= close
```

```
        & Doors_States={close, open}
        & lgOfDRS: Doors >->>LandingGearSystem
        & lgOfDRS ={DRS|->LG}
        & GRS: Gears
        & Gears={GRS}
        & extended : Gears_States
        & retracted : Gears_States
        & extended /= retracted
        & Gears_States={extended,retracted}
        & lgOfGRS: Gears >->>LandingGearSystem
        & lgOfGRS ={GRS|->LG}
        & stimulated : EV_States
        & stopped : EV_States
        & stimulated /= stopped
        & EV_States={stimulated, stopped}
        & G_EV : ELECTRO_VALVES
        & ELECTRO_VALVES ={G_EV}
        & lgOfG_EV : ELECTRO_VALVES >->>LandingGearSystem
        & lgOfG_EV = {G_EV|->LG}
END
```

```
REFINEMENT Landing_Gear_System_L3
REFINES Landing_Gear_System_L2
SEES
        LandingGearSystem_CONTEXT,
        LandingGearSystem_CONTEXT_L1,
        LandingGearSystem_CONTEXT_L2,
        LandingGearSystem_CONTEXT_L3
ABSTRACT_VARIABLES
        currentManeuver, lgState, decision, handleState, extensionOrder, GEVstate, doorsState,
        gearsState
INVARIANT
        handleState : Handle --> HANDLE_STATES &
        extensionOrder : LandingGearSystem --> BOOL &
        GEVstate : ELECTRO_VALVES --> EV_States &
        doorsState : Doors --> Doors_States &
        gearsState : Gears --> Gears_States
INITIALISATION
        lgState::LandingGearSystem --> LG_STATES
        || currentManeuver::LandingGearSystem --> MANEUVER_TYPES
        || decision :: LandingGearSystem --> Actions
        || handleState :: Handle --> HANDLE_STATES
        || doorsState :: Doors --> Doors_States
        || gearsState :: Gears --> Gears_States
        || extensionOrder :: LandingGearSystem -->BOOL
        || GEVstate :: ELECTRO_VALVES --> EV_States
EVENTS
putHandleDown ref_and decideToExtend=
BEGIN
        handleState(LgOfHd~(LG)):=down
END;

communicateExtensionOrder ref_and decideToExtend=
SELECT handleState(LgOfHd~(LG))=down THEN
        extensionOrder(LG) :=TRUE
        || decision(LG):=Extension
END;
////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////

stimulateGeneral_EV ref_milestone makeLGExtended =
SELECT extensionOrder(LG) = TRUE & decision(LG)=Extension THEN
        GEVstate(lgOfG_EV~(LG)):= stimulated
END;


makeDoorsOpen ref_milestone makeLGExtended =
SELECT GEVstate(lgOfG_EV~(LG))=stimulated THEN
        doorsState(lgOfDRS~(LG)) := open
END;


makeGearsExtend ref_milestone makeLGExtended =
SELECT doorsState(lgOfDRS~(LG)) = open THEN
        gearsState(GRS) := extended
END;
makeDoorsClose ref_milestone makeLGExtended =
```

```
SELECT gearsState(lgOfGRS~(LG)) = extended THEN
        doorsState(DRS) := close
END;
stopGeneral_EVstimulation ref_milestone makeLGExtended =
SELECT doorsState(lgOfDRS~(LG)) = close THEN
        GEVstate(lgOfG_EV~(LG)):= stopped
        || lgState(LG):=lg_extended
        || currentManeuver(LG):=landing
END

END
```

- 5éme niveau :



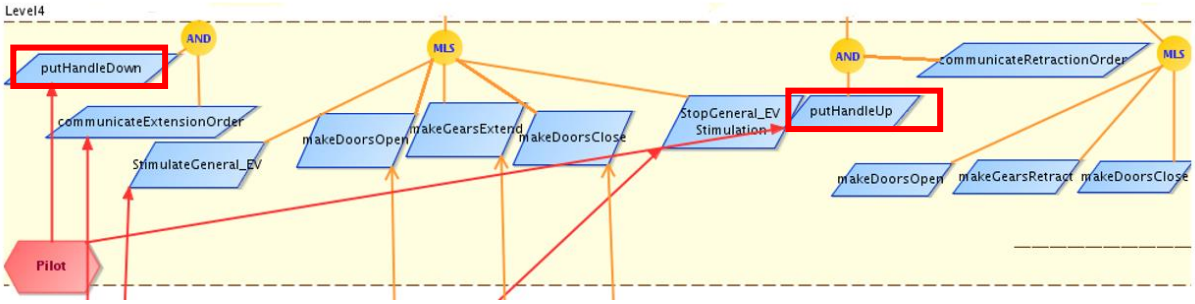

```
SYSTEM
        LandingGearSystem_CONTEXT_L4
SEES
        LandingGearSystem_CONTEXT_L3
CONSTANTS
        DO_EV, DRSofDO_EV,
        GE_EV, GRSofGE_EV,
        DC_EV, DRSofDC_EV
PROPERTIES
        DO_EV : ELECTRO_VALVES &
        DRSofDO_EV: ELECTRO_VALVES >->> Doors &
        DRSofDO_EV= {DO_EV|->DRS} &
        GE_EV : ELECTRO_VALVES &
        GRSofGE_EV: ELECTRO_VALVES >->> Gears &
        GRSofGE_EV={GE_EV|->GRS} &
        DC_EV : ELECTRO_VALVES &
        DRSofDC_EV: ELECTRO_VALVES >->> Doors &
        DRSofDC_EV={DC_EV|->DRS} &
        G_EV /= DO_EV &
        DO_EV /= GE_EV &
        G_EV /= GE_EV &
        GE_EV /= DC_EV &
        DO_EV /= DC_EV &
        DC_EV /= G_EV &
        ELECTRO_VALVES={G_EV, DO_EV, GE_EV, DC_EV}

END
```

```
REFINEMENT Landing_Gear_System_L4
REFINES Landing_Gear_System_L3
SEES
        LandingGearSystem_CONTEXT,
        LandingGearSystem_CONTEXT_L1,
        LandingGearSystem_CONTEXT_L2,
        LandingGearSystem_CONTEXT_L3,
        LandingGearSystem_CONTEXT_L4

ABSTRACT_VARIABLES
        currentManeuver, lgState, decision, handleState, extensionOrder, GEVstate, doorsState,
        gearsState, DO_EVstate,
        GE_EVstate,
        DC_EVstate
```

```
INVARIANT
        DO_EVstate :ELECTRO_VALVES --> EV_States &
        GE_EVstate :ELECTRO_VALVES --> EV_States &
        DC_EVstate :ELECTRO_VALVES --> EV_States
INITIALISATION
        lgState::LandingGearSystem --> LG_STATES
        || currentManeuver::LandingGearSystem --> MANEUVER_TYPES
        || decision :: LandingGearSystem --> Actions
        || handleState :: Handle --> HANDLE_STATES
        || doorsState :: Doors --> Doors_States
        || gearsState :: Gears --> Gears_States
        || extensionOrder :: LandingGearSystem -->BOOL
        || GEVstate :: ELECTRO_VALVES --> EV_States
        || DO_EVstate :: ELECTRO_VALVES --> EV_States
        || GE_EVstate :: ELECTRO_VALVES --> EV_States
        || DC_EVstate :: ELECTRO_VALVES --> EV_States
EVENTS

        stimulateDoorOpeningEV ref_milestone makeDoorsOpen =
        SELECT GEVstate(lgOfG_EV~(LG))=stimulated THEN
                DO_EVstate(DRSofDO_EV~(DRS)) := stimulated
        END;

        OpenDoors ref_milestone makeDoorsOpen =
        SELECT DO_EVstate(DRSofDO_EV~(DRS)) = stimulated THEN
                doorsState(lgOfDRS~(LG)) := open
        END;
////////////////////////////////////////////////////


        stimulateGearExtension_EV ref_milestone makeGearsExtend =
        SELECT doorsState(lgOfDRS~(LG)) = open THEN
                GE_EVstate(GRSofGE_EV~(GRS)):=stimulated
        END;

        ExtendGears ref_milestone makeGearsExtend =
        SELECT GE_EVstate(GRSofGE_EV~(GRS))=stimulated THEN
                gearsState(lgOfGRS~(LG)) := extended
        END;

        stopGearExtension_EV ref_milestone makeGearsExtend=
        SELECT gearsState(lgOfGRS~(LG))= extended THEN
                GE_EVstate(GRSofGE_EV~(GRS)):=stopped
        END;

////////////////////////////////////////////////////

        stopDoorOpeningEV ref_milestone makeDoorsClose =
        SELECT gearsState(lgOfGRS~(LG)) = extended THEN
                DO_EVstate(DRSofDO_EV~(DRS)) := stopped
        END;

        stimulateDoorClosure_EV ref_milestone makeDoorsClose =
        SELECT DO_EVstate(DRSofDO_EV~(DRS)) = stopped THEN
                DC_EVstate(DRSofDC_EV~(DRS)):= stimulated
        END;


        closeDoors ref_milestone makeDoorsClose =
        SELECT DC_EVstate(DRSofDC_EV~(DRS)) = stimulated THEN
                doorsState(lgOfDRS~(LG)) := close
        END;

        stopDoorClosure_EV ref_milestone makeDoorsClose =
        SELECT doorsState(lgOfDRS~(LG)) = close THEN
                DC_EVstate(DRSofDC_EV~(DRS)):= stopped
        END
END
```

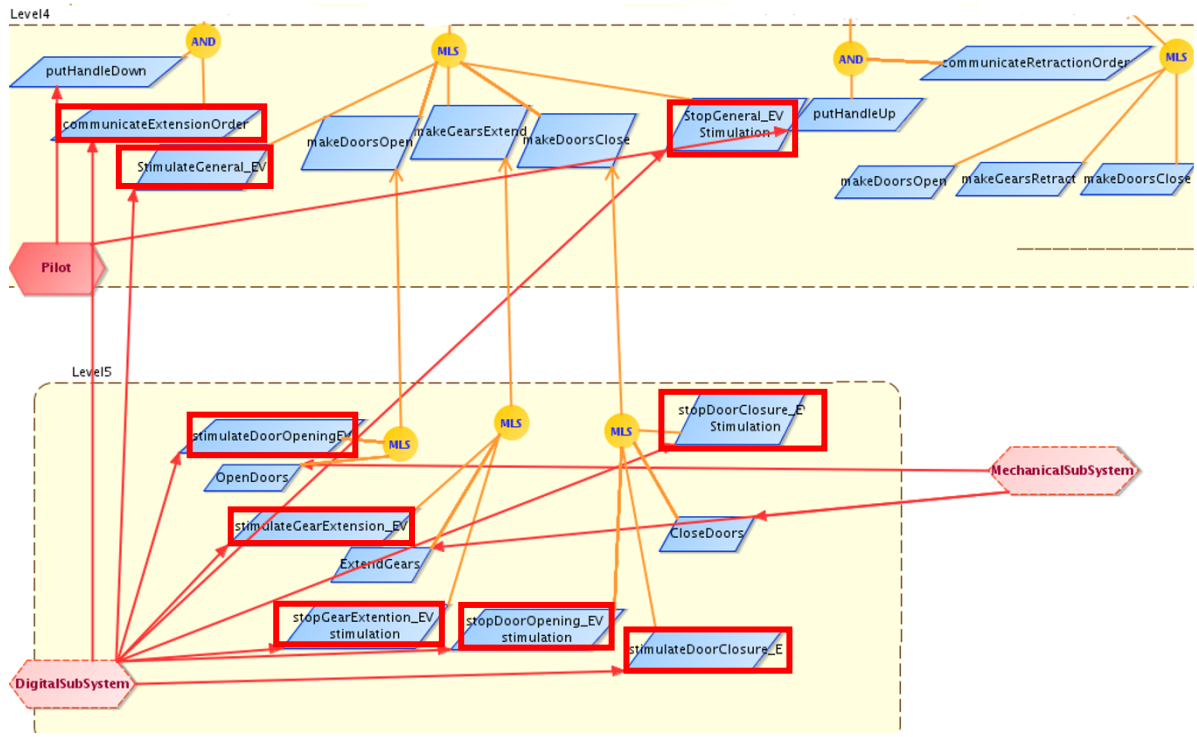
Décomposition System/Sous-Systèmes en utilisant l'approche de Steve (représentation des sous-systèmes par les agents du modèle SysML/KAOS) :

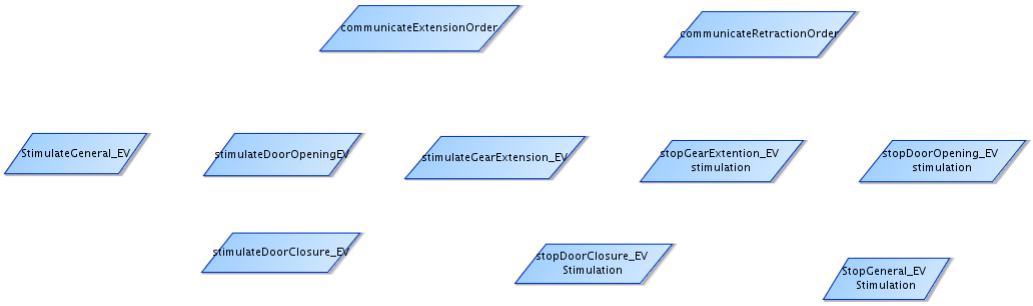
- Sous-Système Pilote :




```
SYSTEM
        PiloteSS_Interface
SEES
        LandingGearSystem_CONTEXT,
        LandingGearSystem_CONTEXT_L3
ABSTRACT_VARIABLES
        /****Shared variable used in DigitalSS_Interface*******/
        handleState //updated in PiloteSS_Interface and read in DigitalSS_Interface
INVARIANT
        handleState : Handle --> HANDLE_STATES
INITIALISATION
        handleState :: Handle --> HANDLE_STATES

EVENTS
        putHandleDown =
        BEGIN
                handleState(LgOfHd~(LG)):=down
        END;

        putHandleUp =
        BEGIN
                handleState(LgOfHd~(LG)):=up
        END
        /******************Shared Events of DigitalSS_Interface*********************
        communicateExtensionOrder =
        SELECT handleState(LgOfHd~(LG))=down THEN
        extensionOrder(LG) :=TRUE
        || decision(LG):=Extension
        END;
        ********************************************************************************/
END
```

- Sous-Système Digitale :

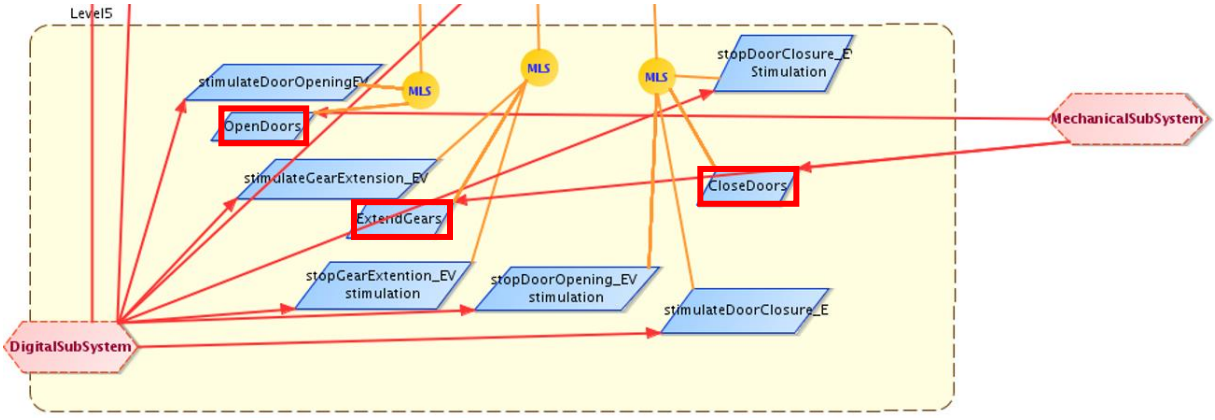


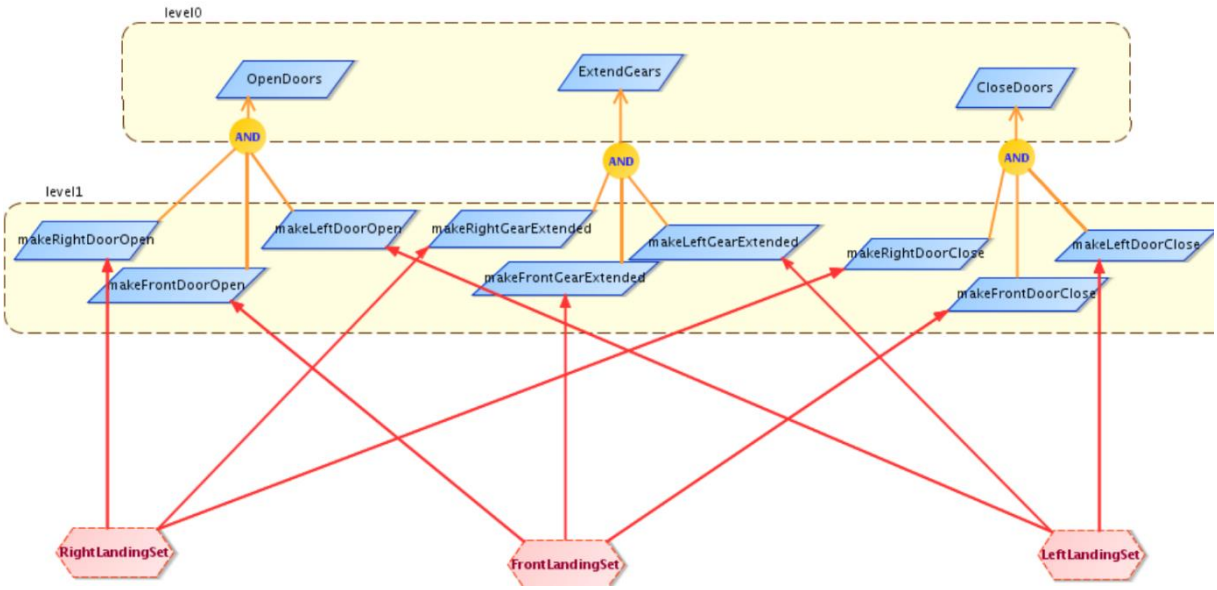



```
SYSTEM
        DigitalSS_Interface
SEES
        LandingGearSystem_CONTEXT,
        LandingGearSystem_CONTEXT_L1,
        LandingGearSystem_CONTEXT_L2,
        LandingGearSystem_CONTEXT_L3,
        LandingGearSystem_CONTEXT_L4
ABSTRACT_VARIABLES
        currentManeuver, lgState, decision, extensionOrder, GEVstate,

        /*********Shared variable used in PiloteSS_Interface*******/
        handleState, //updated in PiloteSS_Interface and read in DigitalSS_Interface

        /*********Shared variables used in MechanicalSS_Interface****/
        DO_EVstate, GE_EVstate, DC_EVstate, //read in MechanicalSS_Interface and updated in
        DigitalSS_Interface

        doorsState, gearsState //updated in MechanicalSS_Interface and read in
        DigitalSS_Interface

INVARIANT
        lgState:LandingGearSystem --> LG_STATES &
        currentManeuver:LandingGearSystem --> MANEUVER_TYPES &
        decision : LandingGearSystem --> Actions &
        handleState : Handle --> HANDLE_STATES &
        extensionOrder : LandingGearSystem --> BOOL &
        GEVstate : ELECTRO_VALVES --> EV_States &
        doorsState : Doors --> Doors_States &
        gearsState : Gears --> Gears_States &
        DO_EVstate :ELECTRO_VALVES --> EV_States &
        GE_EVstate :ELECTRO_VALVES --> EV_States &
        DC_EVstate :ELECTRO_VALVES --> EV_States

INITIALISATION
        lgState::LandingGearSystem --> LG_STATES
        || currentManeuver::LandingGearSystem --> MANEUVER_TYPES
        || decision :: LandingGearSystem --> Actions
        || handleState :: Handle --> HANDLE_STATES
        || extensionOrder :: LandingGearSystem --> BOOL
        || GEVstate :: ELECTRO_VALVES --> EV_States
        || doorsState :: Doors --> Doors_States
        || gearsState :: Gears --> Gears_States
        || DO_EVstate :: ELECTRO_VALVES --> EV_States
        || GE_EVstate :: ELECTRO_VALVES --> EV_States
        || DC_EVstate :: ELECTRO_VALVES --> EV_States
EVENTS

        communicateExtensionOrder =
        SELECT handleState(LgOfHd~(LG))=down THEN
                extensionOrder(LG) :=TRUE
                || decision(LG):=Extension
        END;

        stimulateGeneral_EV =
        SELECT extensionOrder(LG) = TRUE & decision(LG)=Extension THEN
                GEVstate(lgOfG_EV~(LG)):= stimulated
        END;


        stopGeneral_EVstimulation =
        SELECT doorsState(lgOfDRS~(LG)) = close THEN
                GEVstate(lgOfG_EV~(LG)):= stopped
                || lgState(LG):=lg_extended
                || currentManeuver(LG):=landing
        END;
        ////////////////////////////////////////////////////////////////////////////////
        stimulateDoorOpeningEV =
```

```
        SELECT GEVstate(lgOfG_EV~(LG))=stimulated THEN
               DO_EVstate(DRSofDO_EV~(DRS)) := stimulated
        END;

        stimulateGearExtension_EV =
        SELECT doorsState(lgOfDRS~(LG)) = open THEN
               GE_EVstate(GRSofGE_EV~(GRS)):=stimulated
        END;

        stopGearExtension_EV =
        SELECT gearsState(lgOfGRS~(LG)) = extended THEN
               GE_EVstate(GRSofGE_EV~(GRS)):=stopped
        END;
        ////////////////////////////////////////////////////
        stopDoorOpeningEV =
        SELECT gearsState(lgOfGRS~(LG)) = extended THEN
               DO_EVstate(DRSofDO_EV~(DRS)) := stopped
        END;

        stimulateDoorClosure_EV =
        SELECT DO_EVstate(DRSofDO_EV~(DRS)) = stopped THEN
               DC_EVstate(DRSofDC_EV~(DRS)):= stimulated
        END;
        stopDoorClosure_EV =
        SELECT doorsState(lgOfDRS~(LG)) = close THEN
               DC_EVstate(DRSofDC_EV~(DRS)):= stopped
        END
        /******************Shared Events of MechanicalSS_Inteface**********************
        OpenDoors =
        SELECT DO_EVstate(DRSofDO_EV~(DRS)) = stimulated THEN
               doorsState(lgOfDRS~(LG)) := open
        END;

        ExtendGears=
        SELECT GE_EVstate(GRSofGE_EV~(GRS))=stimulated THEN
               gearsState(lgOfGRS~(LG)) := extended
        END;
        closeDoors=
        SELECT DC_EVstate(DRSofDC_EV~(DRS)) = stimulated THEN
               doorsState(lgOfDRS~(LG)) := close
        END
        ********************************************************************************/

        /******************Shared Events of PiloteSS_Inteface**********************

        putHandleDown =
        BEGIN
        handleState(LgOfHd~(LG)):=down
        END;

        putHandleUp =
        BEGIN
        handleState(LgOfHd~(LG)):=up
        END
*************************************************************************************/

END
```

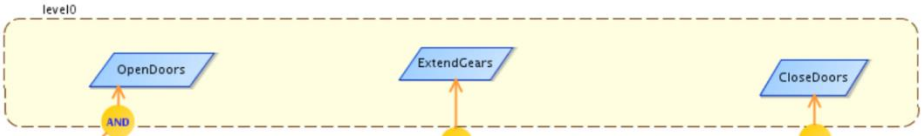
- Sous-Système mécanique :





Le sous-système mécanique est composé de deux niveaux :

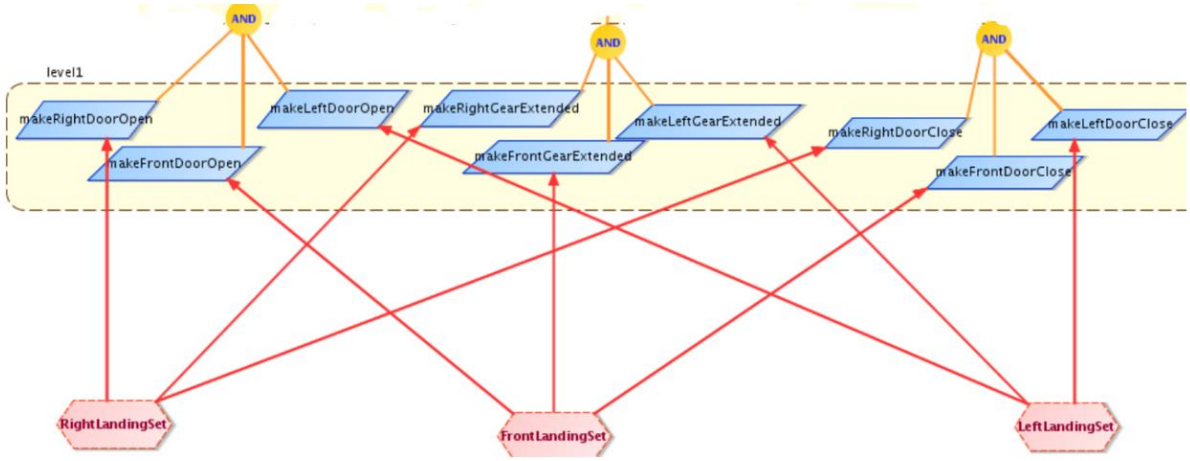
- 1er niveau :




```
SYSTEM
        MechanicalSS_Interface
SEES
        LandingGearSystem_CONTEXT,
        LandingGearSystem_CONTEXT_L3,
        LandingGearSystem_CONTEXT_L4
ABSTRACT_VARIABLES
        /****Shared variables used in DigitalSS_Interface*******/
        doorsState, gearsState, //updated in MechanicalSS_Interface and read in
        DigitalSS_Interface


        DO_EVstate, GE_EVstate, DC_EVstate //read in MechanicalSS_Interface and updated in
        DigitalSS_Interface
INVARIANT
        doorsState : Doors --> Doors_States &
        gearsState : Gears --> Gears_States &
        DO_EVstate :ELECTRO_VALVES --> EV_States &
        GE_EVstate :ELECTRO_VALVES --> EV_States &
        DC_EVstate :ELECTRO_VALVES --> EV_States
INITIALISATION
        doorsState :: Doors --> Doors_States
        || gearsState :: Gears --> Gears_States
        || DO_EVstate :: ELECTRO_VALVES --> EV_States
        || GE_EVstate :: ELECTRO_VALVES --> EV_States
        || DC_EVstate :: ELECTRO_VALVES --> EV_States
EVENTS
        OpenDoors =
        SELECT DO_EVstate(DRSofDO_EV~(DRS)) = stimulated THEN
               doorsState(lgOfDRS~(LG)) := open
        END;
        ExtendGears=
        SELECT GE_EVstate(GRSofGE_EV~(GRS))=stimulated THEN
               gearsState(lgOfGRS~(LG)) := extended
        END;
        closeDoors=
        SELECT DC_EVstate(DRSofDC_EV~(DRS)) = stimulated THEN
               doorsState(lgOfDRS~(LG)) := close
        END
        /******************Shared Events of DigitalSS_Interface********************
        stopGeneral_EVstimulation =
        SELECT doorsState(lgOfDRS~(LG)) = close THEN
               GEVstate(lgOfG_EV~(LG)):= stopped
               || lgState(LG):=lg_extended
               || currentManeuver(LG):=landing
        END;
        ////////////////////////////////////////////////////////////////////////////////
        stimulateDoorOpeningEV =
        SELECT GEVstate(lgOfG_EV~(LG))=stimulated THEN
```

```
            DO_EVstate(DRSofDO_EV~(DRS)) := stimulated
        END;
        stimulateGearExtension_EV =
        SELECT doorsState(lgOfDRS~(LG)) = open THEN
            GE_EVstate(GRSofGE_EV~(GRS)):=stimulated
        END;
        stopGearExtension_EV =
        SELECT gearsState(lgOfGRS~(LG)) = extended THEN
            GE_EVstate(GRSofGE_EV~(GRS)):=stopped
        END;
        ////////////////////////////////////////////////////
        stopDoorOpeningEV =
        SELECT gearsState(lgOfGRS~(LG)) = extended THEN
        DO_EVstate(DRSofDO_EV~(DRS)) := stopped
        END;

        stimulateDoorClosure_EV =
        SELECT DO_EVstate(DRSofDO_EV~(DRS)) = stopped THEN
        DC_EVstate(DRSofDC_EV~(DRS)):= stimulated
        END;
        stopDoorClosure_EV =
        SELECT doorsState(lgOfDRS~(LG)) = close THEN
        DC_EVstate(DRSofDC_EV~(DRS)):= stopped
        END
        ********************************************************************************/
END
```

- 2éme niveau :




```
SYSTEM
        LandingGearSystem_MechanicalSS_CONTEXT
SEES
        LandingGearSystem_CONTEXT_L3
SETS
        SIDES_DOORS; SIDES_GEARS
CONSTANTS
        leftD, DRSofLD,
        rightD, DRSofRD,
        frontD, DRSofFD,
        leftG, GRSofLG,
        rightG, GRSofRG,
        frontG, GRSofFG
PROPERTIES
        leftD:SIDES_DOORS&
        DRSofLD: SIDES_DOORS >->> Doors &
        DRSofLD = {leftD|->DRS} &

        rightD:SIDES_DOORS&
        DRSofRD: SIDES_DOORS >->> Doors &
        DRSofRD = {rightD|->DRS} &

        frontD:SIDES_DOORS&
        DRSofFD: SIDES_DOORS >->> Doors &
        DRSofFD = {frontD|->DRS} &
```

```
        leftD /= rightD&
        rightD/=frontD&
        leftD/=frontD&
        SIDES_DOORS={leftD,rightD,frontD}&

        leftG:SIDES_GEARS&
        GRSofLG: SIDES_GEARS >->> Gears &
        GRSofLG = {leftG|->GRS} &

        rightG:SIDES_GEARS&
        GRSofRG: SIDES_GEARS >->> Gears &
        GRSofRG = {rightG|->GRS} &

        frontG:SIDES_GEARS&
        GRSofFG: SIDES_GEARS >->> Gears &
        GRSofFG = {frontG|->GRS} &

        leftG /= rightG&
        rightG/=frontG&
        leftG/=frontG&
        SIDES_GEARS={leftG,rightG,frontG}
    END
```

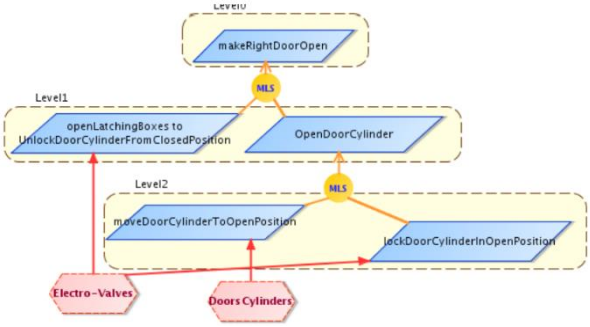
```
REFINEMENT MechanicalSS_L1
REFINES MechanicalSS_Interface
SEES
        LandingGearSystem_CONTEXT,
        LandingGearSystem_CONTEXT_L3,
        LandingGearSystem_CONTEXT_L4,
        LandingGearSystem_MechanicalSS_CONTEXT

ABSTRACT_VARIABLES
        DO_EVstate,
        GE_EVstate,
        DC_EVstate,
        doorsState, gearsState,
        sides_doorState, sides_gearState
INVARIANT
        sides_doorState: SIDES_DOORS --> Doors_States &
        sides_gearState: SIDES_GEARS --> Gears_States &
        (sides_doorState[SIDES_DOORS]={open}=> doorsState[Doors]={open}) &
        (sides_doorState[SIDES_DOORS]={close}=> doorsState[Doors]={close}) &
        (sides_gearState[SIDES_GEARS]={extended}=> gearsState[Gears]={extended})
INITIALISATION
        DO_EVstate :: ELECTRO_VALVES --> EV_States
        || GE_EVstate :: ELECTRO_VALVES --> EV_States
        || DC_EVstate :: ELECTRO_VALVES --> EV_States

        || doorsState, sides_doorState:
        ((sides_doorState[SIDES_DOORS]={open}=> doorsState[Doors]={open}) &
         (sides_doorState[SIDES_DOORS]={close}=> doorsState[Doors]={close}) &
        (doorsState : Doors --> Doors_States) &
        (sides_doorState : SIDES_DOORS --> Doors_States))

        || gearsState, sides_gearState:
        ((sides_gearState[SIDES_GEARS]={extended}=> gearsState[Gears]={extended}) &
        (gearsState : Gears --> Gears_States) &
         (sides_gearState : SIDES_GEARS --> Gears_States))
EVENTS
        makeRightDoorOpen ref_and OpenDoors =
        SELECT DO_EVstate(DRSofDO_EV~(DRS)) = stimulated THEN
                sides_doorState(DRSofRD~(DRS)):=open
        END;

        makeLeftDoorOpen ref_and OpenDoors =
        SELECT DO_EVstate(DRSofDO_EV~(DRS)) = stimulated THEN
                sides_doorState(DRSofLD~(DRS)):=open
        END;

        makeFrontDoorOpen ref_and OpenDoors =
        SELECT DO_EVstate(DRSofDO_EV~(DRS)) = stimulated THEN
                sides_doorState(DRSofFD~(DRS)):=open
        END;
```

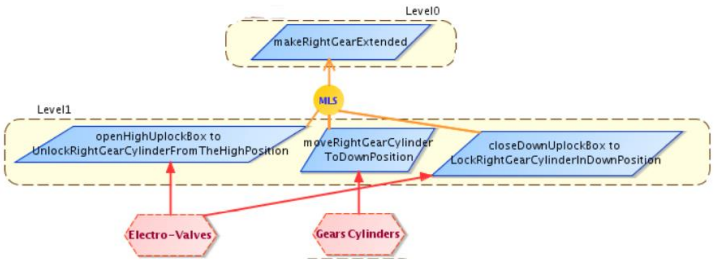
```
        makeRightGearExtended ref_and ExtendGears =
        SELECT GE_EVstate(GRSofGE_EV~(GRS))=stimulated THEN
               sides_gearState(GRSofRG~(GRS)):=extended
        END;

        makeLeftGearExtended ref_and ExtendGears =
        SELECT GE_EVstate(GRSofGE_EV~(GRS))=stimulated THEN
               sides_gearState(GRSofLG~(GRS)):=extended
        END;

        makeFrontGearExtended ref_and ExtendGears =
        SELECT GE_EVstate(GRSofGE_EV~(GRS))=stimulated THEN
               sides_gearState(GRSofFG~(GRS)):=extended
        END;

        makeRightDoorClose ref_and closeDoors =
        SELECT DC_EVstate(DRSofDC_EV~(DRS)) = stimulated THEN
               sides_doorState(DRSofRD~(DRS)):=close
        END;

        makeLeftDoorClose ref_and closeDoors =
        SELECT DC_EVstate(DRSofDC_EV~(DRS)) = stimulated THEN
               sides_doorState(DRSofLD~(DRS)):=close
        END;

        makeFrontDoorClose ref_and closeDoors =
        SELECT DC_EVstate(DRSofDC_EV~(DRS)) = stimulated THEN
               sides_doorState(DRSofFD~(DRS)):=close
        END
END
```

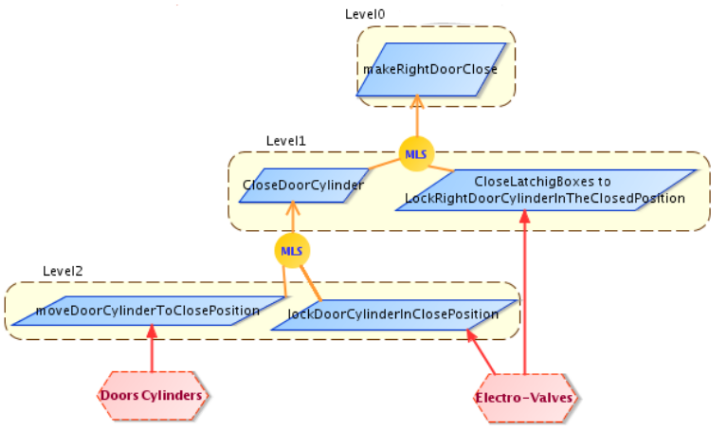
Décomposition sous-système mécanique en 3 sous-systèmes :

- Right Landing Set,
- Front Landing Set,
- Left Landing Set.

Modèle SysML/KAOS du « Right Landing Set » :

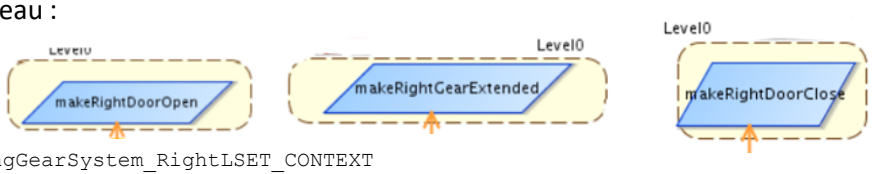






Le sous-système « Right Landing Set » est composé de 3 niveaux :

- 1er niveau :

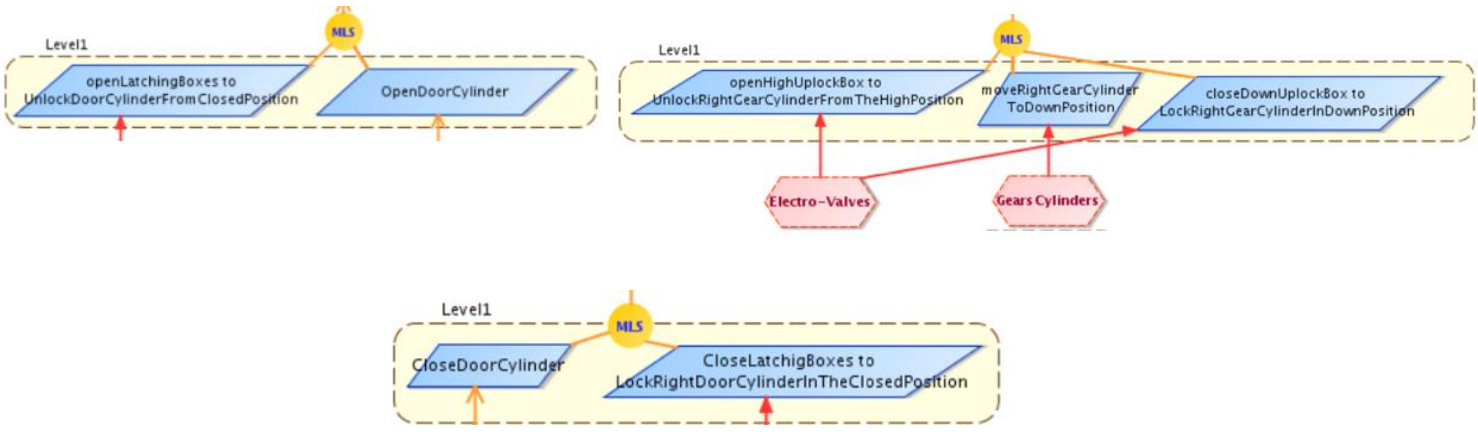



```
SYSTEM
        LandingGearSystem_RightLSET_CONTEXT
SEES
        LandingGearSystem_MechanicalSS_CONTEXT
SETS
        LatchingBoxes;
        LatchingBoxes_States;
        DoorsCylinders;
        DC_POSITION;
        C_States;
        HighUplockBox;
        DownUplockBox;
        UB_STATES;
        GearsCylinders;
        GC_POSITION
CONSTANTS
        LB, DRSofLB,
        opened, closed,
        DC, DRSofDCyl,
        closedPosition,
        openPosition,
        lock, unlock,
        HighUB, GRSofHUB,
        DownUB, GRSofDUB,
        openedBox, closedBox,
        GC, GRSofGCyl,
        highPosition, downPosition
PROPERTIES
        LB:LatchingBoxes &
        LatchingBoxes={LB} &
        DRSofLB: LatchingBoxes >->> SIDES_DOORS &
        DRSofLB={LB|->rightD} &
        opened:LatchingBoxes_States &
        closed: LatchingBoxes_States &
        closed/=opened &
        LatchingBoxes_States={closed, opened} &


        DC: DoorsCylinders &
        DoorsCylinders={DC} &
        DRSofDCyl: DoorsCylinders >->> SIDES_DOORS &
        DRSofDCyl={DC|->rightD} &
        closedPosition: DC_POSITION &
        openPosition : DC_POSITION &
        closedPosition /= openPosition &
        DC_POSITION={closedPosition, openPosition} &
        lock:C_States &
        unlock: C_States &
        lock /=unlock &
        C_States={lock,unlock}&


        HighUB: HighUplockBox &
        HighUplockBox={HighUB}&
        GRSofHUB: HighUplockBox >->> SIDES_GEARS &
        GRSofHUB={HighUB|-> rightG} &
        DownUB:DownUplockBox &
        DownUplockBox={DownUB}&
        GRSofDUB: DownUplockBox >->> SIDES_GEARS &
        GRSofDUB={DownUB |-> rightG} &


        GC: GearsCylinders &
        GearsCylinders={GC} &
        GRSofGCyl: GearsCylinders >->> SIDES_GEARS &
        GRSofGCyl={GC |-> rightG} &
        highPosition: GC_POSITION &
        downPosition : GC_POSITION&
```

```
        highPosition/= downPosition &
        GC_POSITION={highPosition, downPosition}&

        openedBox:UB_STATES&
        closedBox:UB_STATES&
        openedBox /= closedBox &
        UB_STATES={openedBox,closedBox}
END
```

```
SYSTEM
        RightLandingSet_Interface
SEES
        LandingGearSystem_CONTEXT_L3,
        LandingGearSystem_CONTEXT_L4,
        LandingGearSystem_MechanicalSS_CONTEXT
ABSTRACT_VARIABLES
        DO_EVstate,
        GE_EVstate,
        DC_EVstate,
        sides_doorState, sides_gearState
INVARIANT
        DO_EVstate :ELECTRO_VALVES --> EV_States &
        GE_EVstate :ELECTRO_VALVES --> EV_States &
        DC_EVstate :ELECTRO_VALVES --> EV_States &
        sides_doorState: SIDES_DOORS --> Doors_States &
        sides_gearState: SIDES_GEARS --> Gears_States
INITIALISATION
        DO_EVstate :: ELECTRO_VALVES --> EV_States
        || GE_EVstate :: ELECTRO_VALVES --> EV_States
        || DC_EVstate :: ELECTRO_VALVES --> EV_States
        || sides_doorState :: SIDES_DOORS --> Doors_States
        || sides_gearState :: SIDES_GEARS --> Gears_States
EVENTS
        makeRightDoorOpen =
        SELECT DO_EVstate(DRSofDO_EV~(DRS)) = stimulated THEN
                sides_doorState(DRSofRD~(DRS)):=open
        END;

        makeRightGearExtended =
        SELECT GE_EVstate(GRSofGE_EV~(GRS))=stimulated THEN
                sides_gearState(GRSofRG~(GRS)):=extended
        END;

        makeRightDoorClose =
        SELECT DC_EVstate(DRSofDC_EV~(DRS)) = stimulated THEN
                sides_doorState(DRSofRD~(DRS)):=close
        END
END
```

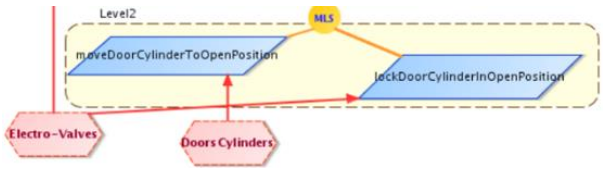
- 2éme niveau :



```
REFINEMENT RightLandingSet_L1
REFINES RightLandingSet_Interface
SEES
        LandingGearSystem_CONTEXT_L3,
        LandingGearSystem_CONTEXT_L4,
        LandingGearSystem_MechanicalSS_CONTEXT,
        LandingGearSystem_RightLSET_CONTEXT
ABSTRACT_VARIABLES
        DO_EVstate,
        GE_EVstate,
        DC_EVstate,
        sides_doorState, sides_gearState,
        LBState, dcPos, dcState,
        hubState,gcPos, gcState,dubState
INVARIANT
        LBState: LatchingBoxes --> LatchingBoxes_States &
        dcPos: DoorsCylinders --> DC_POSITION &
        dcState: DoorsCylinders --> C_States &
        hubState: HighUplockBox --> UB_STATES &
        gcPos: GearsCylinders --> GC_POSITION &
        gcState: GearsCylinders --> C_States &
        dubState: DownUplockBox --> UB_STATES

INITIALISATION
        DO_EVstate :: ELECTRO_VALVES --> EV_States
        || GE_EVstate :: ELECTRO_VALVES --> EV_States
        || DC_EVstate :: ELECTRO_VALVES --> EV_States
        || sides_doorState :: SIDES_DOORS --> Doors_States
        || sides_gearState :: SIDES_GEARS --> Gears_States
        || LBState:: LatchingBoxes --> LatchingBoxes_States
        || dcPos:: DoorsCylinders --> DC_POSITION
        || dcState:: DoorsCylinders --> C_States
        || hubState:: HighUplockBox --> UB_STATES
        || gcPos:: GearsCylinders --> GC_POSITION
        || gcState:: GearsCylinders --> C_States
        || dubState:: DownUplockBox --> UB_STATES
EVENTS
        OpenLatchingBoxesForDoorCylinderUnlck ref_milestone makeRightDoorOpen =
        SELECT DO_EVstate(DRSofDO_EV~(DRS)) = stimulated THEN
                LBState(DRSofLB~(rightD)):=opened
                || dcState(DRSofDCyl~(rightD)):=unlock
        END;


        openDoorCylinder ref_milestone makeRightDoorOpen =
        SELECT LBState(DRSofLB~(rightD))=opened & dcState(DRSofDCyl~(rightD))=unlock THEN
                sides_doorState(DRSofRD~(DRS)):=open || dcPos(DRSofDCyl~(rightD)):=openPosition
                || dcState(DRSofDCyl~(rightD)):=lock
        END;

        OpenHighUplockBoxForGearCylinderUnlck ref_milestone makeRightGearExtended =
        SELECT GE_EVstate(GRSofGE_EV~(GRS))=stimulated THEN
                hubState(GRSofHUB~(rightG)):=openedBox
                ||gcState(GRSofGCyl~(rightG)):=unlock
        END;


        moveGearCylinder ref_milestone makeRightGearExtended =
        SELECT hubState(GRSofHUB~(rightG))=openedBox & gcState(GRSofGCyl~(rightG))=unlock THEN
                gcPos(GRSofGCyl~(rightG)):= downPosition
        END;


        closeDownUplockBoxForGearCylinderLock ref_milestone makeRightGearExtended =
        SELECT gcPos(GRSofGCyl~(rightG))= downPosition THEN
                dubState(GRSofDUB~(rightG)):=closedBox
                || gcState(GRSofGCyl~(rightG)):=lock
                || sides_gearState(GRSofRG~(GRS)):=extended
        END;

        closeDoorCylinder ref_milestone makeRightDoorClose =
        SELECT DC_EVstate(DRSofDC_EV~(DRS)) = stimulated THEN
                dcPos(DRSofDCyl~(rightD)):=closedPosition ||
                sides_doorState(DRSofRD~(DRS)):=close || dcState(DRSofDCyl~(rightD)):=lock
        END;
```

```
        closeLatchingBoxesForDoorCylinderLock ref_milestone makeRightDoorClose =
        SELECT sides_doorState(DRSofRD~(DRS))=close THEN
                LBState(DRSofLB~(rightD)):=closed
        END
END
```

- 3éme niveau :




```
REFINEMENT RightLandingSet_L2
REFINES RightLandingSet_L1
SEES
        LandingGearSystem_CONTEXT_L3,
        LandingGearSystem_CONTEXT_L4,
        LandingGearSystem_MechanicalSS_CONTEXT,
        LandingGearSystem_RightLSET_CONTEXT
ABSTRACT_VARIABLES
        DO_EVstate,
        GE_EVstate,
        DC_EVstate,
        sides_doorState, sides_gearState,
        LBState, dcPos, dcState,
        hubState,gcPos, gcState,dubState
INITIALISATION
        DO_EVstate :: ELECTRO_VALVES --> EV_States
        || GE_EVstate :: ELECTRO_VALVES --> EV_States
        || DC_EVstate :: ELECTRO_VALVES --> EV_States
        || sides_doorState :: SIDES_DOORS --> Doors_States
        || sides_gearState :: SIDES_GEARS --> Gears_States
        || LBState:: LatchingBoxes --> LatchingBoxes_States
        || dcPos:: DoorsCylinders --> DC_POSITION
        || dcState:: DoorsCylinders --> C_States
        || hubState:: HighUplockBox --> UB_STATES
        || gcPos:: GearsCylinders --> GC_POSITION
        || gcState:: GearsCylinders --> C_States
        || dubState:: DownUplockBox --> UB_STATES


EVENTS
        moveDoorCylinderToOpenPosition ref_milestone openDoorCylinder =
        SELECT LBState(DRSofLB~(rightD)) = opened & dcState(DRSofDCyl~(rightD))=unlock THEN
                dcPos(DRSofDCyl~(rightD)):=openPosition
                || sides_doorState(DRSofRD~(DRS)):=open
        END;

        lookDoorCylinderinOpenPosition ref_milestone openDoorCylinder =
        SELECT dcPos(DRSofDCyl~(rightD))=openPosition THEN
                dcState(DRSofDCyl~(rightD)):=lock
        END;
        /////
        moveDoorCylinderToClosePosition ref_milestone closeDoorCylinder =
        SELECT DC_EVstate(DRSofDC_EV~(DRS)) = stimulated THEN
                dcPos(DRSofDCyl~(rightD)):=closedPosition
                || sides_doorState(DRSofRD~(DRS)):=close
        END ;

        lookDoorCylinderinClosedPosition ref_milestone closeDoorCylinder =
        SELECT dcPos(DRSofDCyl~(rightD))=closedPosition THEN
                dcState(DRSofDCyl~(rightD)):=lock
        END
END
```