# Introduction

This note captures some observations and lessons from work relating to the specification and system modeling of the open landing gear case study [[1]](https://paperpile.com/c/ilxpn9/v4pe).We address many of the research questions explored under the Aerospace Vehicle System Institute (AVSI) AFE 86 research project, and augment these research findings with additional modeling and specification work performed with industrial research partners.

The primary goal of the AVSI AFE 86 project is to understand how property-based specification approaches can be effectively applied within assurance-oriented Model-based System Engineering (MBSE) workstreams. Specifically, the following research questions are of interest

* What type of requirement is amenable to property-based specification?
* Is property-based specification sufficient?
  + Or do we also need to have *actionable requirements* [[2]](https://paperpile.com/c/ilxpn9/QmrX)?
* How can we systematically integrate property-based requirements with the 'data dictionary/information model'?
  + Can SysMLv2 serve as a suitable foundation for such a model?
* How does property-based specification augment and complement existing best-practice requirement engineering guidance, such as the FAA Requirement Management Handbook [[3]](https://paperpile.com/c/ilxpn9/VtmB)?
* How can we systematically manage and link different levels of abstraction and functional decomposition?
  + How can we support and link heterogenous specifications and models?
* How can we better tie and integrate V&V campaigns to the operational scenarios?
  + Can we leverage such integration toward more automated workflows?
  + How can we develop systematic verification and validation coverage metrics?
  + Can we realize systematic guidance for model-based intent coverage metrics to systematically capture how well V&V campaigns cover the required behaviors within environmental, operational and failure assumptions?
* How can we provide systematic guidance with respect to HLR and LLR specifications?
  + Are the specific attributes that are necessary toward the validation of HLR/LLR sufficient?
    - For example, should architectural orchestration, such as channel in control logic specification be a necessary attribute of HLR baselines?
* How can we better manage cyber-physical system refinement?
  + How can we develop some better insight with respect to the boundary of specification with respect to the cyber-and physical system properties?
    - How are physical properties and logical properties best integrated?
* How can we develop and refine systematic guidance to aid architectural refinement?
  + How do we express and integrate architectural requirements and architectural orchestration behavior?
  + How do we link the architectural requirements with the system safety and hazard processes?
  + How do we justify architectural validation?
    - With respect to the case study, the sufficiency of the architectural integrity may warrant a longer discussion.

Additionally, as we worked through the work described herein, the following additional questions become apparent (not yet explored under the current work)?

* How can we develop and illustrate different levels of functional refinement?
  + Consider a PID control loop controlling the LGS extension
    - How would such a specification be different from the discrete control of the hydraulic-driven actuation in the basic case study?
    - How would the top-level abstractions bind to the lower levels of functional and behavioral refinement?
* How & can such properties be formally validated?
  + What is the preferred form of the flow down to software?
  + At what point does a model become a model?

# Exploring Information Model Binding

We have included the following look ahead vision as a quick aside to illustrate how an integrated information model context may be useful should it yield the necessary semantic traceability through different levels of abstraction/refinement. As part of the work we worked bottom up from a physics model and in parallel explored top-down functional decomposition and refinement of the system context (ideally to eventually link and include operational and system analysis). Our key goal is to understand information across the entire MBSE workflow that can be integrated to better support a more holistic property-driven development workflow toward more automated and computer/knowledge-assisted system validation and verification. We believe that PMM provides a very strong foundation on which to build such a workflow.

However, at this time, we are still in the process of generating and integrating the different model sets which are intended to comprise

* functional model within the context of the system operational scenarios (Capella/OPM)
* logical behavioral model of the digital control (OPM/EARS/ASSERT based requirements)
* The physical behavioral model of the system hydraulics (VHDL-AMS)

Our goal is to explore how different elements of the information model can fuse within the MBSE workflow, in order to provide improved semantic traceability and parameterized property-based specification (and V&V) throughout the different levels of abstraction.

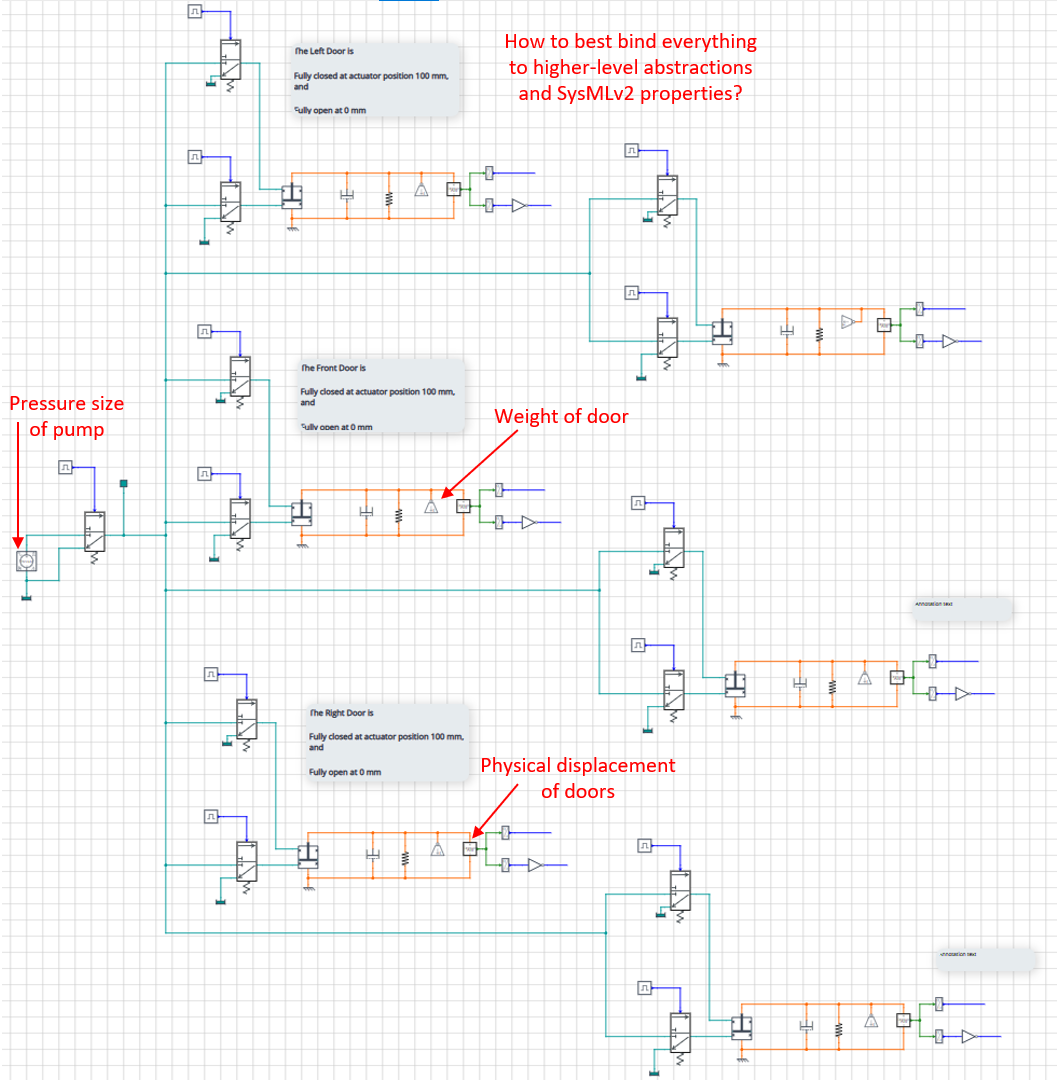
As part of this work we intend to explore the fusion of the OPM, Capella/SysMLv2 and physical behavioral (VHDL-AMS) models. Ideally we want to

* link functional chains of the LGS to the operational scenarios within Capella
  + this may serve as the basis of system scenario capture toward simulation scenario generation
* link higher-level functional abstractions and properties to lower-level *concretized* physical and behavioral properties
  + Such as abstracted functional states with respect to parameterized physical observations
    - With a goal of exploring how properties and property preservation can be discharged throughout different levels of abstraction/refinement.

Please note, the above notations and technologies are of secondary interest, and as such our key focus is what information is captured rather than any specific notational format. That said, we have found the different perspectives that are driven from this use of the different notations to be interesting with respect to the kind of information elicited by the different modeling perspectives.

Ideally, we also want to link physical properties, such as the landing gear door mass, to the parameterized behavioral simulation of the lower level models. This will provide an idea backdrop from which to explore:

* property binding and aggregated of behavioral contribution
  + The load comprises all three door loads
* binding of the functional abstraction
  + E.g. the functional ‘open door state’ to the positional state observers of the physical landing gear doors.
* Scenario binding
  + Where operational scenarios within the operational analysis use-cases may support the simulation based validation of the required system properties
    - In our case study, this is the landing abort scenario, which forces an immediate retraction of the landing gear following the requested extension, which may help uncover requirement weakness. (See later discussion)



Draft Physical Model of LGS – note the latest has been updated to match the functional fanout of the valves. We want to also tie this to the functional model in Capella/SysMLv2

As an end goal, and a possible follow-on of the AFE 86 work, we want to link the property-based V&V with emerging open simulation and instrumentation standards such as FMI and VISAS. This may also integrate with the recent adjacent work [[4], [5]](https://paperpile.com/c/ilxpn9/LIK5+ojYA)

## Tailoring EARS for Practical Use

In prior work we had explored the integration of property-model methodology and EARS within MIDAS [[2]](https://paperpile.com/c/ilxpn9/QmrX). As part of the landing gear case study the goal is to continue to explore the application of EARS [[2], [6], [7]](https://paperpile.com/c/ilxpn9/QmrX+XcZW+OKYw) to requirement specification within property-driven requirements flows. EARS promotes focus and better understanding of event and state semantics, as well the systematic treatment of nominal and off nominal scenarios. We have found such a focus to be valuable with respect to clarifying system behavior. However, in prior experiments, we have also found that the unguided application of EARS can be potentially problematic should 'composite events' not be properly recognized. A 'composite event' occurs when multiple conditions are necessary to become true in order for a trigger to occur, but the order of the events is not important.

That is to say, that it is not the case that any of the conditions act as a guard on the other triggering events.

We have found that if not properly recognized composite events may cause EARS requirements to expand and/or inadvertently miss key triggering conditions.

In our work here-in, we have found the combination of *property-based thinking* with EARS-based actionable requirements to be a promising direction (see later).

## Exploring System-Level Requirements

Following a review of adjacent work [[8]](https://paperpile.com/c/ilxpn9/A50M), which was largely focused on the software process refinement, as part of our work we initially stepped back to revisit the system level. This was driven with the following goals in mind:

* As discussed above, as part of the work we want to explore how the system to software refinement can be better illustrated/facilitated. With respect to the current case-study this is not straightforward, given the software context of the example. When we consider the initial requirements in the case study levels of the specification are indeed mixed. See the excerpt below.

The blue text refers to the visible externals of the system, yet the red text refers to the internals of the system. Hence, within our revised property-driven requirements a primary goal is to more cleanly separate and capture the respective boundaries of intent, especially regarding cyber-property requirements, and physical property properties.

With respect to computer-assisted V&V how functional, physical and logical properties can be effectively integrated through multiple levels of system refinement is a prime research interest.

With respect to software, the prior guidance within the FAA Requirement Engineering Management Handbook and the recommended use and extension of Parnas’s Four Variable model [[9]](https://paperpile.com/c/ilxpn9/COOK) may blend well with property-driven specification workflows.

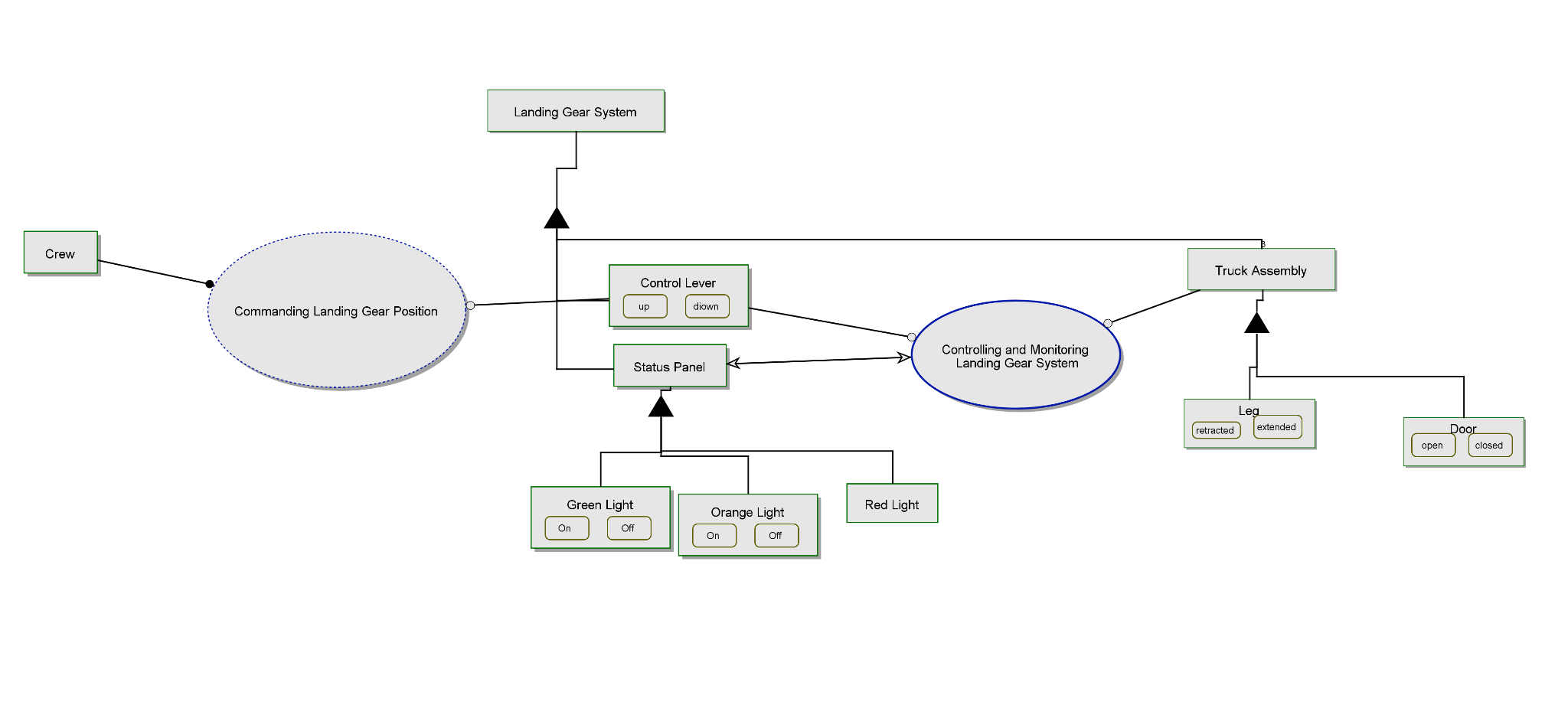
One related research question is how such coupling between logical and physical behaviors is best expressed at the high levels of system specification.

* From our work today, we also believe that the system specification may be greatly simplified by better factoring in the intent at the system level; understanding that system behavior is emergent of the system components. For example, within the landing gear case study the opening of the door is agnostic with respect to gear extension or retraction need. A research question related to factoring requirements may aid specification simplification. For example, is it preferred to specify sequences of behaviors or just properties that relate to components.
* We also believe that the treatment of architecturally related orchestration and refinement was insufficient within prior work. Hence, this refinement may also benefit from more systematic treatment within this case-study, given the multiple channels of control authority and the relatively tight temporal constraints on valve actuation and sequencing.

*Note that two control channels operating fully independently are unlikely to be able to satisfy such constraints. Hence some form of channel orchestration appears necessary. How such orchestration can be mapped into the requirement flow-down is there for an area of exploration.*

## System-Level Properties

To assist with specification, the initial draft of the physical system boundary using OPM is shown below.



We also introduce the conceptual state of the Landing Gear System State to denote when landing gear position adjustment is necessary/complete.

| Control Lever | Aggregated Landing Gear Truck State | Landing Gear SubFunction State | Landing Gear System State |
| --- | --- | --- | --- |
| UP | Not all Retracted and locked | Retraction Adjustment In Progress | Adjustment in Progress |
| DOWN | Not all Retracted and locked | Extension Adjustment In Progress | Adjustment in Progress |
| UP | All Retracted and locked | Retraction Adjustment Complete | Adjustment Complete |
| DOWN | All Extended and locked | Extension Adjustment Complete | Adjustment Complete |

*Note: One area of consideration is the degree of control authority. For example, the transitions between locked and locked states of the doors and gears are solely under the purview of the hydraulic system and therefore beyond the influence of the digital sequencing logic. How such separation is captured is an interesting consideration. However, at the higher level, such details can be abstracted as the conceptual system* final *states and introduced within the information mode.*

*The external system properties are first defined within the context of this information model.*

*[P\_ACT\_1] All Landing Gear Truck Assemblies shall be 'Extended and Locked' and all Landing Gear Doors shall be ‘Closed and Locked’ while all the following conditions hold*

* *Aircraft is not on Ground*
* *Control Lever has been Down for at least 15 seconds*
* *No failure exists*
  + *Not sure if this is needed or can be assumed from a property synthesis perspective it is a necessary precondition*

*[P\_ACT\_2] All Landing Gear Truck Assemblies shall be 'Retracted and Locked' and all Landing Gear Doors shall be ‘Closed and Locked’ while all the following conditions hold*

* *Aircraft is not on Ground*
* *Control Lever has been Up for at least 15 seconds*
* *No failure exists*

*[P\_ACT\_3] Physical Retraction/Extension (movement) of the Landing Gear Track Assemblies shall only occur when the Landing Gear Doors are Fully Open.*

*[P\_SAT\_1] The Orange Light of the Status Panel shall be illuminated while the following conditions hold*

* *Aircraft is not on Ground*
* *Landing Gear System State has been 'Extension In Progress' for less than 15 seconds*

*OR*

* *Landing Gear System State has been 'Retaction In Progress' for less than 15 seconds*

*[P\_STAT\_2] The Red Light of the Status Panel shall be illuminated while any of the following conditions exist*

* *Landing Gear System State has been 'Adjustment In Progress' for greater than 7 seconds and all Landing Gear Doors are ‘Not Open’.*
* *Landing Gear System State has been 'Adjustment Complete' for greater than 7 seconds and all Landing Gear Doors are ‘Not Closed’.*
* *Landing Gear System State has been 'Adjustment Complete' for greater than 15 seconds and all Landing Gear Doors are ‘Not Closed’.*

*[P\_STAT\_3] The Green Light of the Status Panel shall be illuminated while*

* *All truck assemblies ares ‘Extended and Locked’ and all doors have become ‘Closed and Locked’ within 15 seconds of the Control Lever becoming ‘Down’.*

Note that all of the above can be discharged without any knowledge of the internal system design and be assessed using external observations of the system only. Note that the added constraint of P\_ACT\_3 implies the required sequencing dependency, which by implication means that it is necessary to open the doors before extending or retracting the gear.

With respect to our basic research question “Are such property-driven requirements sufficient to serve as specification or is more specific behavioral specification necessary?”

The above properties should be sufficient to constitute the required observers necessary to discharge partial system correctness. But the implied behavior of the door is not explicit.

## 

## Separation of Logical/Physical and Architectural Properties

At the next level of refinement, we need to ascertain the functionality that will satisfy these properties. From a functional perspective, we may need to define the required functional chains. As we refine the top-level decomposition we may also need to consider the logical and physical functional exchanges (functional chain interactions).

From the functional perspective we may decompose the system into the following.

**Physical Functions**

1. The Mechanical Actuation and Positioning of the Landing Gear Doors and Legs
   * Opening Landing Gear Doors
   * Closing Landing Gear Doors
   * Extending Landing Gear Truck Assemblies
   * Retracting Landing Gear Truck Assemblies
2. Visual Indication of System Status
   * Gear Operation is In Progress
   * All Gears are Down and Locked
   * System has failed Failure

**Logical Functions**

1. The logical orchestration of the mechanical actuation
   * The generation of control signals to coordinate the mechanical actuation
   * Open door fluent
   * Close door fluent
   * Extend gear fluent
   * Retract gear fluent
2. The Monitoring and Collection of System Status
   * The monitoring and generation of status signals to support 2.

* Gear moving fluent
* Gear locked fluent
* System Fail fluent

**Architectural Aspect**

1. In addition to the logical and physical functional chains may also need to capture the use of a mechanical force interlock, to prevent inadvertent mechanical actuation resulting from failures within the logical control orchestration.

Note that at this higher-level we may not have decided on the application of hydraulic actuation technology. Hence, the detailed design decisions with respect to the use of hydraulic actuation, mechanical hydraulic force interlocking may not yet become visible within the specification. Similarly, at this black box level of specification we have not yet refined or allocated functional behavior to software and hardware. Hence, it may be premature to introduce Parnas’s Four-Variable Model at this stage of refinement.

However, the interaction of logical and physical fluents is an important consideration. Are such interactions event- or state-based? Such aspects may be important to fully define the life-cycle of fluents, i.e. when fluents initiate and terminate.

In the context of this case-study the design decision not to lock the doors open has a significant impact on the control orchestration; given that it requires the door fluent to be maintained in order to hold the door open. Although this may be a detailed design decision, it is an interesting attribute to explore with regard to how such designs flow down.

Given this context we define a set of actionable requirements below.

## System Actionable Requirements

*First a naive set of requirements. The goal is to capture the required orchestration of the physical processes. In reality, the properties of the physical processes, such as the time needed to complete the mechanical door opening gear adjustment, would also be required. At this stage we omit such matters for completeness. They are introduced later as the requirement is refined.*

**SYS\_REQ\_1** While the aircraft is in the air, when the following control lever transitions to down, the Landing Gear Control System shall initiate the mechanical opening of all landing gear doors within X ms.

**SYS\_REQ\_2** While the aircraft is in the air and the control lever is down when all landing gear doors are confirmed to be open, the Landing Gear Control system shall initiate the mechanical extension of all landing gear truck assemblies within x ms..

**SYS\_REQ\_3** While the aircraft is in the air and the control lever is down when all landing gear truck assemblies are confirmed to be down and locked, the Landing Gear Control system shall terminate the mechanical opening of all landing doors within X ms, before initiation the mechanical closure of all landing gear doors within Y ms.

**SYS\_REQ\_4** While the aircraft is in the air when the control lever transitions to up, the Landing Gear Control System shall initiate the mechanical opening of all landing gear doors within X ms

**SYS\_REQ\_5** While the aircraft is in the air and the control lever is up, when all landing gear doors are confirmed to be open, the Landing Gear Control system shall initiate the mechanical retraction of all landing gear truck assemblies within X ms.

**SYS\_REQ\_6** While the aircraft is In air and the Control Lever is up when all landing gear truck assemblies are confirmed to be up and locked the Landing Gear Control system shall terminate the mechanical opening of all landing doors within X ms, before initiation the mechanical closure of all landing gear doors within Y ms.

*At first sight the above look promising at first sight. However, when we consider the full context of the operational scenarios and the potential for an aborted retraction/extension, we may find that the composite guard conditions such as ‘*While the Aircraft is In Air and the Control Lever is Up’ may be too restricting. As a guard condition the control lever state would not trigger a reaction in such scenarios, and hence the system may be unresponsive.

*Aside: Eventually, we aim to discover such requirement issues by utilizing the formal specification of operational scenarios in conjunction with the system properties and system behaviors (and possibly anticipated component failure mode). Should time permit, we may demonstrate such a problem discovery by building on the anwer-set based analysis of the event calculus model of the scenarios, and requirements However, at this time we rely on mamua review to uncover such issues and also look toward how better guidance, such as property-driven thing may avoid them,*

To explore the dynamics, we now take a property-based approach to the same requirements. Our goal is to better explore the subtle differences between the two forms of specification, and also better understand how each style of specification can better guide an examination of conscience. Within the property-based specification we first look to the composite state conditions that define the context within fluent (properties) are expected to hold.

For example, for the door commands we explore what at the necessary conditions during which the door open command should be asserted

**[PROP\_1]** Doors are commanded open **only** while ->

aircraft is in the air

and

(control lever is up and all truck assemblies are not retracted and locked

Or

control lever is down and all truck assemblies are not extended and locked)

**[PROP\_2]** Doors are commanded close **only** while ->

aircraft is in the air

and

( control lever is up and all truck assemblies are retracted and locked

Or

Control lever is down and all truck assemblies are extended and locked)

**[PROP\_3]** Truck Assemblies are commanded to Extend **only** while ->

Aircraft in the air

And

Control lever is down

And

all doors are open

And

all truck assemblies are not down and locked

**[PROP\_4]**Truck Assemblies are commanded to Retract **only** while ->

Aircraft in the air

And

Control is UP

And

All Doors are Open

And

all truck assemblies are up and locked

With respect to the physical system behaviors, these may be constrained to use trajectories that relate to the fluent above. For example we may constrain the door opening time, or rate of change.

**[PROP\_T1]** Door opening shall complete within 6 seconds.

**[PROP\_T2]** Door closing shall complete within 6 seconds.

**[PROP\_T3]** While door closing the minimal rate of positional change shall be x % of full deflection per second.

**[PROP\_T4]** Gear extension of all tuck assembling shall complete within 7 seconds

**[PROP\_T1]**Gear retraction of all truck assemblies sha;; complete within 7 seconds

Within the event calculus specification (see later) such constraints can be mapped to fluent trajectory constraints. See later section

*Aside : One potential research goal would be to explore the mapping of such trajectory constraints within event calculus to the specification observers vie the synthesis of FMI-based observers or co-integration of the logical event calculus observers within FMI simulation campaign.*

***Exploring the differences between the two sets of requirements.***

In our experience, property-guided thinking simplifies the identification of composite triggering conditions and may therefore serve as a better foundation on which to establish the EARS based actionable requirements. However, as discussed above we believe that such actionable requirements may be necessary to avoid the inherent loss within properties with respect to the separation of guards and triggering events.

* Within the properties, the differentiation of guards and events are not of interest given that we are specifying the aggregated observational state at a single temporary qualified single point.
* Within the actionable requirements we are specifying the system reaction to events. In this context the differentiation between triggers and guards is an important aspect of behavioral specification.

Big question: With respect to DO-178C, and DO-331 a key question is whether such textual specification is necessary, or can such refinement and elaboration be performed directly within design models. Answering this question is not easy. A key consideration is whether the objectives of model coverage analysis can be achieved without such declaration of behavioral intent.

Next we refine the fusion of the two approaches as we refine the properties above into actionable requirements, by separating guard (in blue) from trigger conditions (in orange) we may reformulate following the set of actionable requirements. It is interesting that composite events become more dominant.

**SYS\_REQ\_100a.** While the aircraft is in the air, when either of following composite conditions become satisfied

* control lever is up and all truck assemblies are not retracted and locked
* control lever is down and all truck assemblies are not extended and locked

the Landing Gear System shall initiate the opening of all landing gear doors with X ms.

**SYS\_REQ\_101.** While the aircraft is in air and all doors are not confirmed closed, when any of the following composite conditions become satisfied

* control lever is up and all truck assemblies are retracted and locked
* control level is down and all truck assemblies are extended and locked

the Landing Gear System shall initiate the closing of all landing gear doors with X ms.

**SYS\_REQ\_102.** While the aircraft is in air, when *all* of the following conditions become satisfied

* control is lever is up
* all landing gear doors are open
* all truck assembles are not up and locked

the Landing Gear System shall initiate the retraction of all truck assemblies

**SYS\_REQ\_103.** While the aircraft is in air, when any of the following conditions become satisfied

* control lever is down
* all landing gear doors are not open
* All track assemblies are are up and locked

the Landing Gear System shall terminate the retraction of all truck assemblies

**SYS\_REQ\_104.** While the aircraft is in the air, when all of the following conditions become satisfied

* Control lever is up
* All landing gear doors are open
* All truck assemblies are not up and locked

the Landing Gear System shall initiate retraction of all truck assemblies

**SYS\_REQ\_105 .**While aircraft is in the air, when any of the following conditions become satisfied

* Control lever is down
* all landing gear doors are open
* All truck assemblies are up and locked

the Landing Gear System shall terminate retraction of all truck assemblies

BIG TODO

At the software level we initially introduce the input and output variables. See section “Experiments with Input and Output Variables and Layered Abstraction”. In the ideal, the terminology of these variables and derived layered attractions would make up the working vocabulary of the SW HLR requirements.

Unfortunately, due to the lack of tooling support and late changes due to the exploration of architectural refinement within the HLRs below we have not yet reharmonized the vocabulary within the requirement below.

Aside : A note on architectural orchestration.

Explaining that the correct requirements and casting of control priority within a composite event affords implicit definition of the control take over behavior.

## Software High-Level Requirements – very draft

**[SW\_REQ1]** While aircraft is in the air, when the following conditions become simultaneously satisfied

* Local lane has priority control
* The analog switch interlock has been confirmed within 1 second of the control lever changing position to a confirmed state

the Landing Gear Control Software shall command the stimulation of the general electro-valve within 10 ms.

**[SW\_REQ2]** While aircraft is the air, when the following conditions become simultaneously satisfied

* Local lane has priority control
* The general electro-valve has been stimulated for at least 200 ms
* Hydraulic pressure is confirmed to be at an appropriate level

The Landing Gear Control software shall command the stimulation of the door opening electro-valve with 10 ms.

**[SW\_REQ3]** While aircraft is in the air, when the following conditions become simultaneously satisfied

* Local lane has priority control
* the control lever is ‘Confirmed Down’
* Aggregated Door Status is ‘All Doors Confirmed Open’

the Landing Gear Control Software shall command the stimulation gear outgoing electro-valve within 10 ms.

**[SW\_REQ4]** While aircraft is in the air, when the following conditions become simultaneously satisfied

* Local control lane has priority control
* Control lever is ‘Confirmed Down’ and Aggregated Truck Status us ‘All Trucks are Down and Locked’
* The gear outgoing electro-valve has been stimulated for at least 100 ms.

the Landing Gear Control Software shall command the deactivation of the gear outgoing electro-valve

**[SW\_REQ3]** While aircraft is in the air, when the following conditions become simultaneously satisfied

* Local lane has priority control
* the control lever is ‘Confirmed Up’
* Aggregated Door Status is ‘All Doors Confirmed Open’

the Landing Gear Control Software shall command the stimulation gear retraction electro-valve within 10 ms.

within 10 ms.

**[SW\_REQ5]** While Aircraft is In Air and Landing Gear Control is Down, when the following conditions become simultaneously satisfied

* Local control lane has priority control
* Aggregated Truck Status us ‘All Trucks are Down and Locked’
* All landing gear doors are not confirmed closed and locked
* The gear outgoing electro-valve has been deactivated for at least 200 mns

the Landing gear control software shall command the stimulation of the door closure electro-valve within 10 ms.

**[SW\_REQ6]** While Aircraft is In Air and Landing Gear Control is Down, when the following conditions become simultaneously satisfied

* Local control lane has priority control
* All landing gears are confirmed to be down and locked
* All landing gear doors are confirmed to be closed and locked
* The door closure electro-valve has been stimulated for at least 100 ms.

the Landing Gear Control Software shall command the deactivation of the door closure electro-valve within 10 ms.

**[SW\_REQ7]** While Aircraft is In Air Landing Gear Control is Down when the following conditions become simultaneously satisfied

* Local control lane has priority control
* All landing gears are confirmed to be down and locked
* All landing gear doors are confirmed to be closed and locked
* The door closure electro-valve has been deactivated for at least 1 s

the Landing gear control software shall command the deactivation of the general electro-valve within 10 ms.

Work to do

* Integrate failure models into models and requirements
  + Demonstrate multi-perspective test generation
  + Demonstrate multi-perspective property preservation
* Project properties into open standards VISTAS[[10]](https://paperpile.com/c/ilxpn9/qV50) FMI[[11]](https://paperpile.com/c/ilxpn9/J1kd)
* Derive simulation and property preservation scenarios from operational scenario

# References

[1] [F. Boniol, V. Wiels, Y. Aït-Ameur, and K.-D. Schewe, “The landing gear case study: challenges and experiments,” *Int. J. Softw. Tools Technol. Trans.*, vol. 19, no. 2, pp. 133–140, Apr. 2017.](http://paperpile.com/b/ilxpn9/v4pe)

[2] [B. Hall, J. Fiedor, and Y. Jeppu, “Model integrated decomposition and assisted specification (MIDAS),” *INCOSE Int. Symp.*, vol. 30, no. 1, pp. 821–841, Jul. 2020.](http://paperpile.com/b/ilxpn9/QmrX)

[3] [D. L. Lempia and S. P. Miller, “Requirements engineering management handbook,” *National technical information service (ntis)*, vol. 1, 2009.](http://paperpile.com/b/ilxpn9/VtmB)

[4] [D. Bouskela and A. Jardin, “ETL: A new temporal language for the verification of cyber-physical systems,” in *2018 Annual IEEE International Systems Conference (SysCon)*, Apr. 2018, pp. 1–8.](http://paperpile.com/b/ilxpn9/LIK5)

[5] [D. Bouskela *et al.*, “Formal requirements modeling for cyber-physical systems engineering: an integrated solution based on FORM-L and Modelica,” *Requirements Engineering*, vol. 27, no. 1, pp. 1–30, Mar. 2022.](http://paperpile.com/b/ilxpn9/ojYA)

[6] [A. Mavin, P. Wilkinson, A. R. G. Harwood, and M. Novak, “Easy approach to requirements syntax (EARS),” in *Requirements Engineering Conference, 2009. RE ’09. 17th IEEE International*, Oct. 2009, pp. 317–322.](http://paperpile.com/b/ilxpn9/XcZW)

[7] [A. Mavin Mav and P. Wilkinson, “Ten Years of EARS,” *IEEE Softw.*, vol. 36, no. 5, pp. 10–14, Sep. 2019.](http://paperpile.com/b/ilxpn9/OKYw)

[8] [A. F. Paz and G. El Boussaidi, “Landing Gear Control Software: An avionics software development case study,” unknown, Jun. 2017. doi:](http://paperpile.com/b/ilxpn9/A50M) [10.13140/RG.2.2.34900.19848.](http://dx.doi.org/10.13140/RG.2.2.34900.19848.)

[9] [S. P. Miller, A. Tribble Rockwell Collins, and C. Rapids, “EXTENDING THE FOUR-VARIABLE MODEL TO BRIDGE THE SYSTEM-SOFTWARE GAP,” *), Daytona Beach, Florida, Oct.*, vol. 14, p. 18, 2001.](http://paperpile.com/b/ilxpn9/COOK)

[10] [“VISTAS – WG-97 Standard of Virtual Interoperable 3 Simulation for Tests of Avionics Systems in Virtual or 4 Hybrid Bench Networks.pdf.”](http://paperpile.com/b/ilxpn9/qV50)

[11] [T. Neidhold and T. Blochwitz, “FMI: Functional Mockup Interface for Model Exchange and Co-Simulation”.](http://paperpile.com/b/ilxpn9/J1kd)

## Experiments with Input and Output Variables and Layered Abstractions

The control software shall determine the status of each Landing Door, by continually evaluating the conditions in the following table at a minimal rate of 5 Hz.

| **Door Status** | **Conditions** |
| --- | --- |
| 'Door Confirmed Open' | While at least two of the three '**Door Open Signals**' are 'True' and at least two of the three '**Door Closed Signals**' are False |
| 'Door Confirmed Close' | While at least two of the three '**Door Closed Signals**' are 'True' and at least two of the three '**Door Open Signals**' are False |
| 'No Confirmed Door State' | Otherwise |

The control software shall determine the status of each Landing Gear Leg, by continually evaluating the conditions in the following table at a minimal rate of 5 Hz.

| **Landing Gear Leg Status** | **Conditions** |
| --- | --- |
| 'Confirmed Retracted' | While at least two of the three \*\*Gear Retracted Signals \*\* are 'True' and at least two of the three '**Gear Extended Signals**' are 'False'. |
| 'Confirmed Extended' | While at least two of the three \*\*Gear Extended Signals \*\* are 'True' and at least two of the three '**Gear Retracted Signals**' are 'False'. |
| 'No Confirmed Leg State' | Otherwise |

The control software shall determine the status of each Landing Gear Control Position by continually evaluating the conditions in the following table at a minimal rate of 5 Hz.

| **Landing Gear Control Position** | **Conditions** |
| --- | --- |
| 'Confirmed UP' | While at least two of the three \*\*Control Up \*\* are 'True' and at least two of the three '**Control Down**' are 'False'. |
| 'Confirmed DOWN' | While at least two of the three \*\*Control Down \*\* are 'True' and at least two of the three '**Control Up**' are 'False'. |
| 'No Confirmed State' | Otherwise |

The control software shall determine the **Aircraft On Ground Status** status ing table at a minimal rate of 5 Hz.

| **Aircraft On GrounD Status** | **Conditions** |
| --- | --- |
| 'Confirmed UP' | While at least two of the three \*\*Control UP \*\* are 'True' and at least two of the three '**Control**' are 'False'. |
| 'Confirmed DOWN' | While at least two of the three \*\*Control Down \*\* are 'True' and at least two of the three '**Control UP**' are 'False'. |
| 'No Confirmed State' | Otherwise |

## 

### Layered Abstractions

First we need to build the aggregate state of the system components, given that the requirements expressed over the aggregate system states,

The control software shall determine the **Aggregated Landing Gear Door Status** by continually evaluating the conditions in the following table

| **Aggregated Door Status** | **Conditions** |
| --- | --- |
| 'All Doors Confirmed Open' | While the **Door Status** of *each* is 'Door Confirmed Open' |
| 'All Doors Confirmed Close' | While the **Door Status** of *each* is 'Door Confirmed Close' |
| 'Aggregate Door State not Confirmed' | Otherwise |

The control software shall determine the **Aggregate Landing Gear Leg Status** by continually evaluating the conditions in the following table

| **Aggregated Landing Gear Leg Status** | **Conditions** |
| --- | --- |
| 'All Legs Confirmed Extended' | While the **Landing Gear Leg Status** of *each* Leg Assembly is 'Confirmed Extended' |
| 'All Legs Confirmed Retracted' | While the **Landing Gear Leg Status** of *each* Leg Assembly is 'Confirmed Retracted' |
| 'Aggregate Leg State not Confirmed' | Otherwise |

Define **Landing Gear Adjustment Status** by evaluating the conditions in the following table at

| **Landing Gear Adjustment Status** | **Conditions** |
| --- | --- |
| 'Extension Necessary' | \*\*Landing Gear Control Position \*\* is 'Up' and **Aggregate Landing Gear Leg Status** is not *' Confirmed Retraced'* |
| 'Retraction Necessary' | **Landing Gear Control Position** is 'Down' and **Aggregate Landing Gear Leg Status** is not *'Confirmed Extended'* |
| 'Adjustment Complete' | Otherwise |

Define **Landing Gear Adjustment** as 'Necessary' while the **Landing Gear Adjustment Status** is either 'Extension Necessary' or 'Retraction Necessary', otherwise not 'Necessary'.

| **Door Status** | **Conditions** |
| --- | --- |
| 'Door Confirmed Open' | While at least two of the three '**Door Open Signals**' are 'True' and at least two of the three '**Door Closed Signals**' are False |
| 'Door Confirmed Close' | While at least two of the three '**Door Closed Signals**' are 'True' and at least two of the three '**Door Open Signals**' are False |
| 'No Confirmed Door State' | Otherwise |

# 

# Event Calculus Representation of Requirements

### Older Work

### 

### Actuation variables

*Hydraulic Landing Gear Door Opening shall be commanded only while all the following conditions hold*

* *Aircraft is not on the ground*
* *Landing Gear Position Adjustment is Necessary*

*Hydraulic Landing Gear Door Closing shall be commanded only while all the following conditions hold*

* *Aircraft is not on the ground*
* *Landing Gear Position Adjustment is not Necessary*
* *Landing Gear Doors are not Confirmed Closed.*

*Hydraulic Landing Gear Extension shall be commanded only while all the following conditions hold*

* *Aircraft is not on the ground*
* *Landing Gear Doors are Confirmed to Open*
* *Landing Gear Control is Down*
* *Landing Gear Leg Status is not All Gears Confirmed Extended*

*Hydraulic Landing Gear Retraction shall be commanded only while all the following conditions hold*

* *Aircraft is not on the ground*
* *Landing Gear Doors are Confirmed to Open*
* *Landing Gear Control is Up*
* *Landing Gear Leg Status is not All Gears Confirmed Retra*cted

### System Reporting variables

The Status Panel Green Light shall be commanded ON when all the following conditions hold and commanded OFF otherwise.

Control Level is Down and All Landing Gear Truck Assembles are Extended

All Landing Gear Doors are Closed.

The Status Panel Orange Light shall be commanded ON when either of the following conditions hold, and commanded OFF otherwise

Landing Gear Control has been Down for less than 15 seconds and All Truck Assemblies are not Extended \*and All Landing Gear Doors are not Closed\*

Landing Gear Control has been Up for less than 15 seconds and All Truck Assemblies are not Retracted \*and All Landing Gear Doors are not Closed\*

Note should we add the \*conditions in italics\*

The Status Panel Red Light shall be commanded ON when any of the following conditions hold, and command OFF otherwise.

Landing Gear Control has been Down for greater than 15 seconds and All Truck Assemblies are not extended

Landing Gear Control has been Up for greater than 15 seconds and All Truck Assemblies are not Retracted

Landing Gear Control has been Up for greater than 6 seconds and All Landing Gear Doors are not Open

Landing Gear Control has been Down for greater than 6 seconds and All Landing Gear Doors are not Open

In this regard, the prior guidance within the FAA Requirement Engineering Management Handbook, with respect to the use and extension of Parnas’s Four Variable model [[9]](https://paperpile.com/c/ilxpn9/COOK) may blend well with property-driven specification workflows.

One related research question is how or indeed how such coupling between logical and physical behaviors is best expressed at the high levels of system specification.

OLD PICTURES

