Structural Scoped Review method

DATE: 28/10/2022

Summary

This document aims to make explicit and to explain the review method designed within the framework of the S2C project, more precisely in WP2 "Methods and means of implementing and maintaining system/safety consistency for the integrated system and system levels".

This method is based on the concept of "Consistency link" and benefits from existing work of MOISE project. The method is illustrated by a proof-of-concept (study case to apply the method and tooling to support the method) so as to experiment, improve and finally validate the method.

Author(s)	*Function(s) & name(s)*	*Research engineers*	*S. Guilmeau*
Checker(s)	*Function(s) & name(s)*	*Head of Systems Engineering Centre of Competence Saint Exupéry*	*J. Baclet*
Approver	*Function & name*	*Project leader IRT Saint Exupéry*	*J. Perrin*

Table of Contents

Evolutions

Version	Date	Modified §	Modification summary	Modified by
N/A	31/03/2020	All	Creation	All
	26/08/2020	3.2.2	Add (b) Sibling rule	MM
N/A	15/03/2021	From 3.3 to 5	Change or creation	All
1	01/12/2021	First page, Headers and Footers 1.2.2 2.1, 2.2, 4.1.1	Change title, add new member, reference and version as the previous is for TOP document (see NT-S085L02T00-026). Add reference to TOP document. Moved to TOP document except specific info	SG
2	14/04/2022	§2.1 §3 and §3.1	Add note induced by remark done during BCC V2 review about a possible more extended usage of the method. Add notes about limits, divergence and convergence about document against other documents of work package.	SG
3	28/10/2022	§1.1.2	Update (minor) TOP document reference	SG

Table of figures

Table of tables

1 Introduction

1.1 *Purpose of document*

This document aims to make explicit and to explain the review method designed within the framework of the S2C project, more precisely in WP2 “Methods and means of implementing and maintaining system/safety consistency for the integrated system and system levels”.

This method is based on the concept of “Consistency link” and benefits from existing work of MOISE project. The method is assessed by a proof-of-concept (study case to apply the method and tooling to support the method) so as to experiment, improve and finally validate the method.

Section 2 presents the problematic and gives our working assumptions. Sections 3 explains the proposed method, gives guidelines to apply it and toy examples to illustrate it. Section 4 presents the activities performed to validate the method by applying it on a study case of size representative of a small aeronautics system, the AIDA drone, with the support of dedicated demonstrator tool. Finally, Section 5 presents an overview of progress and future work.

1.2 *Referenced documents*

1.2.1 S2C reference documents

Title	Reference
State of the Art of the S2C Project	LIV-S085L01-001-V2, ISX-S2C-LIV-1001
Method to ensure and to maintain consistency of systemic levels & Validation report MBSE/MBSA consistency	LIV-S085L02-007-V6, ISX-S2C-LIV-1037-V6

Table 1: S2C reference documents

1.2.2 External reference documents

Title	Reference
MBSE/MBSA consistency. Activity report and synthesis (BIP from MOISE project)	LIV-S-014-S2.21-61-457-V1
MOISE method for collaborative Systems Engineering in extended enterprise (BIP from MOISE project)	LIV- S -014- S4.22-42-511-V1
Aerospace Recommended Practice - Guidelines For Development Of Civil Aircraft and Systems, Revision A, 2010	ARP4754A
Aerospace Recommended Practice - Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, 1996	ARP4761

Table 2: External reference documents

2 Context and objectives

On the one hand, Aeronautical authorities (EASA, FAA…) define the requirements that aircraft and embedded systems need to comply with in order to be allowed to fly. Safety teams support the related certification activities. They assess systems conformity with high level safety requirements and make sure the system fulfills its functions with the appropriate level of confidence.

On the other hand, system architects have to consider different points of view during systems development (e.g. security, performance, thermal, etc.). In order to integrate constraints and requirements, they need to rely on an efficient process. In particular, it is critical to ensure the conformity to safety requirements from the early design phases and all along the development.

Usually, during the development of critical systems, the system definition and the corresponding safety assessments are performed by different teams. This distribution is mainly due to the teams specialized skills, but it also allows complying with the independency required between system architecture design and validation means. Each team works with a representation of its own type of analysis. Furthermore, the formalism used by each domain varies regarding to the underneath language induced by the tools they used.

Safety analyses are validated by the system design team, complying with ARP4754A recommendations. This activity is intended to provide the necessary confidence in the system definition used to perform the architecture safety assessment. Mostly based on reviews, this activity involves both system and safety specialists and their analysis supports, which are the models. The system design team validates that the safety model is consistent with the system definition.

The diversity of formats complicates the communication between teams and the safety model validation by system engineers. For example, system description information is in textual or informal graphical form, while safety analyses are mainly supported by fault trees or Excel files. This makes it difficult to trace the reviews exchanges and outputs and to keep track of system modifications.

The understanding of the system by safety specialist is the core activity of the consistency. It is currently validated by a proof-reading process of safety analyses by the system design team. In case of critical systems (DAL A and B).It is important to note that the first consistency review is generally acceptable in term of cost and quality but the management of different reviews throughout the development iterations is much complicated. Indeed, this process is very time-consuming and operationally difficult as the whole review should be redone even if changes are very local.

Our proposition focus on the formalization of the SE and SA review including changes from a previous validated review. This enables to reduce review time and improve quality of the review. The earlier Inconsistencies are discovered, the greater the confidence in safety analysis and the better the development process is and the earlier anomalies can be detected. Therefore, costs of late discovering of inconsistency and redesign are avoided.

This proposition is described (as it was developed in the last months) in the following within the context of preliminary development phases and the use of MBSE and MBSA models.

2.1 Considered development phases and safety analyses

As this section is common with the Section 2.2 of LIV-S085L02-007-V6, ISX-S2C-LIV-1037-V6, text is communalized into this last one.

Note: Section 2.2 of LIV-S085L02-007-V6, ISX-S2C-LIV-1037-V6 limits designed methods to pSSA and SSA perimeter. But this method could be used to consistency for other assessment (contrarily to other of the work package), for example, by making link between function in FHA with functions in SE models to ensure covering. In such case of use, only little part of the method will be used (so there is less interest for these usage but is not null).

2.2 Considered models

As this section is common with the Section 2.4 of LIV-S085L02-007-V6, ISX-S2C-LIV-1037-V6, text is communalized into this last one.

Note: The method could also be applied, yet untested, to two copies of an overloaded model (refer to Section 4.2.1.1 in LIV-S085L01-001-V2, ISX-S2C-LIV-1001).

2.3 Needs for a consistency review method

The needs for a consistency review method are the following:

- Formalize the consistency between SE model and SA model.
- Focus on changes either from SE or SA to make review efficient.
- Enable parallel work for SE and SA as they do not use same model (latest unofficial version for SE in edition while last SE published version for SA works). In particular, a safety analysis takes some time to perform and safety specialist needs a stable version of the SE model all along its analysis.
- Allow freedom to model, i.e., add minimum possible constraints on modeling formulation. All modeling languages, in particular those that enable simulation and computation, as required in simulation have already specific constraints.
- Ensure SE non-disturbing activity by the other domain concerns. For example, the supplementary modelisation required by SA for dysfunctional analysis while not required by the SE model.
- Enable to preserve the independence between design and safety analysis for critical systems of DAL A and B, as required by ARP4761.
- Ensure capitalization of review to improve quality.

3 Developed method

This section focuses on the definition of our developed consistency review method. As a method dealing with models and changes conveys inherently a huge amount of artefacts, a “consistency management tool” is explicitly introduced in Section 3.1.3 to support our developed method. Nevertheless, we consider in the whole Section 3 the tool without implementation choice at this level.

Section 3.1 gives an overview of all steps of the method and 3.2 specializes it for the model elements related to functional architecture. Sections 3.3 to 3.5 detail the three main steps of the method.

The method elements defined in this section will be illustrated and validated by application on a use case in Section 4. More precisely, Sections 3.2 to 3.5 are illustrated in Sections 4.2 to 4.5 respectively.

Note: This document being written anteriorly to document LIV-S085L02-007-V6, ISX-S2C-LIV-1037-V6 and updated on purpose updates, its organization differ slightly from Section 2.3 of document LIV-S085L02-007-V6, ISX-S2C-LIV-1037-V6.and corrections to converge to rules formulated by LIV-S085L02-007-V6, ISX-S2C-LIV-1037-V6 will not be done.

Note: This document differs also because it does not trace against User’s needs Analysis (Section 3.1 of document LIV-S085L02-007-V6, ISX-S2C-LIV-1037-V6).

3.1 Method overview

This section gives an overview of the method and how it targets the specified needs. These elements are explained through sequence diagrams in a top-down approach, beginning with a high-level scenario that makes explicit the targeted consistency need and then introducing the method steps and the positioning of a consistency management tool.

Note: as this method is based upon structural perspective, any behavioral consideration introduced in LIV-S085L02-007-V6, ISX-S2C-LIV-1037-V6 is not applicable to this document.

3.1.1 Targeted consistency needs and scenario

As previously introduced, safety analysis is required very early in the development and should be done at each design step, ensuring that “unsafe” design choices are identified and mitigated as soon as possible. Consequently, the development iterations follow some generic steps:

- Creation of a first system architecture model (based on the requirements)
- Creation of a first safety model (based on the requirements and the system architecture)
- Review to validate that system and safety models are consistent with one another
- Exploitation of safety model resulting in safety analysis, and possibly in safety requirements or recommendations for system architecture
- When a new version of system model is published (to take into account recommendations from safety and other trades),
    - Safety model is updated

- Consistency review is redone

In Figure 1, these steps are detailed and allocated to different actors: system architect, safety specialist and reviewers (reviewers are the people that have built both models or other people with knowledge of the system and the models). The system architecture definition and modeling evolves all along the development. Some versions are published and made accessible to safety specialist (and other engineering fields too) as baselines of study. Nevertheless, the system design continues to change. When a new version called V2 is published as a new baseline version, where changes are linked to safety recommendations about version V1 and other sources, then the safety specialist has to carry out an analysis to update its safety model accordingly to changes done.

The consistency review aims to validate the MBSA model with respect to the relevant published version of MBSE model (not with the current MBSE model version). Consistency review requires to have access in reading mode to models. It highlights some inconsistencies which require modeling work to be solved. These modeling changes shall also be reviewed, making a loop in the scenario. When the MBSA model is fully consistent and reviewed, the MBSA specialist computes the cutsets[1] and performs the safety analysis so as to give safety recommendations or requirements.

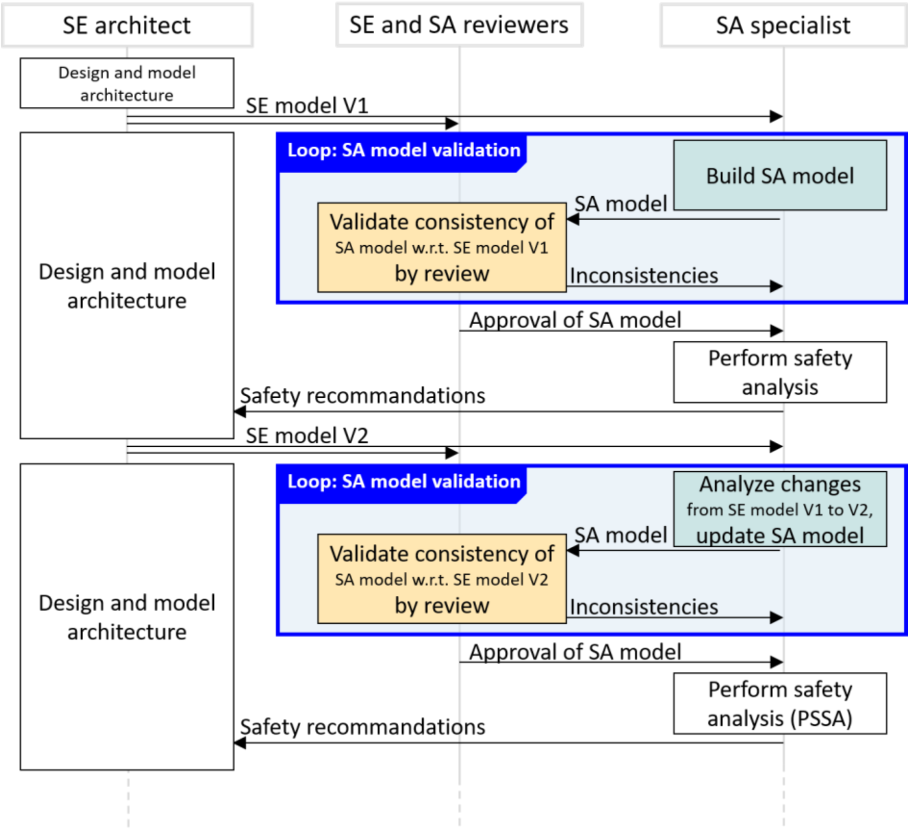

SE architect
SE and SA reviewers
SA specialist
Design and model architecture
SE model V1
Loop: SA model validation
Build SA model
SA model
Validate consistency of SA model w.r.t. SE model V1 by review
Inconsistencies
Design and model architecture
Approval of SA model
Perform safety analysis
Safety recommandations
SE model V2
Loop: SA model validation
Analyze changes from SE model V1 to V2, update SA model
SA model
Validate consistency of SA model w.r.t. SE model V2 by review
Inconsistencies
Design and model architecture
Approval of SA model
Perform safety analysis (PSSA)
Safety recommandations

Figure 1: Targeted scenario: initial and following iterations

In this scenario, each step has a duration. In particular, building a MBSA model requires time to read the MBSE model, gather other design information, in particular for behavior or physical dependencies, reconcile modeling objectives and modeling tooling capabilities. In the same way, the activities of MBSE architect have a non-negligible duration. Because of industrial development time constraints, performing these MBSE and MBSA activities sequentially, one after another is not industrially possible: these activities are rather performed simultaneously.

The need of parallel working by both teams implies to work with distinct baselined versions for example: SE with its last up to date one and safety with the previous stable one.

3.1.2 The consistency link as method basis

[1] A cutset is a minimal combination of basic failures resulting in a given failure condition.

To facilitate the steps of consistency review and safety model update at each iteration, we propose a method inspired by the model breakdown and the traceability approaches:

- To decompose the review in small reviews, following the model breakdown. Instead of reviewing the models as a whole, we review a small number of model elements, in particular those that have changed since the previous review.
- To define and navigate links between engineering artefacts existing independently from these artefacts and carrying traceability properties similar to traceability links for example.

As a result, we propose to define a specific link for consistency. It relates on one side MBSE model elements and on other side MBSA model elements. We introduce a new artefact type "Consistency Link" (CL), which means that:

"the MBSE model element(s) and the MBSA model element(s) linked together represent the same object".

A consistency link carries the consistency validation by both reviewers of a small part of the models (linked model elements) and thus complies with certification process. This validation is made concrete by the attributes of the CL:

- The rationale that captures the justification, the assumptions, the conditions of the consistency of linked model elements
- The validation status whose possible values are:
  - Suspect: the CL shall be validated by review. Either, the CL has never been validated or some changes have been done since the last validation of the CL.
  - In revision: the CL has been reviewed and considered not valid. Some work is required.
  - Validated: the CL has been reviewed and validated. Since then, no changes have been done in the linked model elements or in the CL itself.
- Validation date
- Validation authors

The lifecycle of the CL, regarding its validity status is summed up in Figure 2. During review, the state is manually modified, whereas changes in models or CL automatically change the status to suspect.

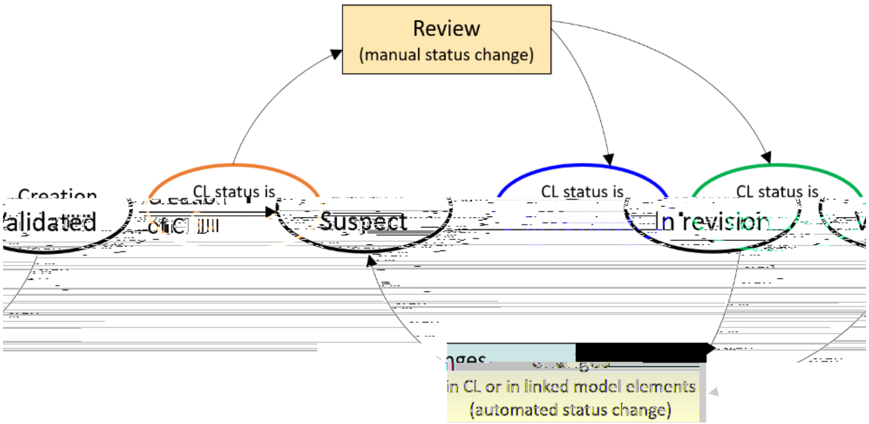


Figure 2: The status of a CP and its changes

The consistency link complies with the needs for both domains to have a non-constraining relation, as mentioned in Section 2.3. It is applicable to MBSE and MBSA models whether they are homogeneous or heterogeneous in term of language, modeling method, edition tool and storage format.

Complying with the need to fully validate and review the MBSA model (cf Section 3.1), we choose to require that the set of consistency links (CLset) shall cover the whole MBSA model. Also, this coverage shall be ensured without overlapping, i.e. each object of the model for which the definition of a CL is relevant shall be covered by one and only one CL. This full and non-overlapping coverage enables to define a formal correctness of the set of consistency links, avoiding any confusion in consistency link interpretation. If there is an overlap (i.e., a SE function is linked to 2 CLF), we face confusion: in which SA model elements this function is represented ?

The scenario illustrated in Figure 1 is enriched with a tooling support necessary for consistency management that shall enable creation, edition and storage of consistency links, resulting in Figure 3. Consistency links are edited (creation or update) during the MBSA modeling step.

Consistency links are exploited:

- To support the change analysis between MBSE models. An automatic change of status enables to highlight the MBSE model changes
- To focus the consistency review on MBSE and MBSA model elements that have changed
- To capitalize on the hypothesis, justifications, discussions about the MBSA modelling choices
- To give the validation status of the consistency between MBSE and MBSA models, through the consistency links status

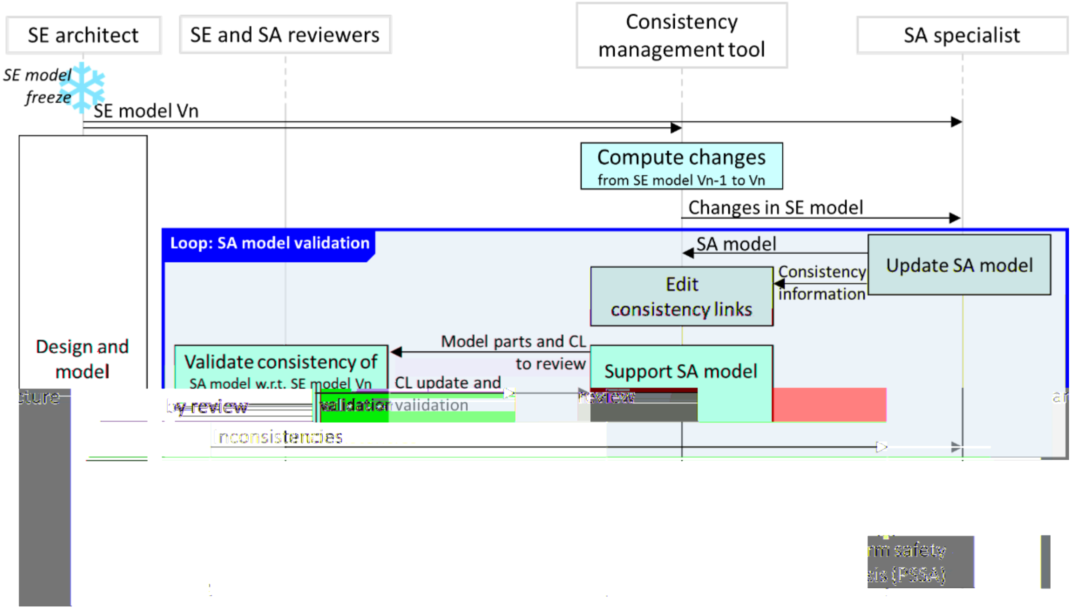


Figure 3: Consistency scenario with CLs (only an nth iteration)

3.1.3 Refining the consistency scenario

In this section, the steps presented in Figure 3 are detailed in Figure 4. A CL defines a local consistency, limited to its linked model element. To fulfill the objective of consistency statement, we propose to carry out some checks to:

- Ensure that both the whole models are covered,
- Avoid that CLs add confusion to model reading, e.g., avoid overlap of CL
- Ensure that, as far as automatic checks are able to detect, there is no mismatch in CL definition.

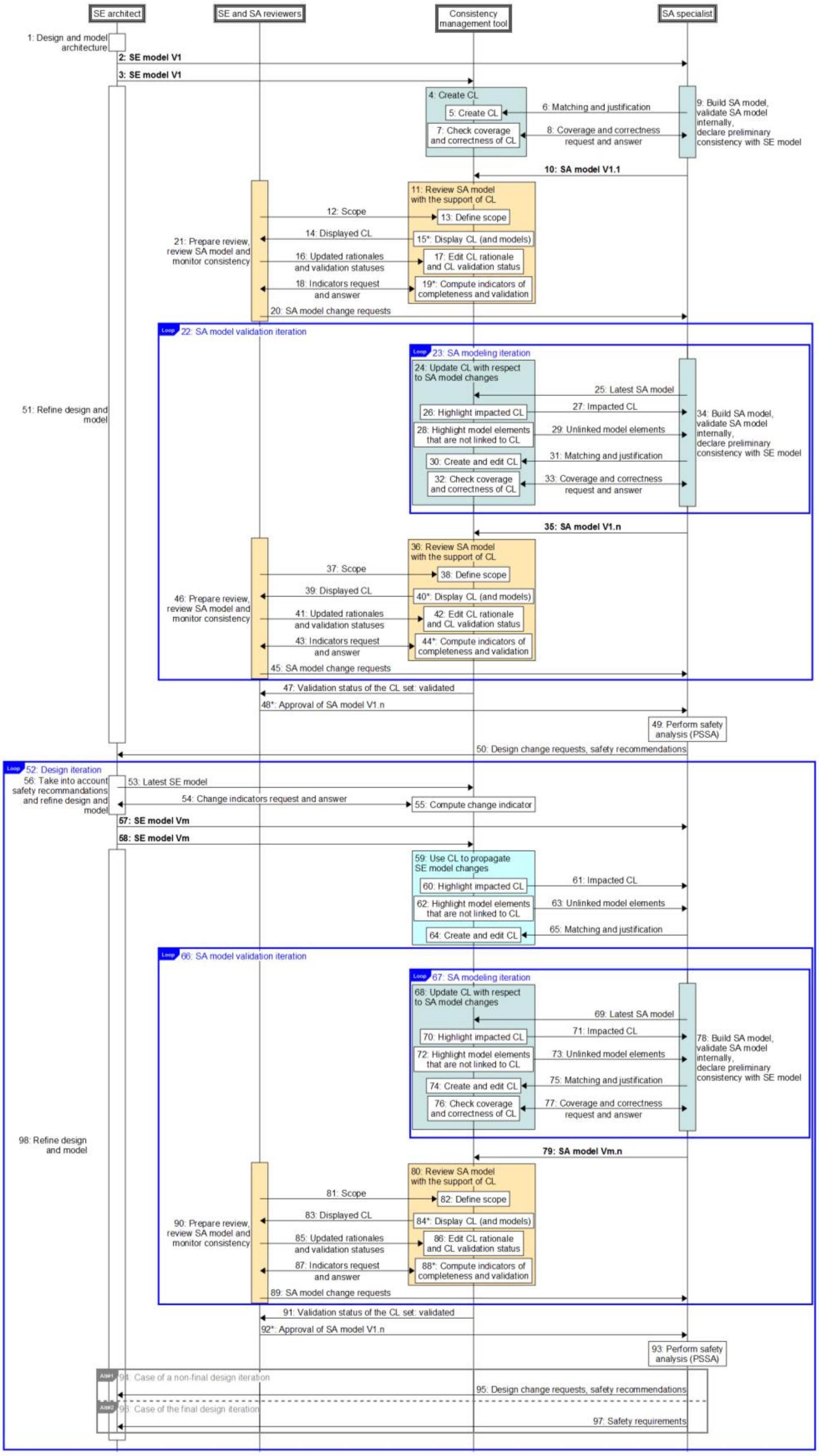
These checks are required before the review, to ensure the formal correctness of CL that will be reviewed. Checks are also used during review to compute indicators, notably indicators of review and validation progress. Before a review, reviewers can limit it by defining a scope, i.e. only subparts of models will be reviewed.

As the consistency links aim to focus review on the changes, the system architect needs an indication of how far his current MBSE model is from the latest published version. When the distance becomes too high, it is recommended to publish a new version, failing that, the whole safety model shall be reviewed (a distance metric is still to be defined by the method).

Figure 4 requires some legend elements and comments on starred elements that are given below:

- The MBSA model version "Vm.n" corresponds to:
  - the mth MBSE model version and
  - the nth MBSA model version of the system as described in the nth MBSE model.
- Color keys:
  - in light blue, MBSA specialist is informed of changes in MBSE model in order to ease MBSA model update (no modeling activities at this step),
  - in blue, building of MBSA model and related consistency activities,

    - in yellow, review of MBSA model and related consistency activities,
    - in bold, published versions of models.
- *14, 39, 83: in order to display CL satisfactorily, some parts of models are displayed by the consistency management tool.
- *18, 43, 87: Indicators are KPI computed by the tool and give an overview of the consistency coverage and the review effort.
- *47, 91: The validated status of all CLs is a necessary condition for the reviewers to approve the model as a whole.


SE architect
SE and SA reviewers
Consistency management tool
SA specialist
1: Design and model architecture
2: SE model V1
3: SE model V1
4: Create CL
5: Create CL
6: Matching and justification
7: Check coverage and correctness of CL
8: Coverage and correctness request and answer
9: Build SA model, validate SA model internally, declare preliminary consistency with SE model
10: SA model V1.1
11: Review SA model with the support of CL
12: Scope
13: Define scope
14: Displayed CL
15*: Display CL (and models)
16: Updated rationales and validation statuses
17: Edit CL rationale and CL validation status
18: Indicators request and answer
19*: Compute indicators of completeness and validation
20: SA model change requests
21: Prepare review, review SA model and monitor consistency
Loop 22: SA model validation iteration
Loop 23: SA modeling iteration
24: Update CL with respect to SA model changes
25: Latest SA model
26: Highlight impacted CL
27: Impacted CL
28: Highlight model elements that are not linked to CL
29: Unlinked model elements
30: Create and edit CL
31: Matching and justification
32: Check coverage and correctness of CL
33: Coverage and correctness request and answer
34: Build SA model, validate SA model internally, declare preliminary consistency with SE model
35: SA model V1.n
36: Review SA model with the support of CL
37: Scope
38: Define scope
39: Displayed CL
40*: Display CL (and models)
41: Updated rationales and validation statuses
42: Edit CL rationale and CL validation status
43: Indicators request and answer
44*: Compute indicators of completeness and validation
45: SA model change requests
46: Prepare review, review SA model and monitor consistency
47: Validation status of the CL set: validated
48*: Approval of SA model V1.n
49: Perform safety analysis (PSSA)
50: Design change requests, safety recommendations
51: Refine design and model
Loop 52: Design iteration
53: Latest SE model
54: Change indicators request and answer
55: Compute change indicator
56: Take into account safety recommandations and refine design and model
57: SE model Vm
58: SE model Vm
59: Use CL to propagate SE model changes
60: Highlight impacted CL
61: Impacted CL
62: Highlight model elements that are not linked to CL
63: Unlinked model elements
64: Create and edit CL
65: Matching and justification
Loop 66: SA model validation iteration
Loop 67: SA modeling iteration
68: Update CL with respect to SA model changes
69: Latest SA model
70: Highlight impacted CL
71: Impacted CL
72: Highlight model elements that are not linked to CL
73: Unlinked model elements
74: Create and edit CL
75: Matching and justification
76: Check coverage and correctness of CL
77: Coverage and correctness request and answer
78: Build SA model, validate SA model internally, declare preliminary consistency with SE model
79: SA model Vm.n
80: Review SA model with the support of CL
81: Scope
82: Define scope
83: Displayed CL
84*: Display CL (and models)
85: Updated rationales and validation statuses
86: Edit CL rationale and CL validation status
87: Indicators request and answer
88*: Compute indicators of completeness and validation
89: SA model change requests
90: Prepare review, review SA model and monitor consistency
91: Validation status of the CL set: validated
92*: Approval of SA model V1.n
93: Perform safety analysis (PSSA)
Alt#1 94: Case of a non-final design iteration
95: Design change requests, safety recommendations
Alt#2 96: Case of the final design iteration
97: Safety requirements
98: Refine design and model

•

Figure 4: Consistency scenario (initial and following iterations)

The method covers the consistency of the whole model by decomposing it in small pieces and capitalizes the review in consistency links throughout development iterations. To be operational, the method has to be specialized for the different types of model elements that are shared by MBSE architect and MBSA specialist. The previous scenarios describe the *steps* of the method. In the following sections, we focus on the *objects* whose consistency shall be ensured.

Note of project organization: we have chosen to begin with the functional architecture.

After a detailed presentation of the way to define CL and the associated checks in Section 3.2, we present the method following the main steps shown in Figure 4:

- Section 3.3 details the usage of CL for the propagation of SE model changes (the light blue step, numbered 59)
- Section 3.4 details the update of CL with respect SA model changes (the blue step, numbered 68)
- Section 3.5 details the SA model review (the yellow step, numbered 80)

3.2 Define consistency links for functional architecture

We work on a MBSA functional model, modeling only the functional architecture, without allocation to logical and physical architecture. This type of modeling is not a common practice, focused on logical/physical architecture, as physical failures are the source of the quantification of failure conditions, a major safety result. Nevertheless, a “functional MBSA model” enables to perform a “functional PSSA” analysis on the detailed functional architecture including functional flows and thus to provide early safety recommendations, for instance, about function segregation or functional behavior.

Note: All the concepts developed for the consistency of functional architecture could be reused and adapted to ensure the consistency of physical architecture (See Section 5.3).

Firstly, functional elements relevant to be shared and managed in consistency between MBSE and MBSA are presented in Section 3.2.1. Then the consistency link is specialized for these elements. For each type of consistency link, we present associated coverage and correctness rules, examples and guidelines. Finally the consistency review is explained by Section 3.5.

3.2.1 Functional architecture abstracted as functions and flows

A functional architecture model is composed of several types of model elements. Figure 5 presents a small example model, representative of usual modeling methods, in MBSE and MBSA modeling. In this example, there are:

- 6 functions (F)
- 3 function breakdown relations
- 11 ports (Fx.Py)
- 11 relations of belongings of ports to functions
- 7 segments (L)

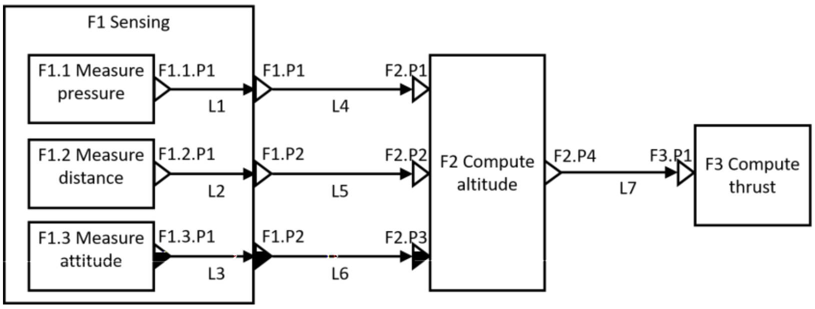


Figure 5: Model example

To ease consistency, we choose to abstract the model of interest, replacing ports and links by flows. The model of Figure 5 is abstracted in Figure 6. In this abstraction, there are less model elements:

- 6 functions (F)
- 3 function breakdown relations
- 4 flows (Flow)

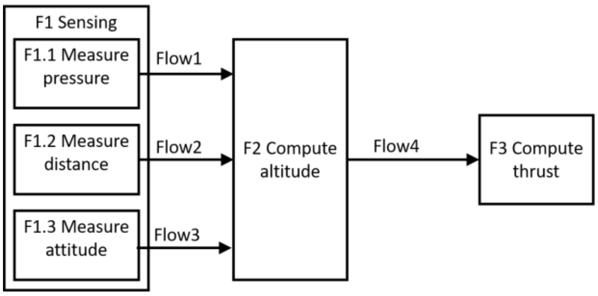


Figure 6: Translated example model

Appendix B gives an example of abstraction on a small model for several MBSE and MBSA modeling methods and tools.

We will see in the following that the user has flexibility to choose the most relevant functional level in consistency management.

3.2.2 Consistency link for functions

A Consistency Link for Functions (CLF) links a set of functions from MBSE model to a set of functions in MBSA model.

(a) Coverage rule

We define a coverage rule to be able to check whether there is enough or too much CLF. This rule enables to:

- check completeness of the CLF coverage,
- define a stopping criteria to the CLF definition activity ,
- avoid overlap between CLFs.

Each leaf function (i.e. lowest function in functional breakdown structure) of MBSE model shall 1) be linked by one CLF, or 2) have one hierarchical function (at any level of breakdown) that is linked by one CLF.

Idem for leaf functions of MBSA model, they shall comply with 1) or 2).

To manage the granularity difference between MBSE and MBSA models, the consistency method shall be flexible on the functional breakdown level taken into account. For instance, a safety model represents with more details than the system model the safety relevant elements of the system. On the contrary, the system model will deeply detail a performance-related payload even if it has no safety impact.

Figure 7 shows an example of correct application of the rule. It aims to illustrate the possibilities of flexibility offered by the coverage rule (and is not representative of a recommended use, see guidelines):

- The high-level function “Sense” has no behavior independently from its sub functions. As its sub functions are linked, the “Sense” function does not require its own consistency link. The review will be performed on sub functions.
- Sub functions of “Measure altitude” are important to compute a precise altitude (performance) but not for safety. As a consequence, they are not modeled in safety and they are directly linked to a CLF. They are covered indirectly by CLF1.
- It is possible that some elements have different names and yet to be consistent, for example, the functions linked by CLF2.
- “Control position” in MBSA model has been modeled by using predefined nodes. For this reason, it is split into two nodes. This modeling trick does not affect the consistency management, as no additional CLF is required for both predefined nodes.
- Some functions are not safety relevant, such as those linked by CLF4.
- In both models, some modeling artefacts may added to the model. In particular, to computable safety models. For example, the failure condition is a modeling artefact that is irrelevant in the MBSE model (CLF6).
- The function “Trigger end of mission” does not appear in the same level of breakdown: it is a high-level function in MBSA model , while it is a sub function in MBSE model (CLF5)

As illustrated in Figure 7 by “Sense” function and subfunctions of “Measure altitude”, the previous rule may be interpreted as “covering only one breakdown level”. The choice of considering coverage only at one breakdown level reduces the number of CLF to be defined, maintained and reviewed.

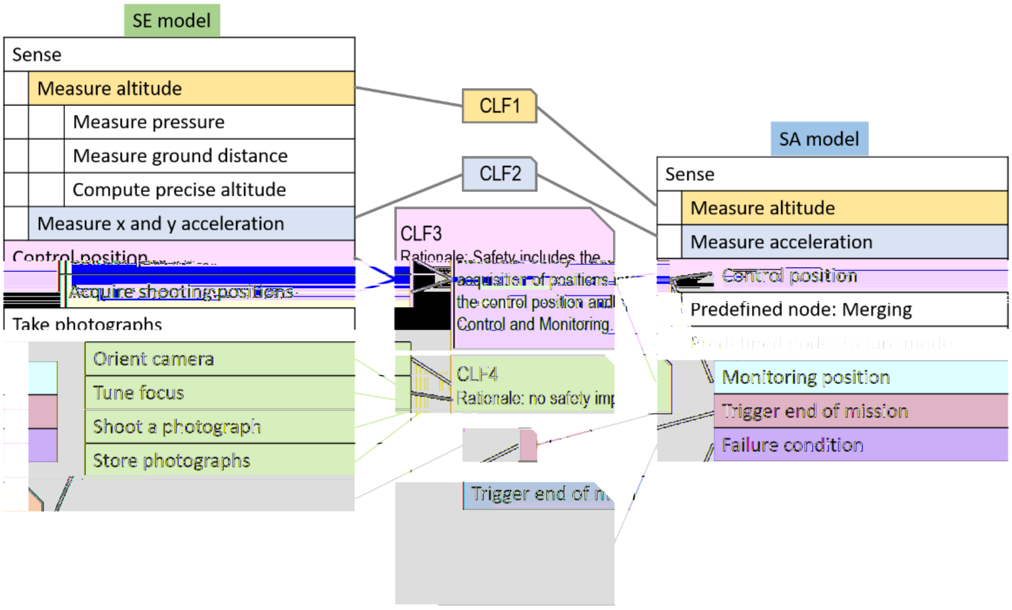


Figure 7: Example of function consistency links shown in functional breakdown of MBSE and MBSA models

Figure 8 and Figure 9 show examples of models and associated CLF that do not comply with the coverage rule. As this rule (and all others following) are symmetric between MBSE and MBSA model, the examples are valid considering MBSE model at left-hand and MBSA model at right-hand or reversely (noted Sx and Sy models in the figures). Figure 8 highlights that a function shall be linked by only one CLF. This avoids double review of the function with possibly different change requests. Furthermore, it limits the number of CLFs. Figure 9 shows that inside a breakdown, only one level shall be linked to a CLF. Similarly, it avoids double and contradictory reviews.

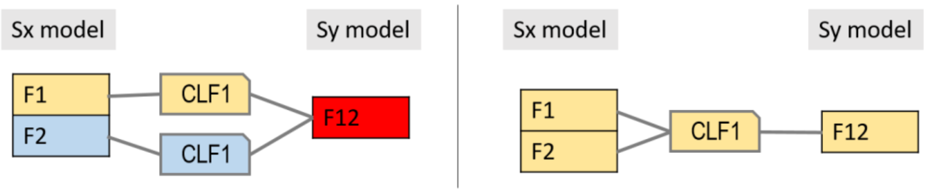


Figure 8: Non-overlap of CLF for each function: Counter-example (left-hand) and possible solution (right-hand) to a rule coverage non-compliance

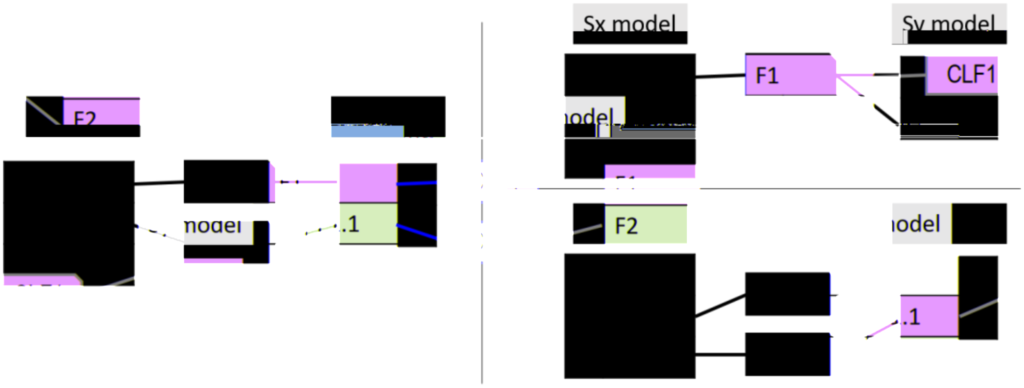


Figure 9: Non-overlap of CLF: Counter-example (left-hand) and two possible solutions (right-hand) to a rule coverage non-compliance

(b) Sibling rule

We add a rule on the possible gathering of functions linked to a CLF in order to ease display and readability of CLF.

If several functions of MBSE model are linked to a CLF, these functions shall share the same immediate parent, in other words, they shall be siblings.

Idem for functions of MBSA model.

Figure 10 shows two cases of violation of sibling rule.

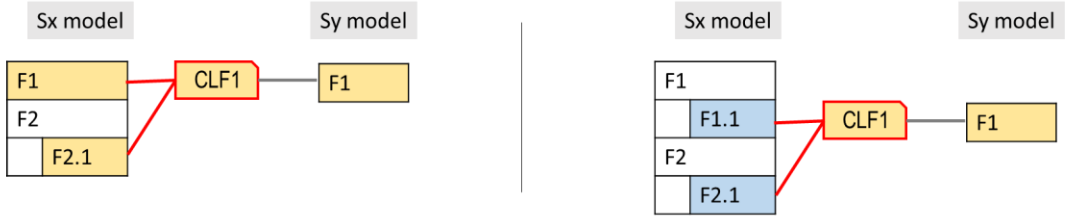


Figure 10: Counter-examples of sibling constraint: gathering functions with different levels of hierarchy (left-hand) and gathering functions with different parent (right-hand)

(c) Guidelines

When building the MBSA model and consistency links for functions, we recommend to keep the same structure as far as possible, i.e., to declare CLF from 1 MBSE function to 1 MBSA function as frequently as possible to foster CFL with cardinality “1-1”. Extension of that case means that cardinality “x-y” is by definition, the amount of “x” SE artefacts linked to “y” SA artefacts. Nevertheless, in some cases, because of modeling constraints or safety concerns, some structural differences are useful or necessary. In case of structural difference between models, it is recommended to justify it by a rationale. Thus, any CLF linking 0 or n>1 MBSE functions shall have a non-void rationale. And symmetrically, any CLF linking 0 or n>1 MBSA functions shall have a non-void rationale.

From this point of view, Figure 7 is NOT representative of a good method application. Its only purpose is to illustrate the possible flexibility in models structure.

As presented in Figure 9, several CLF sets can be applied to the same couple of MBSE and MBSA models. In particular, the MBSA specialist has to choose the good level of breakdown:

- If the chosen level is higher (top right in Figure 9), the number of CLF is reduced and each CLF has a wider scope resulting in general in a difficult behavior review. The consistency management can be seen as coarse-grain, with less confidence gain in models consistency.
- If the chosen level is lower (bottom right of Figure 9), the number of CLF is increased but each CLF has a smaller scope, a priori easing the behavior review. The consistency management can be seen as fine-grain with more confidence gain in models consistency.

Generally speaking, the good level is close to the leaf level and has to be chosen to ease the behavior review.

In term of number of functions, it is recommended to link to one CLF as few functions as possible. On the safety side, a CLF should be linked with as few safety model elements containing failure events as possible. Moreover, the behavior contained by the set of safety model elements must be kept reasonably complex to make review possible and efficient.

In Section 3.2.3, the consistency links for flows will be presented. An important guideline is to take into account the flows that exist in both models and the rules linked to consistency links for flows at CLF creation. In other words, the CLF creation shall be done with a support of functional flow diagram and not only based on functional breakdown.

(d) Added value

The CLF enables to manage consistency without constraining the modeling. The breakdown level that is suitable for consistency management is chosen without constraint. Coverage rule enables to ensure completeness of consistency management. The added value of the CLF is to structure the behavior review, splitting it in suitable sets of functions and ensuring its completeness.

3.2.3 Consistency link for functional flows

A Consistency Link for functional Flows (CLfl) links a set of flows from MBSE model to a set of flows in MBSA model.

(a) Coverage rule

The coverage rule for CLfl has the same objectives as the one for CLF (see Section 3.2.2(a)).

Each flow whose source and destination functions are linked to different CLF shall be linked by a CLfl.

Reformulation: each flow that is external to a CLF shall be linked by a CLfl.

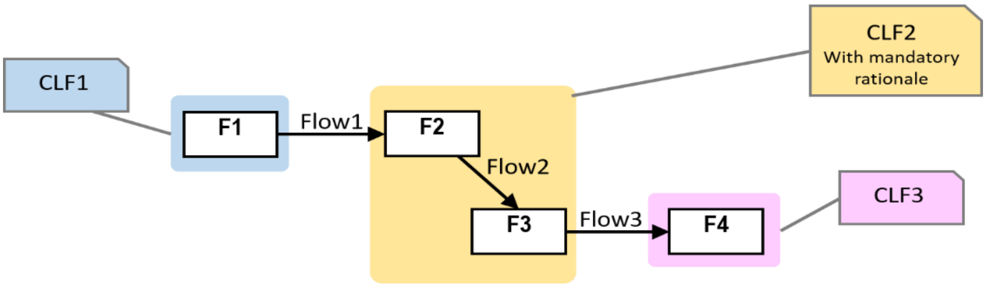


Figure 11: Example for coverage rule of consistency rules for flows

Applying this rule on the example of Figure 11, Flow1 and Flow3 shall be covered by CLfls, whereas Flow2 shall not be linked by any CLfl, given it is internal to the CLF2.

(b) Intra-domain correctness rule

In the MBSE model, two flows that are linked by a same CLfl shall have source functions that are linked to a same CLF. Symmetrically, they shall have destination functions that are linked to a same CLF.

Idem in the MBSA model.

Consequence: a CLfl has one source CLF from MBSE model, one source CLF from MBSA model, one destination CLF from MBSE model and one destination CLF from MBSA model.

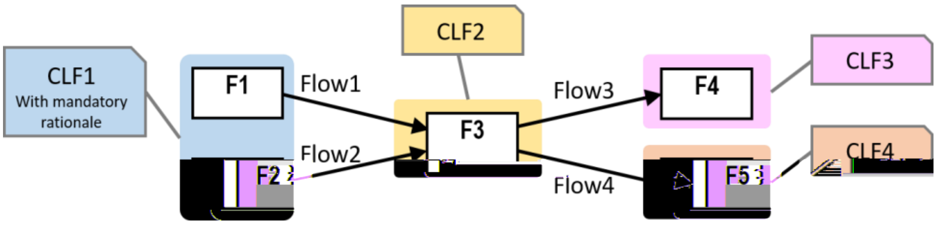


Figure 12: Example for intra-domain correctness rule

Applying this rule on the example of Figure 12, Flow1 and Flows2 can be linked by the same CLfl. Flow3 shall have its own CLfl, idem for Flow4.

(c) Inter-domain correctness rule

The previous rule enables to define an additional rule applying between domains that avoids mismatch in source and destinations of flows.

Given a CLfl, the source CLF from MBSE model shall be the same as the source CLF from MBSA model. Symmetrically for destination CLF.

Consequence: a CLfl has one source CLF and one destination CLF.

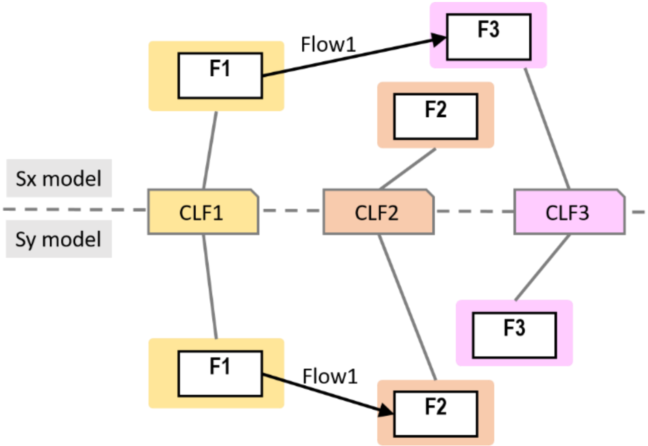


Figure 13: Counter-example for inter-domain correctness rule

On the example of Figure 13, a CLfl linking the two Flow1 would violate this rule, as the destination of the flow is linked to CLF2 in Sy model and CLF3 in Sx model. Indeed, the rule does not allow to have different destination, in terms of CLF.

This rules does not apply to any CLfl that link one or several flows in Sx model to none flow in Sy (illustrated in Figure 14). In this case, no check is applied.

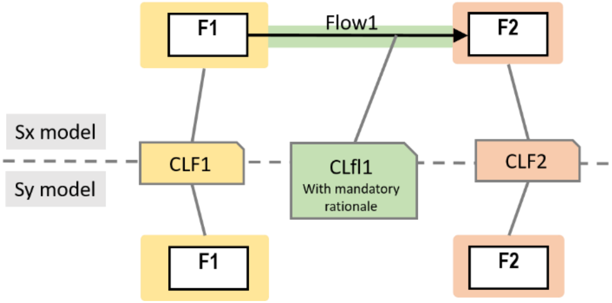


Figure 14: Case of non-application of the intra-domain correctness rule

(d) Guidelines

When building the MBSA model and consistency links for functional flows, we recommend to keep the same structure as far as possible, i.e., to declare CLfl from 1 MBSE flow to 1 MBSA flow as frequently as possible. Similarly to CLF, we foster CLfl with cardinality "1-1". Nevertheless, in some cases, because of modeling constraints or safety concerns, some structural differences are useful or necessary. In these cases, it is recommended to justify it by a rationale. Thus, any CLfl linking 0 or n>1 MBSE flows shall have a non-void rationale. And symmetrically, any CLfl linking 0 or n>1 MBSA flows shall have a non-void rationale.

The intention of defining consistency links for flows is to ensure completeness of flow consistency while limiting the number of CLfl by omitting the internal flows. The different granularity of flow description can be managed by gathering flows into one CLfl, as shown by CLfl1 in the example of Figure 15.

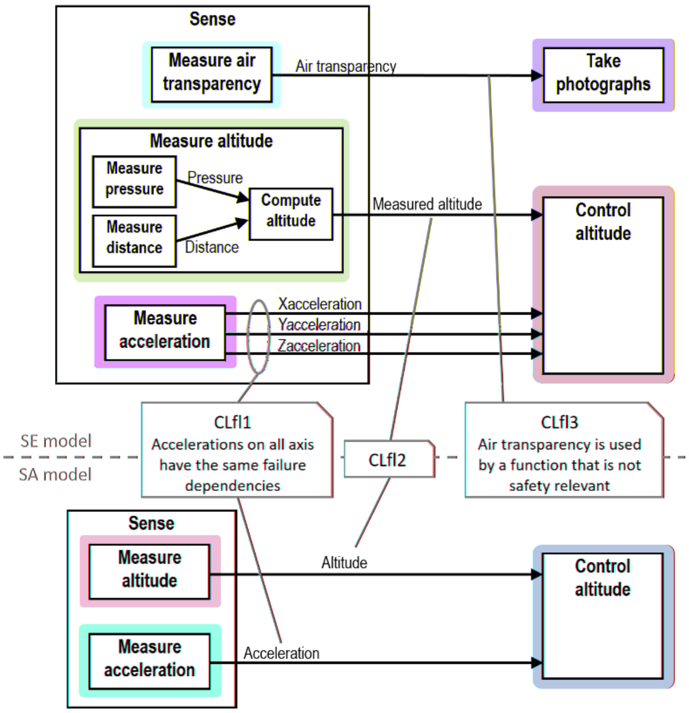


Figure 15: Example of consistency links for flows. Colors are used to indicate CLF.

Even if many modeling languages and tools offer to define a breakdown hierarchy between functional exchanges, it is defined on the segments (cf Section 3.2.1) rather on the flows. The breakdown of flows is applicable in only few cases, when source and destination are common to different flows, whereas the breakdown of links is always applicable between high-level functions. Moreover in a context of review, structured by consistency links, only the scope defined by the consistency link is examined at each step of review. That's why we omit to consider the flow breakdown in the consistency link definition. The user can still use it in modeling (if it used a tool that supports it) but the flow breakdown is not reviewed nor managed in consistency.

Through the intra- and inter-domain correctness rules, the definition of CLfl is dependent from the definition of CLF. In practice, both kinds of CL shall be built at the same time on the same sub part of the model: building all CLF before defining CLfl is NOT a recommended practice. When applying the method, the user will note that the set of functions linked to a CLF may change according to the flows.

(e) Added value

The CLfl enables to manage consistency of flows. It adds some minimal constraints on flow modeling so as to gain confidence. A mistake in the source or destination of a flow can have significant impacts in safety analysis and are difficult to detect in common safety modeling tools when several breakdown levels are crossed.

The CL of flows rules create dependencies to the CL of functions: this has to be considered between building both kinds of CL.

Coverage rule guarantees completeness of consistency management.

3.3 Use CL to propagate SE model changes

This part addresses how CL could help the safety specialist to update his model with respect to SE model changes. Even if CL are by construction good means to guide this kind of update, this part of the method is not addressed in detail. However, as system evolutions and iterations are steps of a system design, each SE model change must be evaluated and the SA model must be updated accordingly when necessary.

There are several ways to identify SE model changes and propagate these changes to the SA model and CLset:

- The safety specialist compares the different SE model versions to identify changes, and “manually” identify the impacts on the SA model and CLset.
- When a “Change Process” is used to manage design models' evolutions (which is the common and mandatory practice especially in late design phases), the safety specialist uses the “Change Reports” (CR) to understand the system changes and update accordingly the SA model and CLset.
- The safety specialist uses the previously reviewed CLset to identify impacted areas of the safety model where the consistency is not ensured anymore.

CL are by construction good means to guide this kind of update. However, this part of the method has not been addressed in detail and deserves further studies (see §5).

3.4 Update CL with respect to SA model changes

Figure 16 focuses on the way the consistency links are edited and impacted when the SA model changes, involving the safety specialist and the consistency management tool.

When a new version of SA model is published, the first task of the safety specialist is to make the CLset covering and correct. For example, if a new flow has been added in the model, the safety specialist has to link it to an existing CLfl or to create a new dedicated CLfl. On the contrary, if the source and destination functions of the flow are linked to the same CLF, there is no “CLset maintenance task” to do.

To be efficient, this model publication and “CLset maintenance” loop should be done regularly during the SA modeling task without waiting for a complete and stable SA model version. Thus, CL are easier to define and rationale easier to write as they capture the modeling intent of the instant.

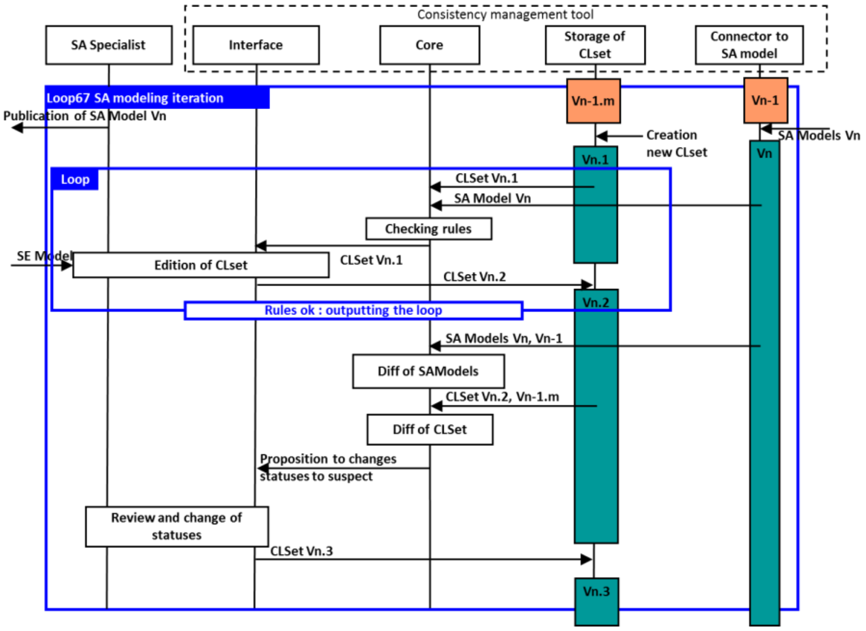


Figure 16: Detail of CL edition when SA model changes

When the SA model is stable and all defined rules pass without errors, the safety specialist prepares the review. Based on the changes in SA model, SE model and CLset, the tool proposes to change the status of some CL to suspect (see Section 3.4.1). Based on these propositions, the safety specialist defines the actual set of CL to review (see Section 3.4.2).

3.4.1 Status changes proposed by the consistency management tool

Proposition of suspect statuses is a trade-off between:

- Make any change in the model suspect, with the result of many CL to review and then long and costly reviews
- Make too few changes suspect, with the result of significant changes in the models that stay unreviewed and then potentially erroneous.

In accordance with the safety approach, we apply a conservative approach, privileging the suspect status in case of doubts of review relevancy. Nevertheless, to reduce the number of suspect CL, we focus on model elements that contribute to behaviors, i.e., leaf functions and flows. Hierarchical functions and naming are considered less important. The approach chosen considers in the same way changes from SE model and changes from SA model.

The model changes resulting in a proposition of suspect status are the following:

- Add a leaf function
- Add a flow
- Delete a leaf function, delete a function previously linked to a CLF (except for functions that are linked to a CLF with cardinality 1-0)
- Delete a flow (except for flows linked to CLfl with cardinality 1-0)
- Rename a function linked to a CLF
- Rename a flow linked to a CLfl

The deletion of a SA model element that has no equivalent in the SE model (or respectively a SE model element with no equivalent in SA model) will not result in a proposition of suspect status, as there is no more model element to review. For example, in Figure 7, the deletion of "Failure conditions" would not result in a proposition of suspect status.

Let note that, the hierarchical functions that are above the level of definition of the CL are not taken into account, in the objective to reduce the number of CL to review and with the justification that they contribute neither to review structuration, nor to model behaviors. For example, in Figure 7, the creation of a function "Managing position" as parent function of "Control position" and "Acquire shooting positions" would not result in any proposition of suspect status and then, in any review.

In the same way, any renaming above or under the level of definition of the CL are not taken into account. For example, in Figure 7, a renaming of the function "Measure pressure" would not result in any proposition of suspect status.

In addition to changes in models, changes in the definition of CLset itself result in proposition of suspect statuses, as some review scopes are modified and may impact the review result. More precisely, the CL changes resulting in a proposition of suspect status are the following:

- Link to a model element (add a function to CLF or add a flow to a CLfl)
- Unlink from a model element
- Change rationale

3.4.2 Status confirmed by the safety modeler

The safety specialist reviews these propositions. He can both:

- Change status of some proposed valid CL to suspect, in case of unmodified modeling elements he wants nevertheless to be reviewed by the architect
- Change status of some proposed suspect CL to valid (auto-validation)

The level of status filtering and CL auto-validation by the safety specialist depends on the organization practices between the system architect and the safety specialist. They can review together all the suspect CLs, which would result in a complete but extensive review and thus necessitate an important availability of the system architect. Alternatively, they can agree beforehand on the level of auto-validation performed by the safety specialist prior to the review and the subset of CLs to be reviewed, e.g. those corresponding to the perimeter of the system evolution defined by a set of Change Reports.

3.5 Review SA model with the support of CL

As presented in Section 3.2, we propose a method to improve the consistency on functional architecture. Firstly, we define the minimal model elements composing functional architecture: functions and flows. To reconcile confidence in models consistency while keeping modeling flexibility for system architect and safety specialist, the previous sections define 5 constraining rules to be applied on the CL building. These rules have to be known by the responsible of the CL edition (which is the safety specialist in our case) and are accompanied by examples, counter-examples and user guidelines.

Once the safety specialist has defined an MBSA model and consistency link, the consistency review involving reviewers from both safety and system domains can take place (in Figure 4, step 80 and its copies: steps 11 and 36).

During this review, the objective is to validate that the safety model is representative of the system design, described by the system model (as a reminder, system model is considered as the reference and assumed consistent with respect to any element of system description external to the model, e.g., textual requirements).Performing efficiently this review requires the following pre-requisites:

- Having an MBSA model and consistency links up to date and correct regarding the rules defined in section 3.2. Performing a review with remaining inconsistencies in the CLset may be possible (e.g. if the review focuses on a part of the model not involved in those inconsistencies), but not recommended as there is a risk of invalidation due to further CLset update.
- Defining a review scope, i.e. a set of consistency links to be reviewed (items 81 and 82 in Figure 4). These consistency links status should be "suspect".
- Making available to reviewers all information to validate a consistency link (items 83 and 84 in Figure 4) :
  - The CL attributes : previous status and rationale
  - The context of the consistency link: to validate a flow, both source and destination function are necessary information; to validate a function, received flows (and their source functions) and sent flows (and their destination functions) are required.

The validation of a CLF shall address:

- Function structure: both reviewers agree with similar or difference of structure between the two models, typically gathering, splitting, ignoring or adding nodes in MBSA model with respect to the MBSE model. For example, id and names of functions are considered.
- Interface: considering the set of functions linked by the CLF, both reviewers agree with the interface of the set. For example, name and direction of flows are considered.

- Behavior: considering the sets of MBSE functions and MBSA functions linked, both reviewers agree that the behavior is consistent. This method takes into account notably sub functions, flows between functions (internal flows). Behavior may be made of formal description (e.g. state machines), informal (e.g. text) or unmodeled.

The validation of a CLfl shall address:

- Flow structure: both reviewers agree with similar or difference of structure between the two models, typically gathering, splitting, ignoring or adding flows in MBSA model with respect to the MBSE models.
- Behavior: considering the sets of MBSE flows and MBSA flows linked, both reviewers agree that the behavior is consistent. In particular, the possible values of a flow in the MBSA model shall be consistent with the flow definition in the MBSE model.

It is important to capitalize the review discussions, both in terms of consistency rationale and actions to perform to solve inconsistencies (items 85 and 86 in Figure 4). After the review, the consistency links statuses are either “Validated” or “In revision” (in case there is some pending action to perform).

In the particular case of the existence of a “Change Process” to drive the system and models evolutions, the review scope can be defined in relation with the CRs that correspond to the evolutions perimeter. For each CR, the involved suspect CL are reviewed. This is useful for the system engineer in charge of this system evolution, so he can validate that the impacts on the safety model has been correctly taken into account.

At the end of the review, the suspect CL that are not part of any CR are reviewed. In general, they correspond to minor changes of SE that were not addressed by any CR.

Sometimes, the review takes place in the particular context of formal project or certification reviews, which are usually directed by a specific process. In this case, the consistency links review can be integrated in this process. It would require especially the formalization of a review report, with for example the following information:

- Review date and reviewers
- References to MBSE and MBSA models versions, associated consistency links set and their authors.
- References of used tools (modelling tool, review tool, connectors,…) and their versions.
- Scope of the review (in terms of CR or consistency link list)
- Status after the review: number of consistency links sets to “Validated” or “In revision”, with associated actions or recommendation in the second case.

We consider that the validation of all the consistency links ensure the validation of local consistency of both models:

- The CLset covers exhaustively both models
- With the correctness rules and review, the structure consistency is ensured
- While reviewing the structure, behavior may be discussed and validated (for the time being, the method does not address formally the behavior review, in particular, the changes of the behavior)
- Each CL is validated at least once in a review involving both system and safety engineers. Significant structure changes (as defined in 3.4) are mandatorily followed by a review (through the suspect status of involved CLs).

4 Validation activities

The method is applied on a study case of size representative of a small aeronautics system, the AIDA system, with the support of dedicated tool. Both the study case and the tool define the proof-of-concept of the method.

The current state of the proof-of-concept validates:

- the need of modeling flexibility and the ability of the method to support it,
- the usability of the defined rules of consistency links for functions and flows, and their efficiency notably in addressing large number of model elements,
- the efficiency of the defined rules of consistency links to detect mismatch in flow consistency,
- the feasibility of model changes detection and report by consistency links,
- the feasibility of a review focused on model changes and structured by consistency links,
- the feasibility of tooling support for consistency link definition and review,
- the relevance of CLs to capitalize discussions and justifications.

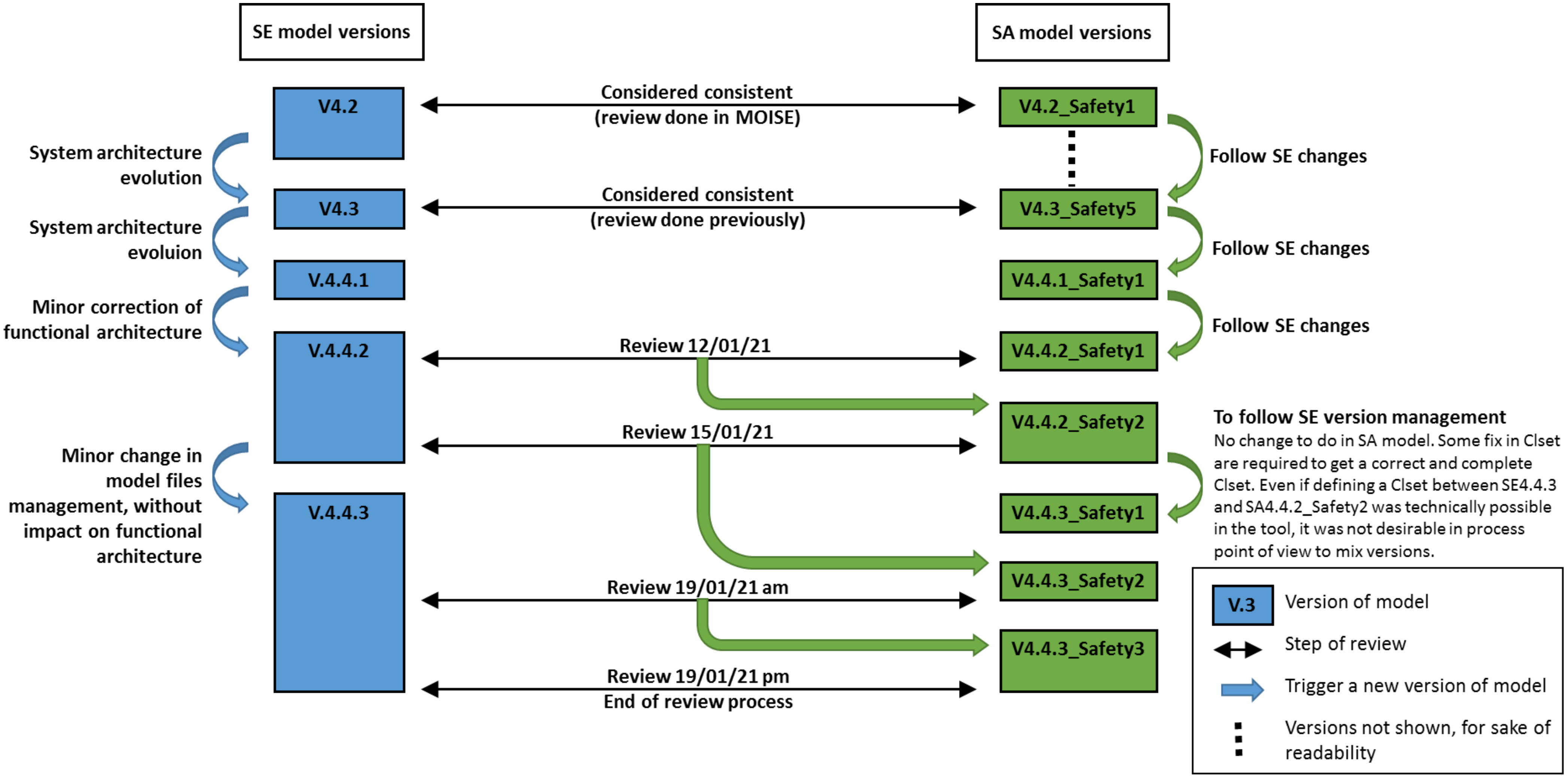
4.1 Proof-of-concept

4.1.1 AIDA study case

As this section is common with the Section 2.5 of LIV-S085L02-007-V6, ISX-S2C-LIV-1037-V6, text is communalized into this last one.

For the needs of document the study case evolved and Figure 17 gives an overview of the different model versions of AIDA.

The validation of definition of consistency links (Section 4.1.2(b)) is illustrated on the version 4.3. Iterations between 4.3 and different subversions of 4.4 are used to illustrate the update of SA model (Section 4.3), the update of consistency links (Section 4.4) and their review (Section 4.5).



No change to do in SA model. Some fix in CIset are required to get a correct and complete CIset. Even if defining a CIset between SE4.4.3 and SA4.4.2_Safety2 was technically possible in the tool, it was not desirable in process point of view to mix versions.

Figure 17: Summary of model versions used for validation activities

4.1.2 Consistency management tools

Although the method can be applied manually (as it was done with its ancestor on MOISE project cf. LIV-S-014-S2.21-61-457-V1), project's team considered that tooling is mandatory to improve performance but without jeopardizing the generic aspect of the method. This desire leads to identify the **phases** and the associated functions (allocation and definition are in section (a)) that are required to support the method.

Regarding Figure 4, the **phases** to implement are:

- The definition of the context, which includes: a version of SE model, a version of SA model (as published like items 3 and 58 for SE) and a version of a CLset (initially void or manually prepopulated regarding the operator needs before the edition phase). This concerns the publication evocated in Sections 3.1.1, 3.1.3 and 3.4
- The edition of CLs in the previous defined context (items 4, 24 and 69). This concerns Section 3.4.
- The evaluation of changes of inputs (items 59) and the consequence on the CLs. This concerns Sections 3.3 and 3.4.
- The support of review (item 11, 36 and 80) and the consequence on the CLs. This concerns Section 3.5.

A common infrastructure called TeePee (legated from MOISE project) is underneath to all these phases. It ensures the functions to store CLs, to extract and to abstract data from models (evocated in Section 3.2.1 and 0). It includes also an administration interface that ensures the **first phase** by carrying functions to reference the domain's models, to initiate/modify and to save a context pointing the referenced models. Although, this phase is mandatory, it is not a core part of the method so this tool is not detailed.

The **second** and **fourth phases** are supported by a graphical tool called review interface. It carries functions to load a context, create/delete a CL into/from context, show CLs and annotate them, filter: CLs and annotations, navigate to a CL and its adjacent CLs, show models artefacts and CLs linked to them regarding the navigation origin zone, navigate between the one displayed, save or derive the modified context and check CLs against it.

It shall be noticed that:

- The tool cannot distinguish the review from the edition so all capabilities of tool are available in both phases.
- The 'annotate' function (e.g. rationale, status and others data carried on by CL) shall be supported by the review interface (per design) but due to bugs, the annotations are done via the capabilities of administration interface. This work around is not detailed and readers shall consider that this function is ensured as expected by the review interface.

The **third phase** is supported via a command line interface tool called combination script. It carries the function to rule the comparisons induced by changes and to propose status regarding CLs that trigger the rules.

(a) Review interface

Figure 18 shows the different zones displayed by the graphical review interface and followed by their description

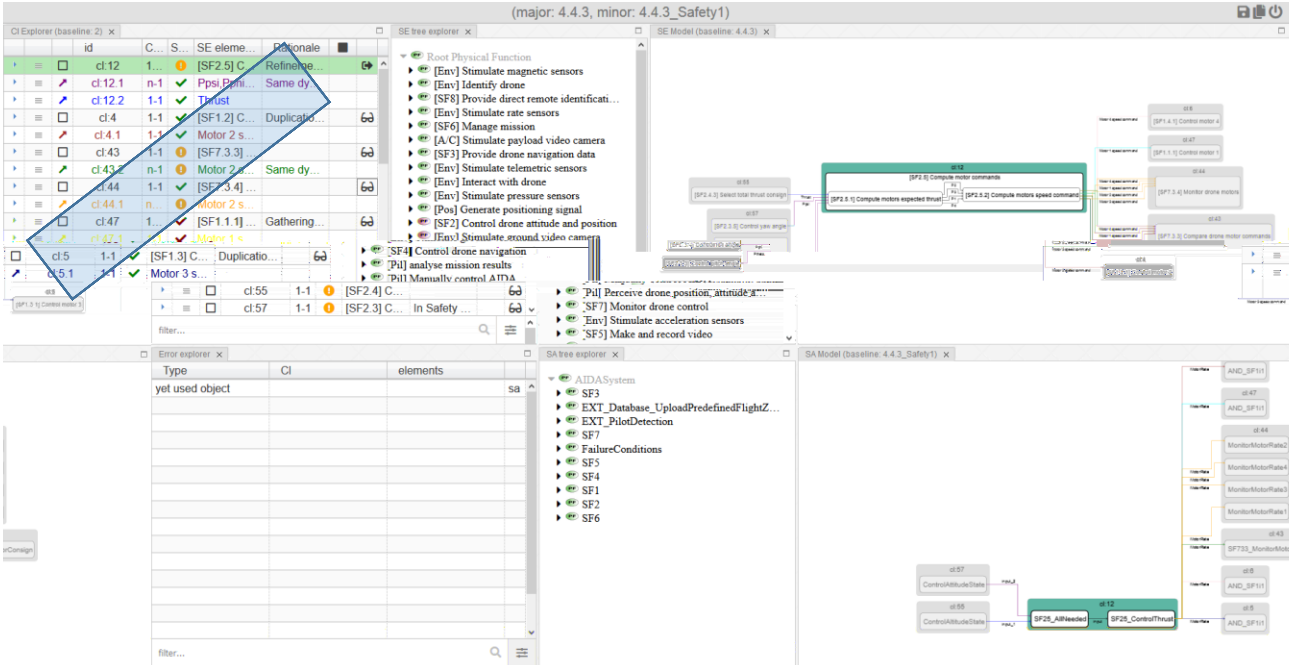

(major: 4.4.3, minor: 4.4.3_Safety1)
Cl Explorer (baseline: 2)
SE tree explorer
SE Model (baseline: 4.4.3)
id
SE eleme...
Rationale
Root Physical Function
[Env] Stimulate magnetic sensors
[Env] Identify drone
[SF8] Provide direct remote identificati...
[Env] Stimulate rate sensors
[SF6] Manage mission
[A/C] Stimulate payload video camera
[SF3] Provide drone navigation data
[Env] Stimulate telemetric sensors
[Env] Interact with drone
[Env] Stimulate pressure sensors
[Pos] Generate positioning signal
[SF2] Control drone attitude and position
[SF4] Control drone navigation
[Pil] analyse mission results
[Pil] Perceive drone position, attitude
[SF7] Monitor drone control
[Env] Stimulate acceleration sensors
[SF5] Make and record video
Error explorer
Type
Cl
elements
yet used object
filter...
SA tree explorer
SA Model (baseline: 4.4.3_Safety1)
AIDASystem
SF3
EXT_Database_UploadPredefinedFlightZ...
EXT_PilotDetection
SF7
FailureConditions
SF5
SF4
SF1
SF2
SF6

Figure 18: Overview of the consistency management tool's graphical user interface

The functions described in Section §4.1.2 are dispatched regarding the zones in Figure 18 as follow:

Zone	Functions	Zone	Functions	Zone	Functions
CL Explorer	Show	SE Tree Explorer	Show	SE Model	View
CL Explorer	Annotate	SE Tree Explorer	Navigate	SE Model	Navigate
CL Explorer	Filter	Context Setter	Load	SE Model	Create/Delete
CL Explorer	Navigate	Context Setter	Save	SA Model	View
CL Explorer	Filter	Context Setter	Derive	SA Model	Navigate
Error Explorer	Show	SA Tree Explorer	Show	SA Model	Create/Delete
Error Explorer	Filter	SA Tree Explorer	Navigate		

Table 3: Allocation of Function against Graphical Zone of the Review interface

The Context Setter Zone concerns the context of the CL, namely the SE, SA versioned models used as reference and the CLs set definition (if some exists already due to a precedent working session on this context). It allows to select a context and save changes done on it. The operator can save the current CL definitions by overwriting the former ones or by deriving the context into a new one where SE and SA referenced model are kept intact but CLs definitions differs from the original context.

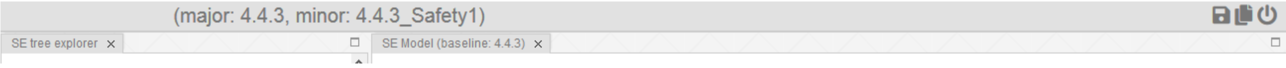

(major: 4.4.3, minor: 4.4.3_Safety1)
SE tree explorer
SE Model (baseline: 4.4.3)

Figure 19: Screenshot of the Context Setter Zone (left upper corner)

The CL explorer Zone offers an overall view of consistency links created over the MBSE and MBSA models referenced by the context selected via the **Context Setter Zone**. The CL explorer's rows represent CLs while columns (sortable) display information about the consistency links itself:

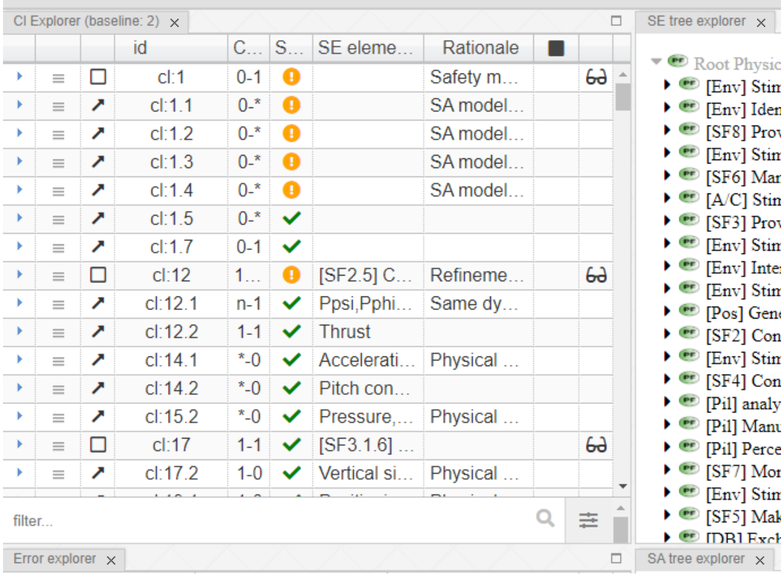
- their nature (function or functional flow),
- their unique identifier,
- their cardinality,
- their last reviewed status,
- their related model elements
- their rationale

Other information (links to change request identifier, comments, etc.) are available when the leftmost-icon of a row is used, as shown Figure 27. CL annotation offers edition for each CL individually and for a selected group of CL (multiple edition).

The **CL Explorer Zone** offers filtering means (basic and advanced) to focus on subsets of CL as illustrated in Figure 26. The advanced filtering is useful to prepare review by defining a limited scope of review (steps 13, 38 and 82 of Figure 4).

The rightmost column (glasses icon for CL functions only) of this panel selects the consistency link of interest. Once it is selected (See Figure 20), then automatically:

- a filter is applied to the zone and shows the selected CL on top then its neighbor CL:
  - CL flows linked to functional flows connecting the functions included in the selected consistency link,
  - CL functions related to functions that are the source or the destination of functional flows connecting the functions included in the selected consistency link.
- the SE Model Zone and SA Model Zone are updated with the CLs filtered (previous point).

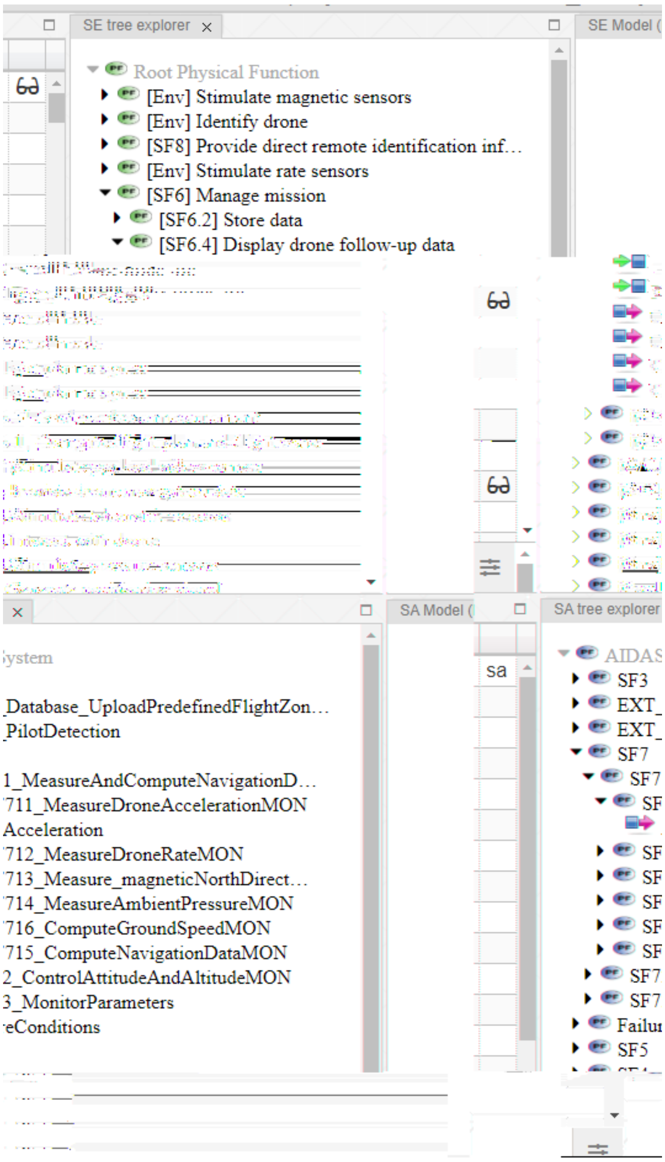


			id	C…	S…	SE eleme…	Rationale	■	
▸	≡	□	cl:1	0-1	!		Safety m…		👓
▸	≡	↗	cl:1.1	0-*	!		SA model…		
▸	≡	↗	cl:1.2	0-*	!		SA model…		
▸	≡	↗	cl:1.3	0-*	!		SA model…		
▸	≡	↗	cl:1.4	0-*	!		SA model…		
▸	≡	↗	cl:1.5	0-*	✓				
▸	≡	↗	cl:1.7	0-1	✓				
▸	≡	□	cl:12	1…	!	[SF2.5] C…	Refineme…		👓
▸	≡	↗	cl:12.1	n-1	✓	Ppsi,Pphi…	Same dy…		
▸	≡	↗	cl:12.2	1-1	✓	Thrust			
▸	≡	↗	cl:14.1	*-0	✓	Accelerati…	Physical …		
▸	≡	↗	cl:14.2	*-0	✓	Pitch con…			
▸	≡	↗	cl:15.2	*-0	✓	Pressure,…	Physical …		
▸	≡	□	cl:17	1-1	✓	[SF3.1.6] …			👓
▸	≡	↗	cl:17.2	1-0	✓	Vertical si…	Physical …		

Figure 20: Screenshot of the CL Explorer Zone filled for illustration

The **SE Tree Explorer Zone** and **SA Tree Explorer Zone** are expandable/collapsible trees that represent the functional breakdown. When a function is selected from an explorer (whatever its depth) its respective view (the **SE Model Zone** or **SA Model Zone**) is automatically update to show:

- Selected function from model explorers or functions linked to the CL selected in CL explorer
- Child functions of selected function
- Input and output flow of selected function
- Neighbor functions of selected function



Root Physical Function
[Env] Stimulate magnetic sensors
[Env] Identify drone
[SF8] Provide direct remote identification inf…
[Env] Stimulate rate sensors
[SF6] Manage mission
[SF6.2] Store data
[SF6.4] Display drone follow-up data

ystem
_Database_UploadPredefinedFlightZon…
_PilotDetection
1_MeasureAndComputeNavigationD…
711_MeasureDroneAccelerationMON
Acceleration
712_MeasureDroneRateMON
713_Measure_magneticNorthDirect…
714_MeasureAmbientPressureMON
716_ComputeGroundSpeedMON
715_ComputeNavigationDataMON
2_ControlAttitudeAndAltitudeMON
3_MonitorParameters
eConditions

AIDAS
SF3
EXT_
EXT_
SF7
SF7
SF
SF
SF
SF
SF
SF
SF7
SF7
Failu
SF5

Figure 21: Screenshot of the SE tree explorer zone (up) and SA tree explorer zone (down) expanded for illustration

The graphical views offered by **SE Model Zone** and **SA Model Zone** are triggered either by selection of CL in **CL explorer Zone** (see Figure 24), or by selection of functions in **SE Tree Explorer Zone** and **SA Tree Explorer Zone** (See Figure 22 for SE). Model Zones draw autonomously dataflow diagrams in a style independent form authoring one but in accordance to the modeling tool conventions see Section 3.2.1. The drawn items are overlaid with a layer carrying the CL link on them. In case of perimeter selected from **SE Tree Explorer Zone** and **SA Tree Explorer Zone,** CL flows are gray-colored while in the other case they are colored (see Figure 24) in accordance with the arrow symbol color in **CL Explorer Zone**

CL Creation and Deletion can be done via a right mouse click on the concerned items.

A navigation bar (See Figure 23) allows user to focus on a particular CL of the perimeter by centering the drawing zone on it without changing the layout of the diagram. The CL function having focus of navigation is in light green colored.

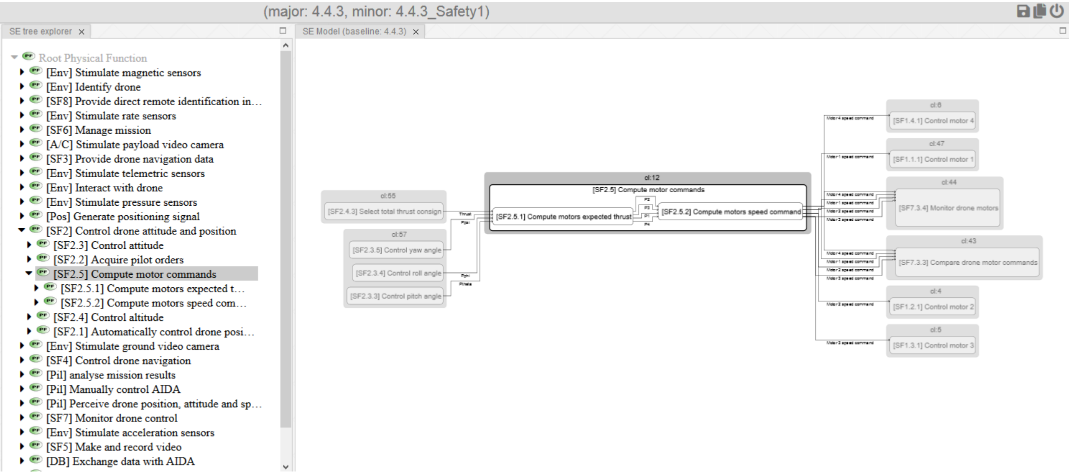

(major: 4.4.3, minor: 4.4.3_Safety1)
SE tree explorer
Root Physical Function
[Env] Stimulate magnetic sensors
[Env] Identify drone
[SF8] Provide direct remote identification in…
[Env] Stimulate rate sensors
[SF6] Manage mission
[A/C] Stimulate payload video camera
[SF3] Provide drone navigation data
[Env] Stimulate telemetric sensors
[Env] Interact with drone
[Env] Stimulate pressure sensors
[Pos] Generate positioning signal
[SF2] Control drone attitude and position
[SF2.3] Control attitude
[SF2.2] Acquire pilot orders
[SF2.5] Compute motor commands
[SF2.5.1] Compute motors expected t…
[SF2.5.2] Compute motors speed com…
[SF2.4] Control altitude
[SF2.1] Automatically control drone posi…
[Env] Stimulate ground video camera
[SF4] Control drone navigation
[PiI] analyse mission results
[PiI] Manually control AIDA
[PiI] Perceive drone position, attitude and sp…
[SF7] Monitor drone control
[Env] Stimulate acceleration sensors
[SF5] Make and record video
[DB] Exchange data with AIDA
SE Model (baseline: 4.4.3)
cl:55
[SF2.4.3] Select total thrust consign
cl:57
[SF2.3.5] Control yaw angle
[SF2.3.4] Control roll angle
[SF2.3.3] Control pitch angle
Thrust
cl:12
[SF2.5] Compute motor commands
[SF2.5.1] Compute motors expected thrust
[SF2.5.2] Compute motors speed command
P1
P2
P3
P4
cl:6
[SF1.4.1] Control motor 4
cl:47
[SF1.1.1] Control motor 1
cl:44
[SF7.3.4] Monitor drone motors
cl:43
[SF7.3.3] Compare drone motor commands
cl:4
[SF1.2.1] Control motor 2
cl:5
[SF1.3.1] Control motor 3
Motor 4 speed command
Motor 1 speed command
Motor 2 speed command
Motor 3 speed command

Figure 22: Screenshot of the SE model zone (right) when a function (SF2.5) is selected from SE tree explorer zone for illustration

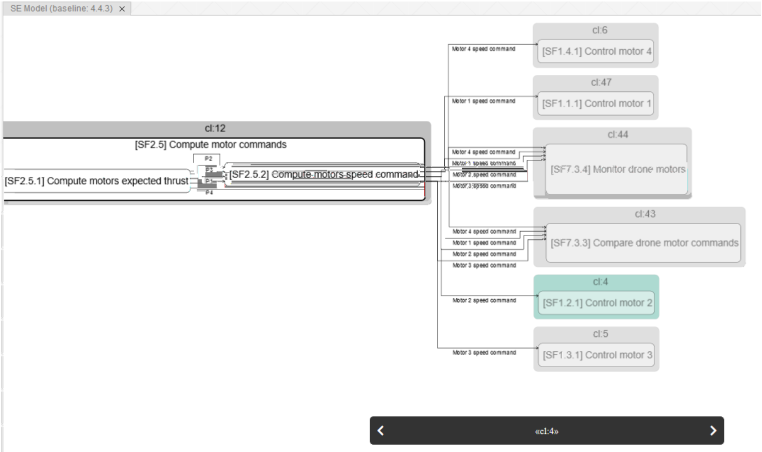
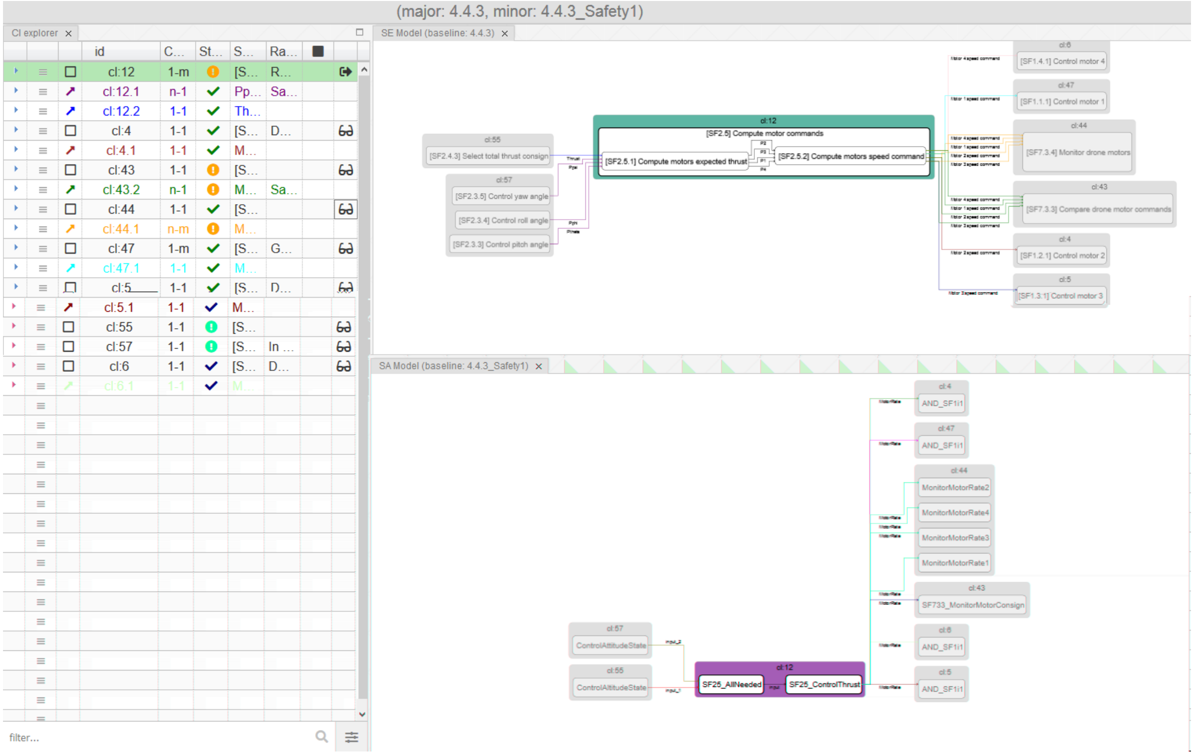

SE Model (baseline: 4.4.3)
cl:12
[SF2.5] Compute motor commands
[SF2.5.1] Compute motors expected thrust
[SF2.5.2] Compute motors speed command
P1
P2
P3
P4
cl:6
[SF1.4.1] Control motor 4
cl:47
[SF1.1.1] Control motor 1
cl:44
[SF7.3.4] Monitor drone motors
cl:43
[SF7.3.3] Compare drone motor commands
cl:4
[SF1.2.1] Control motor 2
cl:5
[SF1.3.1] Control motor 3
Motor 4 speed command
Motor 1 speed command
Motor 2 speed command
Motor 3 speed command
«cl:4»

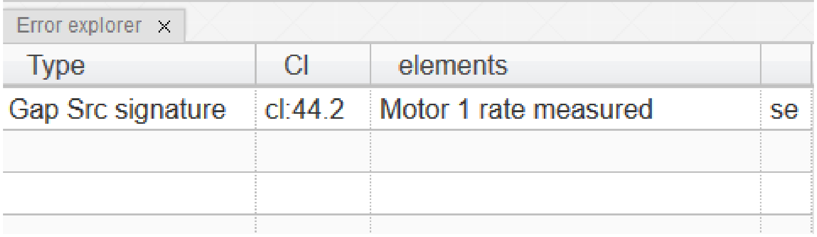
Figure 23: Screenshot of the navigation bar (down) on SE model zone while navigated CL is a neighbor of the cl linked to function (SF2.5) selected from SE tree explorer


(major: 4.4.3, minor: 4.4.3_Safety1)
Cl explorer
SE Model (baseline: 4.4.3)
SA Model (baseline: 4.4.3_Safety1)
cl:12
[SF2.5] Compute motor commands
[SF2.5.1] Compute motors expected thrust
[SF2.5.2] Compute motors speed command
cl:55
[SF2.4.3] Select total thrust consign
cl:57
[SF2.3.5] Control yaw angle
[SF2.3.4] Control roll angle
[SF2.3.3] Control pitch angle
cl:6
[SF1.4.1] Control motor 4
cl:47
[SF1.1.1] Control motor 1
cl:44
[SF7.3.4] Monitor drone motors
cl:43
[SF7.3.3] Compare drone motor commands
cl:4
[SF1.2.1] Control motor 2
cl:5
[SF1.3.1] Control motor 3
AND_SF1i1
MonitorMotorRate2
MonitorMotorRate4
MonitorMotorRate3
MonitorMotorRate1
SF733_MonitorMotorConsign
ControlAttitudeState
SF25_AllNeeded
SF25_ControlThrust
filter...

id	C...	St...	S...	Ra...
cl:12	1-m	!	[S...	R...
cl:12.1	n-1	✓	Pp...	Sa...
cl:12.2	1-1	✓	Th...	
cl:4	1-1	✓	[S...	D...
cl:4.1	1-1	✓	M...	
cl:43	1-1	!	[S...	
cl:43.2	n-1	!	M...	Sa...
cl:44	1-1	✓	[S...	
cl:44.1	n-m	!	M...	
cl:47	1-m	✓	[S...	G...
cl:47.1	1-1	✓	M...	
cl:5	1-1	✓	[S...	D...
cl:5.1	1-1	✓	M...	
cl:55	1-1	!	[S...	
cl:57	1-1	!	[S...	In ...
cl:6	1-1	✓	[S...	D...
cl:6.1	1-1	✓	M...	

Figure 24: Screenshot of the SE and SA model zone (right up and right down) when a CL (cl:12) is selected from CL explorer

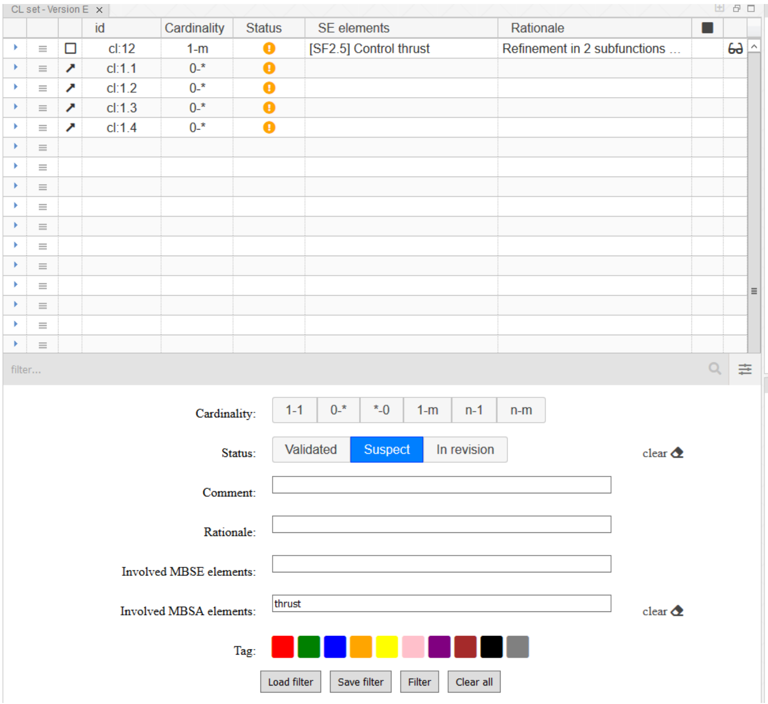
The **Error Explorer zone** shows the violated rules (defined in Section 3.2.2 and 3.2.3) and concerned items of models. The checks are run when operator save (or derive) its context using the **Context Setter Zone** capabilities. Results are displayed in a tabular fashion and a void table means that CLs definitions in the saved context are consistent and complete regarding the rules. A filtering mean is implemented to reduce the amount of rows displayed to focus on a particular set of them. Note that some errors may be hidden by an useless active filter or because a mandatory one has not been solved. But after the solving of such error less mandatory errors will be displayed. In the end the table is void.



Type	Cl	elements	
Gap Src signature	cl:44.2	Motor 1 rate measured	se

Figure 25: Screenshot of the Error Explorer for illustration

To conclude, with such a tool, the review proposed by the method in Section 3.5 can be performed. The SA specialist presents and justifies its modeled understanding respectively to the SE model via the complete and consistent CLs he did. The SE architect can access to the choices or assumptions done by the SA specialist in his model by starting from his own SE model that he recovers in tool (despite the neutral style) then using the consistent and complete CLs available to get the MBSA information.



CL set - Version E

id	Cardinality	Status	SE elements	Rationale
cl:12	1-m	!	[SF2.5] Control thrust	Refinement in 2 subfunctions ...
cl:1.1	0-*	!		
cl:1.2	0-*	!		
cl:1.3	0-*	!		
cl:1.4	0-*	!		

filter...

Cardinality: 1-1 | 0-* | *-0 | 1-m | n-1 | n-m

Status: Validated | [✓ Suspect] | In revision — clear

Comment:

Rationale:

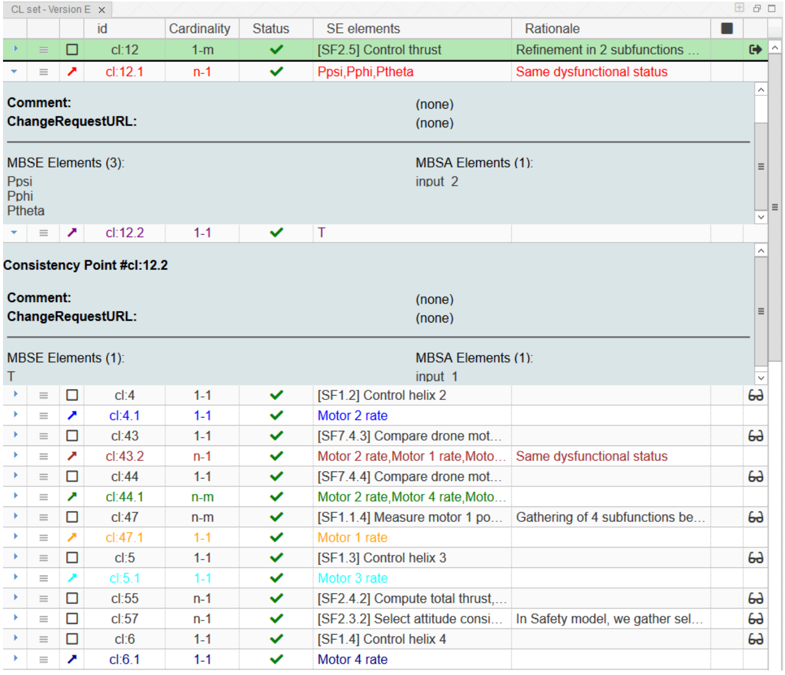
Involved MBSE elements:

Involved MBSA elements: thrust — clear

Tag: 🟥 🟩 🟦 🟧 🟨 ▢ 🟪 🟫 ⬛ ▨

Load filter | Save filter | Filter | Clear all

Figure 26: Screenshot of the consistency link tabular view when no consistency link is selected and filter applied



CL set - Version E

id	Cardinality	Status	SE elements	Rationale
cl:12	1-m	✓	[SF2.5] Control thrust	Refinement in 2 subfunctions ...
cl:12.1	n-1	✓	Ppsi,Pphi,Ptheta	Same dysfunctional status

Comment: (none)
ChangeRequestURL: (none)

MBSE Elements (3):
Ppsi
Pphi
Ptheta

MBSA Elements (1):
input 2

id	Cardinality	Status	SE elements	Rationale
cl:12.2	1-1	✓	T	

Consistency Point #cl:12.2

Comment: (none)
ChangeRequestURL: (none)

MBSE Elements (1):
T

MBSA Elements (1):
input 1

id	Cardinality	Status	SE elements	Rationale
cl:4	1-1	✓	[SF1.2] Control helix 2	
cl:4.1	1-1	✓	Motor 2 rate	
cl:43	1-1	✓	[SF7.4.3] Compare drone mot...	
cl:43.2	n-1	✓	Motor 2 rate,Motor 1 rate,Moto...	Same dysfunctional status
cl:44	1-1	✓	[SF7.4.4] Compare drone mot...	
cl:44.1	n-m	✓	Motor 2 rate,Motor 4 rate,Moto...	
cl:47	n-m	✓	[SF1.1.4] Measure motor 1 po...	Gathering of 4 subfunctions be...
cl:47.1	1-1	✓	Motor 1 rate	
cl:5	1-1	✓	[SF1.3] Control helix 3	
cl:5.1	1-1	✓	Motor 3 rate	
cl:55	n-1	✓	[SF2.4.2] Compute total thrust,...	
cl:57	n-1	✓	[SF2.3.2] Select attitude consi...	In Safety model, we gather sel...
cl:6	1-1	✓	[SF1.4] Control helix 4	
cl:6.1	1-1	✓	Motor 4 rate	

Figure 27: Screenshot of the consistency link tabular view when a consistency link is selected

(b) Combination script

This tool aims to identify the CLs impacted by changes between versions of referenced models and to propose consistent status (i.e. suspect) for them before the review. As it is a status proposal, it is up to the user, here the safety specialist, to modify (or not) the status himself via the **CL Explorer Zone**.

To achieve identification and propositions, the tool finds differences between the same model type of the two contexts (context is defined in Section 4.1.2(a)) and then correlates differences to CLs. One of the contexts is reputed reviewed (as the reference) and the other one is reputed to be reviewed (as the challenger). Some filtering is done on the set of differences found for the correlation. It is realized regarding the rules defined in Sections 3.3 and 3.4.

When the tool runs, the two contexts (given as inputs) shall be consistent and complete (so none of them infringes rules defined in Section 3.2.2 and 3.2.3). The tool's output will be the proposed list of CL whose status may be set to suspect by the user. Each CL of the list is accompanied with information to help the user to assess the relevancy of the tool's proposal, for example:

- Type of change that was trigged
- Artefact concerned by the trig

An example of combination script output is given by Figure 38.

Note that, such a tool can be integrated to the review interface.

4.2 Validation of consistency links definition for functional architecture

This section describes the experiment activities corresponding to the activities defined by the method in Section 3.2.

4.2.1 Validation of functional architecture as functions and flows

As stated in Section 3.2.1, an abstraction of MBSE and MBSA models is realized to ease their comparison and identify their inconsistencies. This abstraction is dependent from the considered viewpoint (here functional architecture) and shall be applicable to various MBSE and MBSA authoring tools, relying on heterogeneous languages.

Regarding the functional architecture, S2C used the functional flow viewpoint, already used in the MOISE project (See Appendix B for details)**.** The use of this abstraction on AIDA input models results in abstracted models (complying with the functional flow viewpoint) with these metrics:

Type of model element	MBSE model	Abstracted MBSE model	MBSA model	Abstracted MBSA model
Functions	159	159	148	148
Functional flows	285	285	438	196

Table 4: Metrics on abstracted MBSE and MBSA models on AIDA study case

For MBSE model, there is no difference as the abstraction used for the functional flow viewpoint is almost the same as the one embedded in Capella. However, for MBSA model, and as explained in section 3.2.1, there is a big difference for Functional Flows. The reduction of the number of model elements is key to reduce the review time to its minimum. Thus, removing intermediate Functional flows from MBSA models seems appropriate and feasible.

4.2.2 Validation of definition of consistency link for function

The AIDA study case reveals the flexibility offered by the consistency link method for functions explained in Section 3.2.2. In next paragraphs, an example of one consistency link (CL) illustrating this is exposed. It is related to the MBSE function named "[SF2.5] Control thrust", and referred as SF2.5 hereafter.

The MBSE model, made with Capella, contains a dedicated diagram (See Figure 28) to show the context of SF2.5 which is composed by ten other MBSE functions. It also shows that SF2.5 is composed of two sub-functions named "[SF2.5.1] Compute thrust" and "[SF2.5.2] Compute motor rate".

Figure 28: Copy of “[PDFB] Functional Flows SF2.5 Control thrust” diagram of the MBSE Capella model

The MBSA

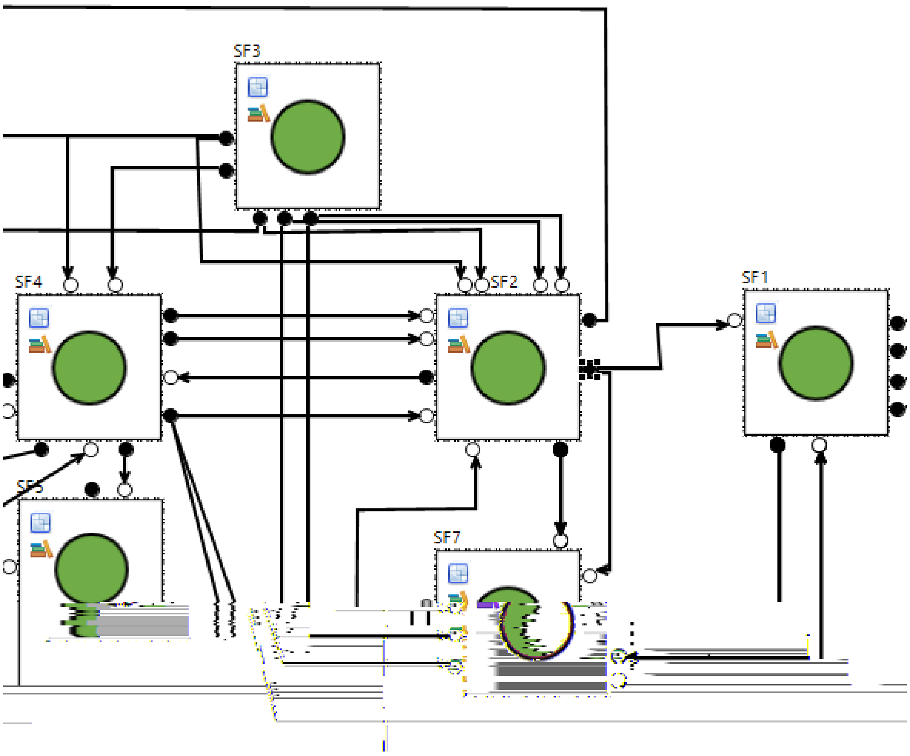
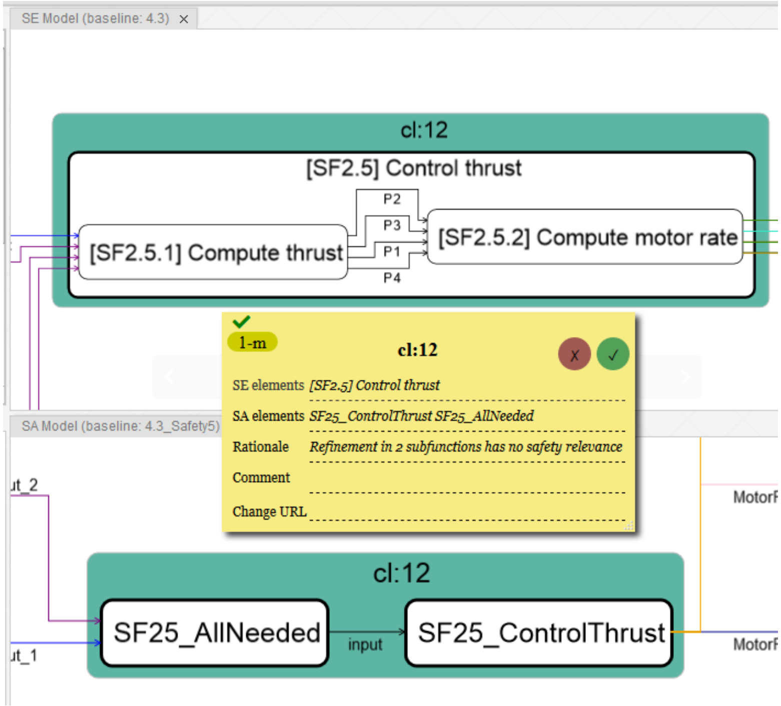


Figure 30: Extract of AIDASystem diagram of the MBSA SimfiaNeo model

Figure 31 is a capture of the consistency management tool displaying the consistency link defined for SF2.5 function where:

- The unique identifier of the consistency link is “cl:12”
- In the top MBSE model panel, “[SF2.5] Control thrust“ is displayed as included in cl:12
- In the bottom MBSA model panel, both “SF25_AllNeeded” and “SF25_ControlThrust” bricks are displayed as included in cl:12
- The center yellow note displays complementary information about the CL, notably:
  - The cardinality of consistency link is 1-to-m as there is one MBSE function and two MBSA bricks that are linked together (top left corner)
  - The status of the CL: validated (green check at top left corner)
  - The rationale



SE elements *[SF2.5] Control thrust*
SA elements *SF25_ControlThrust SF25_AllNeeded*
Rationale *Refinement in 2 subfunctions has no safety relevance*
Comment
Change URL

Figure 31: Extract of screenshot of the consistency management tool focused on cl:12

In this case, it has been decided to put the consistency link on a MBSE parent function (“[SF2.5] Control thrust“). Thus, in accordance with the rule stated in section 3.2.2(a), the leaf functions named “[SF2.5.1] Compute thrust” and “[SF2.5.2] Compute motor rate” have no dedicated consistency link. At the same time, the consistency link method on functions allows the MBSA model to have two different bricks to describe one MBSE function.

On the overall AIDA models, Table 5 shows some metrics for the consistency links for functions.

Cardinality of consistency link for function	Number of consistency link(s)
1-1	33
1 MBSE to n MBSA	2
n MBSE to 1 MBSA	7
n MBSE to m MBSA	2
0 MBSE to n or 1 MBSA	1
n or 1 MBSE to 0 MBSA	6
Total	**51**

Table 5: Metrics on consistency links for function on AIDA study case

Note that both models have more than 100 functions and that only 51 CL are necessary. It is achieved by:

- The fact that in a functional breakdown, only one level is linked to a CL. In the example, MBSE-side, the consistency of SF2.5.1 and SF2.5.1 is ensured indirectly by the CL of SF2.5.
- The possibility to gather several functions. In the example, MBSA-side, the consistency of both “SF25_AllNeeded” and “SF25_ControlThrust” is ensured by only one CL.

One third of the defined CL are different from the intuitive and simplest 1-1 CL, which validates the need to offer flexibility on consistency links definition:

- CLF linking one or several MBSE model elements to 0 MBSA functions are used for functions without safety impact, i.e., environment functions that cannot fail, and functions assigned to ground equipment that have no safety impact.
- The only CLF without MBSE function is linked to the MBSA model element used for FC observation.
- SF2.5 gives an example of CLF linking 1 MBSE function to several MBSA elements.

4.2.3 Validation of definition of consistency link for functional flows

In this section, the same example (focus on MBSE function named "[SF2.5] Control thrust") as in the previous section is used to validate the use of the consistency link method for functional flows explained in section 3.2.3.

Figure 28 already shows the functional flows connecting the SF2.5 function to its neighbors. More specifically, it shows functional flows connecting leaf functions, as defined in Section 3.2.1. That is to say that only one functional flow is necessary to connect functions that do not belong to the same parent. Thus, the SF2.5 function has the following inputs and outputs:

- 4 inputs coming from 4 different leaf functions,
- 4 outputs going to 6 different leaf functions.

On the MBSA model side, the "SF25_AllNeeded" brick has the following inputs and outputs:

- 2 inputs coming from 2 different parent functions,
- 2 outputs going to 2 different parent functions.

To observe the differences between MBSE and MBSA models in terms of functional flows (instead of ports and segments), more diagrams from MBSA are required. To show the context of SF2.5, as Figure 28 does in MBSE side, several diagrams from MBSA have to be linked. From the central screenshot showing SF2.5 modeled in MBSA, Figure 33 shows the details of source function of input flows of SF2.5 (left hand of the figure) and the destinations of the output flow.

On the other hand, the "inter-domain correctness rule" (defined in Section 3.2.3(c)) has been implemented and the detection of mismatch (illustrated in Figure 13) in flow consistency has been validated. For example, the case of source mismatch shown in Figure 32 has been detected by the consistency management tool (even if this detection is not yet displayed by the graphical interface).

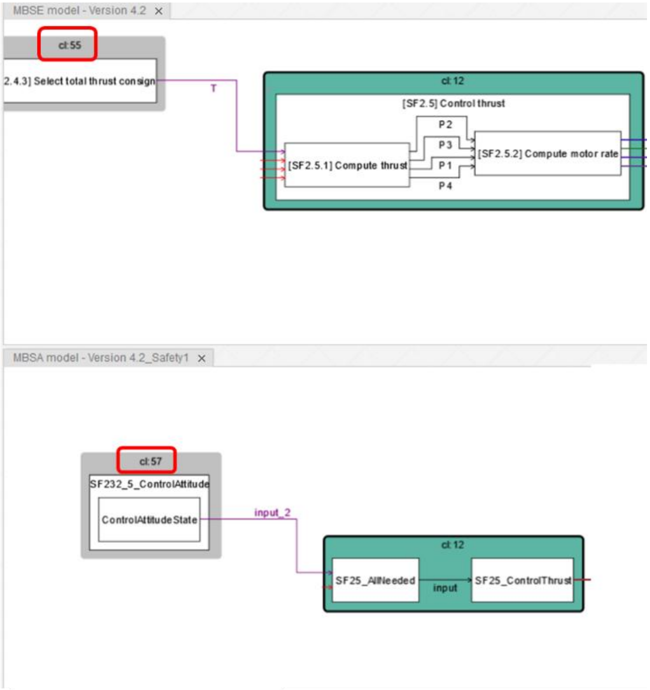


Figure 32: Extract of screenshot of the consistency management tool. The two CLF squared in red do not match.

Figure 34 is a zoom out of Figure 31, displaying the whole tool interface with the chosen focus on cl:12. The tabular left part displays the list of CLfl involved and the list of CLF of neighbor functions.

In input of SF2.5 function, two consistency links are defined to ensure the consistency of functional flows between the MBSE and MBSA models:

- one, with the unique identifier cl:12.1 and displayed in purple for:
    - 3 MBSE functional flows (Ptheta, Pphi, and Ppsi),
    - MBSA functional flows (input_2),
    - A rationale.
- The other one, with the unique identifier cl:12.2 and displayed in blue for:
    - 1 MBSE functional flow (T),
    - 1 MBSA functional flow (input_1).

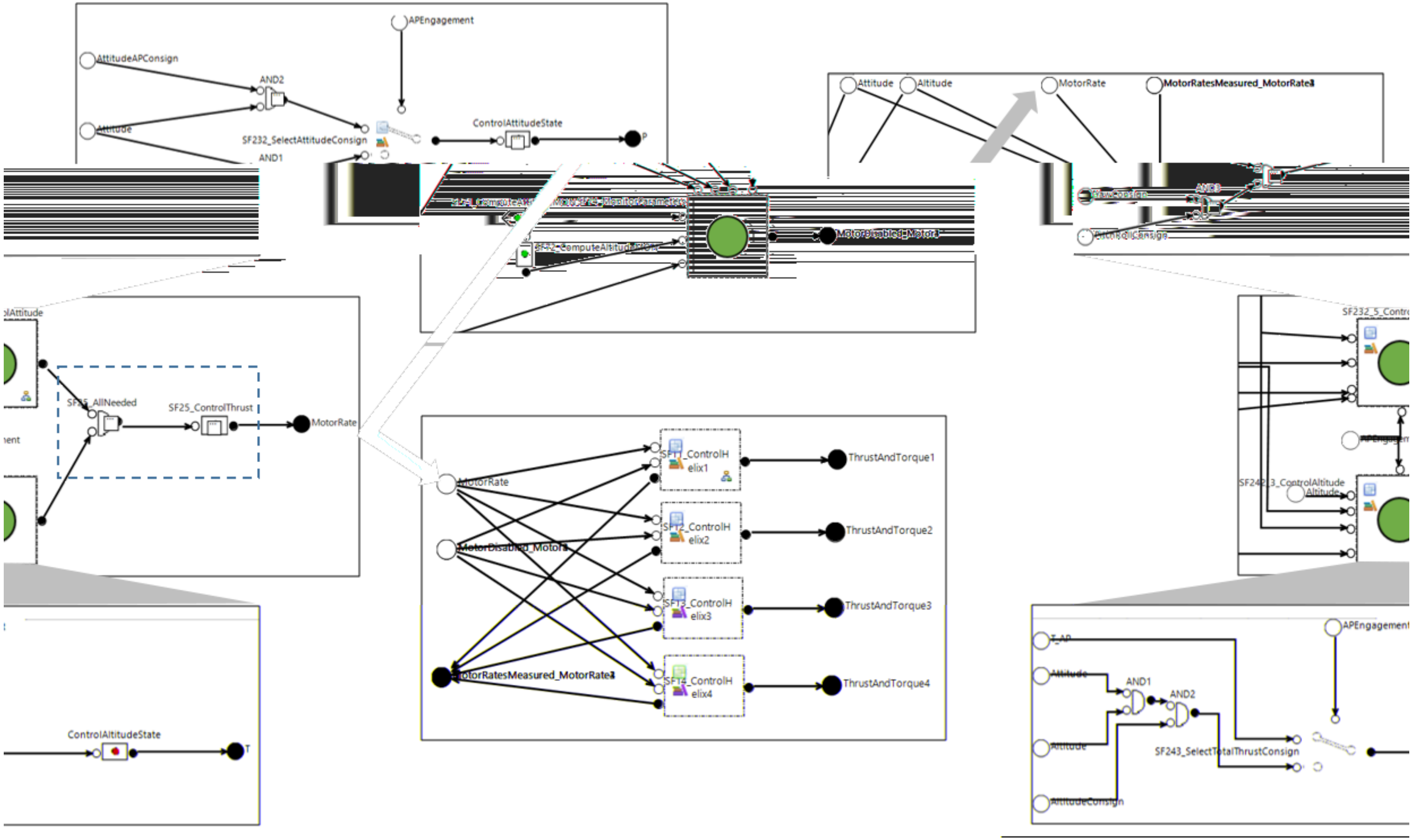
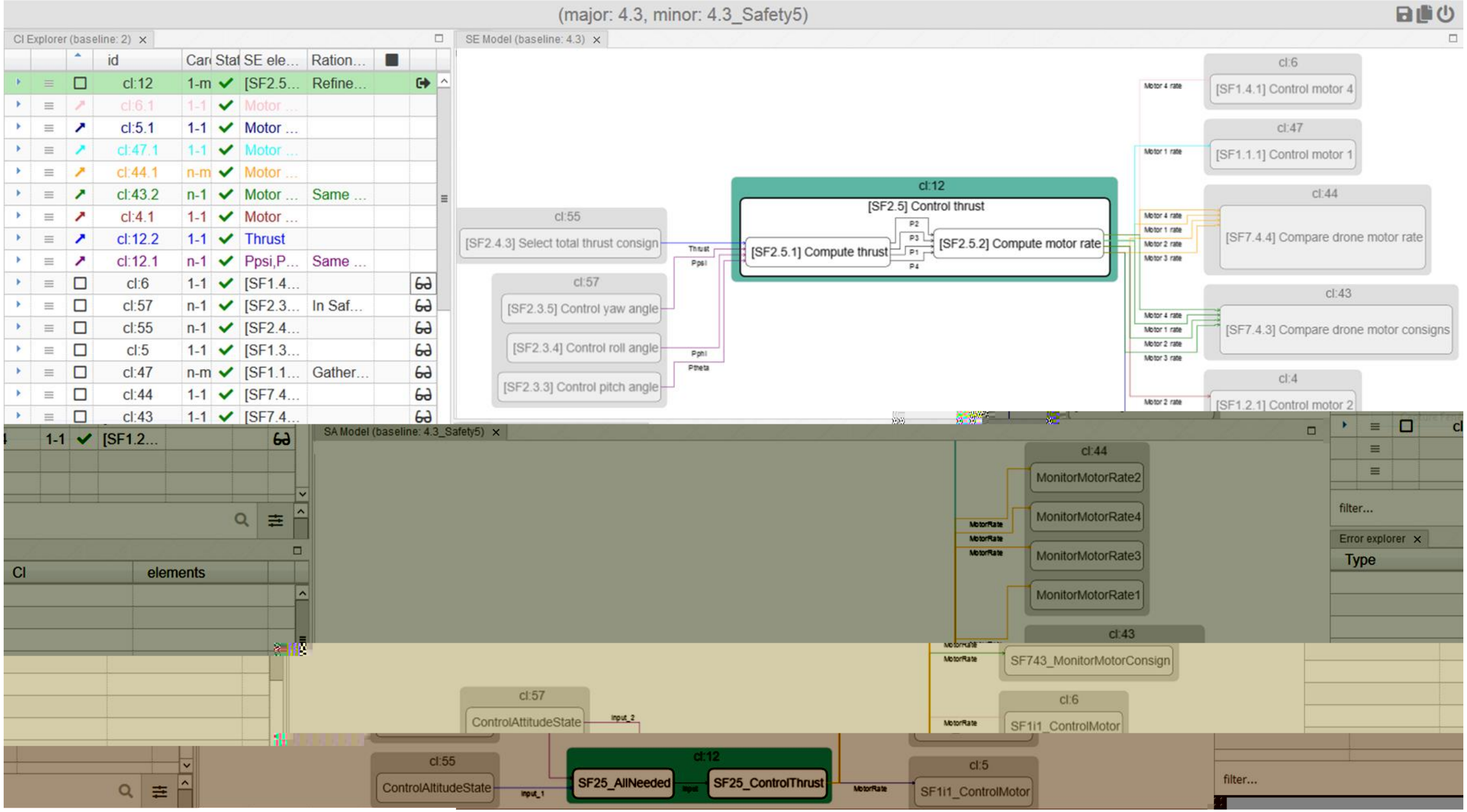


Figure 33: Combined screenshots of MBSA SimfiaNeo diagrams for the SF2.5 function



(major: 4.3, minor: 4.3_Safety5)

CI Explorer (baseline: 2)

			id	Car	Stat	SE ele...	Ration...	■	
	≡	□	cl:12	1-m	✓	[SF2.5...	Refine...		
	≡	↗	cl:6.1	1-1	✓	Motor ...			
	≡	↗	cl:5.1	1-1	✓	Motor ...			
	≡	↗	cl:47.1	1-1	✓	Motor ...			
	≡	↗	cl:44.1	n-m	✓	Motor ...			
	≡	↗	cl:43.2	n-1	✓	Motor ...	Same ...		
	≡	↗	cl:4.1	1-1	✓	Motor ...			
	≡	↗	cl:12.2	1-1	✓	Thrust			
	≡	↗	cl:12.1	n-1	✓	Ppsi,P...	Same ...		
	≡	□	cl:6	1-1	✓	[SF1.4...			
	≡	□	cl:57	n-1	✓	[SF2.3...	In Saf...		
	≡	□	cl:55	n-1	✓	[SF2.4...			
	≡	□	cl:5	1-1	✓	[SF1.3...			
	≡	□	cl:47	n-m	✓	[SF1.1...	Gather...		
	≡	□	cl:44	1-1	✓	[SF7.4...			
	≡	□	cl:43	1-1	✓	[SF7.4...			
				1-1	✓	[SF1.2...			

Figure 34: Screenshot of the consistency management tool focused on SF2.5 and the associated cl:12

It shall be noticed that, as stated in Section 3.2.3, only the functional flows external to a defined consistency link for function have a consistency link for flows. In Figure 19, the flows P1 to P4 are for example internal to the CLF defined in Section 4.2.2. On the overall AIDA models, Table 6 shows some metrics for the consistency links for flow.

Cardinality of consistency link for functional flow	Number of consistency link(s)
1-1	60
1 MBSE to n MBSA	7
n MBSE to 1 MBSA	20
n MBSE to m MBSA	3
0 MBSE to n or 1 MBSA	9
n or 1 MBSE to 0 MBSA	36
Total	**135**

Table 6: Metrics on consistency links for flow on AIDA study case

Note that less than half of CLfl are 1-1, validating the need for structural difference and model flexibility.

4.2.4 Opportunities and difficulties

By practicing the method on the study case, we observed some difficulties and opportunities in method application, independently from the tool.

(a) Opportunity: flexible hierarchy modeling

A repeated pattern of consistency, illustrated in Figure 35, is the following:

- A function (e.g., SF2.3) is decomposed in sub functions in SE model
- Safety analysis does not require the level of detail of the sub functions
- Safety modeling uses library of predefined model elements.

In particular, to model SF2.3, safety specialist uses the following predefined elements:

- AND1 and AND2 (two instances of the same class) dedicated to logical combination of input failure modes
- Selection to model the selection of the channel considering the AP engagement
- ControlAttitudeState to model the failure modes internal to SF2.3 and how they impact the output of SF2.3

The function SF2.3 is modeled as a whole from the safety point of view (in particular, only one loss failure mode and only one erroneous failure mode). Nevertheless, the safety modeling is eased by the use of predefined elements, notably to model the behavior. The way to model is a good practice in safety modeling and eases the model behavior validation.

In safety model, the perception of global torque and thrust has been modeled as a direct flow from SF1 to the pilot.

As a consequence the function "[Env] Interact with drone" and its input and output flows are equivalent to the direct flow of safety model. Nevertheless, regarding the checking rule 3.2.3(b), we cannot define such a consistency in our method. A solution would be to link "[Env] Interact with drone" either to the CLF of SF1 or to the CLF of the pilot. Gathering an environment function either with a system function or with the pilot function has not been seen as satisfying. As shown in Figure 37: Overview of the achieved consistency, the input and output flows of "[Env] Interact with drone" are linked to a CLfl without corresponding flow in safety model, and similarly, the flow of the safety model is linked to a CLfl without corresponding flow in SE model.

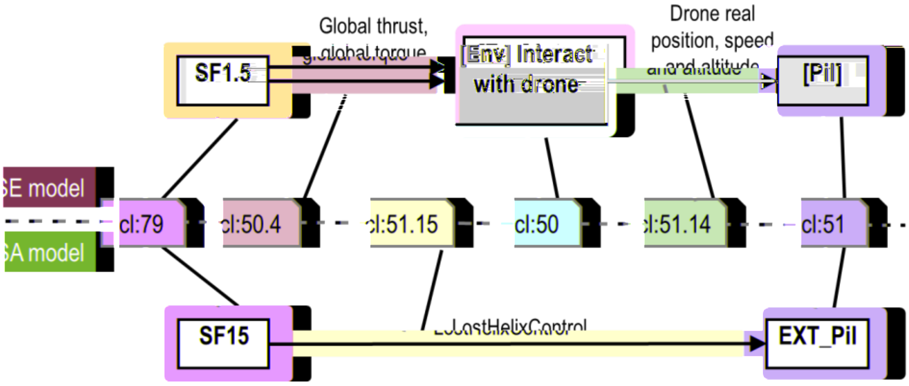

Global thrust, global torque
SF1.5
[Env] Interact with drone
Drone real position, speed and altitude
[Pil]
SE model
SA model
cl:79
cl:50.4
cl:51.15
cl:50
cl:51.14
cl:51
SF15
LostHelixControl
EXT_Pil

Figure 37: Overview of the achieved consistency

4.3 Validation of CL usage to get SA model impacts induced by SE model changes

As explained in Section 3.3, this part of the method has not been addressed in details and deserves further studies. We only describe here how the related activities have been performed in the context of the proof-of-concept..

We have chosen the first of the three solutions described in Section 3.3: SE model changes have been taken into account in SA model independently from the CL method and tool.

After the version 4.3 of SE model, a new version 4.4.1 is published by the safety architect. As a light change process is used to manage AIDA evolutions, the changes done are notably described in Change Requests (CR).

The safety specialist has used the diff feature of Capella to compare the versions of SE model: 4.3 is the starting version, 4.4.1 is the target version (refer to Figure 17 for an overview of model versions). From these differences, also guided by the change requests of the architect, a new version 4.4.1_Safety1 of safety model has been built.

Before the beginning of reviews, a "last minute" change has been introduced in the system functional architecture, resulting in the version 4.4.2 of SE model. The safety specialist has then published the version 4.4.2_Safety1 to follow the changes of SE. During the review process, another version of SE model is published (4.4.3), resulting in an adaptation of the CL set.

The consistency link method is flexible to these publications even if the reviews are not done or not completed. This is necessary to be applicable in to real life conditions, when processes are sometimes adjusted to project needs and timelines.

Each time a new version of safety model is published, an associated version of covering and correct CLset is also published. Moreover, each version of reviewed safety model has at least two different versions of CLset:

- The first one is prepared by the safety specialist before the review, and is the result of activities described in Sections 4.4.
- The second one is the result of the review described in Section 4.5.

4.4 Validation of consistency link update with respect to SE model, SA model and CL set changes

This section describes the experimental activities corresponding to the ones defined by the method in Section 3.4.

From SE model versions 4.3 and 4.4.1, the volume of changes was significant and result in three steps of modeling. At each step, the CL set was modified locally to record the modeling choices of the step. For each following versions, the CL set adaptation was done in only one-step as the volume of changes is little.

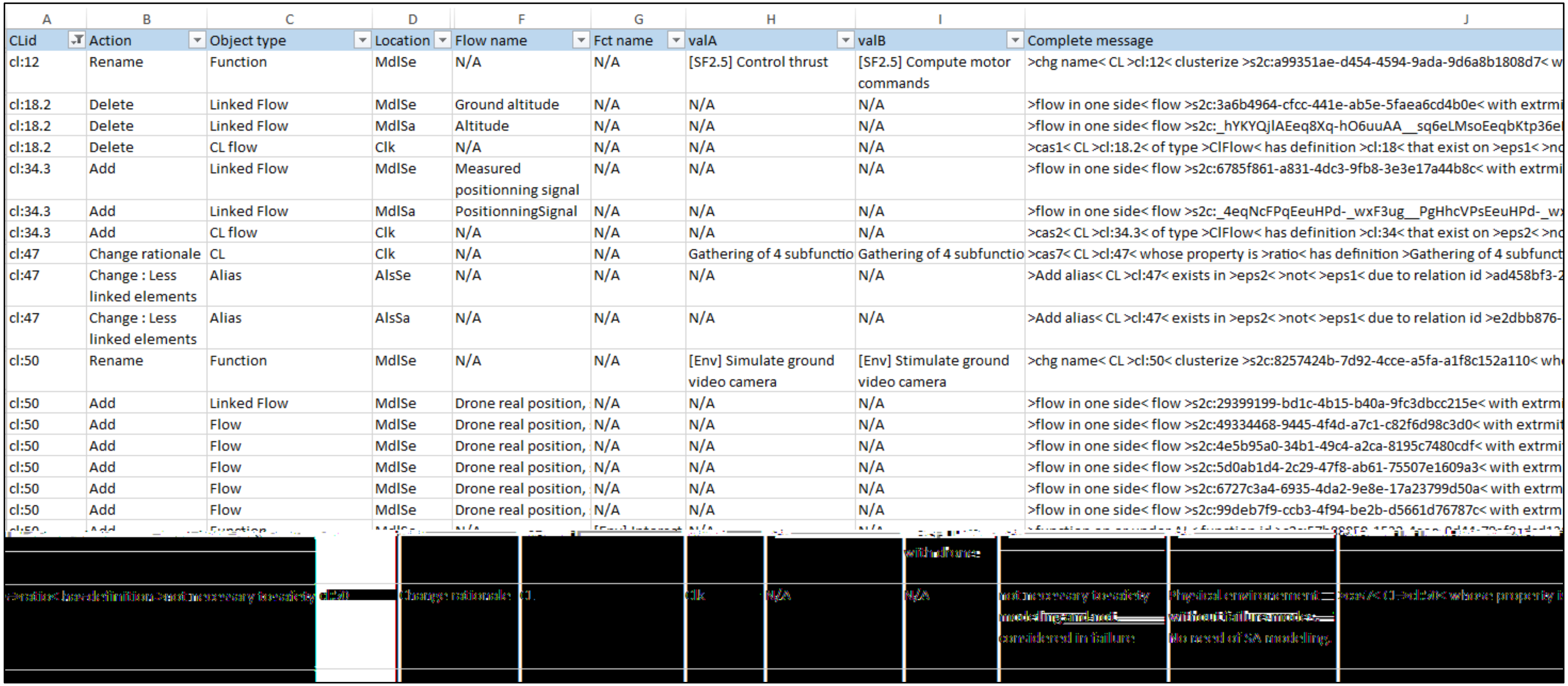
Once the safety model is stable and the associated CL set is covered and correct, the implemented consistency management tool computes the propositions of suspect statuses based on the different versions of models and CL sets. These propositions are displayed to the user (i.e., the safety specialist) in Excel format.

Figure 38 shows an extract of such propositions file. The first column shows the CL id, columns B and C the type of difference. Columns D mentions in which model the difference happen and column F specifies: the particular element implied in the difference. Finally, column J gives a complete and detailed message mentioning the id of the different elements involved.

Each row is a difference between the previous and the new version of the CL set. Rows are grouped by CL id, enabling to read all difference happened in the scope of a CL and decide whether to follow or not the proposition to change its status to suspect. Note that if a CL is already suspect, it will be displayed as any other, function of the differences happened in its scope.

The extract of Figure 38 shows different cases of differences:

- Regarding the cl:12, we note that a renaming has been done in SE (but not “followed” in SA)
- cl:18.2 has been deleted of the CL set as well as its linked flows have been deleted from the SE and SA model
- cl:34.3 has been created, as well as its linked flow
- Regarding cl:47, some flows that were previously linked have been unlinked from cl:47 (but not removed from the model. Consequently, its rationale has been filled in.
- Regarding cl:50, many (internal) flows and functions have been added to model and to the scope of the CL



CLid	Action	Object type	Location	Flow name	Fct name	valA	valB	Complete message
cl:12	Rename	Function	MdlSe	N/A	N/A	[SF2.5] Control thrust	[SF2.5] Compute motor commands	>chg name< CL >cl:12< clusterize >s2c:a99351ae-d454-4594-9ada-9d6a8b1808d7< w
cl:18.2	Delete	Linked Flow	MdlSe	Ground altitude	N/A	N/A	N/A	>flow in one side< flow >s2c:3a6b4964-cfcc-441e-ab5e-5faea6cd4b0e< with extrmi
cl:18.2	Delete	Linked Flow	MdlSa	Altitude	N/A	N/A	N/A	>flow in one side< flow >s2c:_hYKYQjlAEeq8Xq-hO6uuAA__sq6eLMsoEeqbKtp36e
cl:18.2	Delete	CL flow	Clk	N/A	N/A	N/A	N/A	>cas1< CL >cl:18.2< of type >ClFlow< has definition >cl:18< that exist on >eps1< >no
cl:34.3	Add	Linked Flow	MdlSe	Measured positionning signal	N/A	N/A	N/A	>flow in one side< flow >s2c:6785f861-a831-4dc3-9fb8-3e3e17a44b8c< with extrmi
cl:34.3	Add	Linked Flow	MdlSa	PositionningSignal	N/A	N/A	N/A	>flow in one side< flow >s2c:_4eqNcFPqEeuHPd-_wxF3ug__PgHhcVPsEeuHPd-_wx
cl:34.3	Add	CL flow	Clk	N/A	N/A	N/A	N/A	>cas2< CL >cl:34.3< of type >ClFlow< has definition >cl:34< that exist on >eps2< >no
cl:47	Change rationale	CL	Clk	N/A	N/A	Gathering of 4 subfunctio	Gathering of 4 subfunctio	>cas7< CL >cl:47< whose property is >ratio< has definition >Gathering of 4 subfunct
cl:47	Change : Less linked elements	Alias	AlsSe	N/A	N/A	N/A	N/A	>Add alias< CL >cl:47< exists in >eps2< >not< >eps1< due to relation id >ad458bf3-2
cl:47	Change : Less linked elements	Alias	AlsSa	N/A	N/A	N/A	N/A	>Add alias< CL >cl:47< exists in >eps2< >not< >eps1< due to relation id >e2dbb876-
cl:50	Rename	Function	MdlSe	N/A	N/A	[Env] Simulate ground video camera	[Env] Stimulate ground video camera	>chg name< CL >cl:50< clusterize >s2c:8257424b-7d92-4cce-a5fa-a1f8c152a110< whe
cl:50	Add	Linked Flow	MdlSe	Drone real position,	N/A	N/A	N/A	>flow in one side< flow >s2c:29399199-bd1c-4b15-b40a-9fc3dbcc215e< with extrmi
cl:50	Add	Flow	MdlSe	Drone real position,	N/A	N/A	N/A	>flow in one side< flow >s2c:49334468-9445-4f4d-a7c1-c82f6d98c3d0< with extrmit
cl:50	Add	Flow	MdlSe	Drone real position,	N/A	N/A	N/A	>flow in one side< flow >s2c:4e5b95a0-34b1-49c4-a2ca-8195c7480cdf< with extrmi
cl:50	Add	Flow	MdlSe	Drone real position,	N/A	N/A	N/A	>flow in one side< flow >s2c:5d0ab1d4-2c29-47f8-ab61-75507e1609a3< with extrm
cl:50	Add	Flow	MdlSe	Drone real position,	N/A	N/A	N/A	>flow in one side< flow >s2c:6727c3a4-6935-4da2-9e8e-17a23799d50a< with extrm
cl:50	Add	Flow	MdlSe	Drone real position,	N/A	N/A	N/A	>flow in one side< flow >s2c:99deb7f9-ccb3-4f94-be2b-d5661d76787c< with extrm
cl:50	Add	Function	MdlSe	N/A	[Env] Interact	N/A with drone	N/A	>function on or under AL< function id >s2c:57b[illegible]
cl:50	Change rationale	CL	Clk	N/A	N/A	not necessary to safety modeling and not considered in failure	Physical environment without failure modes. No need of SA modelling.	>cas7< CL >cl:50< whose property i

Figure 38: Extract of the suspect status propositions

4.5 Validation of SA model review supported by the CLs

This section describes the experiment activities corresponding to the activities defined by the method in Section 3.5.

The aim of the review was to validate the changes performed in SA model following the system evolutions leading to SE model V4.4. These system evolutions are managed through a light Change Process: 5 Change Reports have been resolved between V4.3 and V4.4. We choose to approach to review by these CRs, i.e. looking the CRs one after another and review all the CLs impacted by the CR scope.

As explained in 4.1.1, the full review necessitated four review sessions performed in real situation and organized as follows:

- Session #1 (12/01/2021, ~2h) : 3 CR reviewed
- Session #2 (15/01/2021, ~2h) : revisions from previous session + 2 CR reviewed
- Session #3 (19/01/2021 am, ~1h) : review of remaining suspect CLs
- Session #4 (19/01/2021 pm, ~15mn) : review of last remaining suspect CLs

In our case, the three pre-requisites defined in Section 3.5 were fulfilled:

- Before each session, the CL set was checked against the coverage and correctness constraints
- The review scope was defined : CR list for the first sessions, then remaining suspects CLs
- The tool provides the capacity to navigate dynamically in the abstracted models, and to display the context information of any CL.

The tables below gives a synthesis of the activities performed during each review session:

	Session #1 (12/01/2021)		Session #2 (15/01/2021)	
	CLset before review	CLset after review	CLset before review	CLset after review
Total number of CL	203	203	203	203
Suspects CLs	106	96	96	65
Validated CLs	95	100	107	137
CLs in revision	2	7	0	1
Updated rationales	9		14	

Table 7: Synthesis of review sessions #1 and #2

	Session #3 (19/01/2021 am)		Session #4 (19/01/2021 pm)	
	CLset before review	CLset after review	CLset before review	CLset after review
Total number of CL	204	204	201	200*
Suspects CLs	52	2	14	0
Validated CLs	150	190	187	200
CLs in revision	2	12	0	0
Updated rationales	18		3	

Table 8: Synthesis of review sessions #3 and #4

**: One unused CL left in the CL set has been deleted directly in session*

We propose here an interpretation of those metrics:

- The state of the CL set has evolved between each session. This can be explained by the following reasons :
  - As explained in 4.1.1, both models have slightly evolved during the review process. In particular, the MBSA model has sometimes been modified to take into account review comments
  - Some CLs have been auto-validated by the safety specialist between the sessions
- After the first two sessions, which focus on system evolutions managed by the CRs, the number of remaining suspects CL is still high (~half the initial number of suspects CLs). This is because the safety specialist did not

performed a complete pre-review and auto-validation of the whole CL set. However, we can see that the sessions #3 and #4 focused on remaining suspects CL allowed to validate an important number of CLs in a reduced period of time. In total, most of the review time has focused on the real system and SE model evolutions.

In our case, the reviewers were the same people involved in editing the models, i.e. the system architect and safety specialists , and both models were well known by each of the reviewers. This may have been a facilitating factor for the review.

As listed in the CL validation criteria in 3.5, the behavior consistency in each CL has been studied and validated. This has been done “informally”, by discussion between the reviewers, and did not rely on a specific behavior consistency method, which is not defined here. However, the validation status reflects this validation: a validated CL means that the reviewers agreed the behavior consistency.

Example of CL79:

In order to illustrate iterations between reviews and actions to solve inconsistency, we detail in Appendix C the case of cl:79, i.e., the CL that has required the most of iterations. Several iterations between system and safety specialists have been needed before converging to an agreement.

This example shows the interest of CLs to capitalize the review results and recommendations. The capacity to visualize CLs at the same time in both SE and SA abstracted model, and to visualize the context of each object (neighbor functions, flows and CLs) is a great help for an efficient review.

It also shows that completing rationales with details during the review is not easy. Comments are not very detailed, and mainly focus on review recommendations. In order to be re-usable in the future for someone not involved in this review, it would require more details and justification, which is time consuming during the review. The details of annotations to be written is related to the capitalization strategy that is out of the scope of our method.

4.6 Conclusions on the validation activities

Through the validation done previously, the seven validation objectives listed in Section 4 are fulfilled, except the corner case showed in Section 4.2.4(b) (but mitigated by the flexibility of our method) and except the methodological discrepancy in Section 4.3 (replaced by the use of CR process instead of using the CLs).

Furthermore, the metrics obtained in Section 4.5 show a rapid convergence to zero inconsistencies while coverage surface concerns the overall of the two models.

The tool (presented in Section 4.1.2) enables to apply the method but remains “cutting edge” so that any lambda user cannot use it without risk of error.

About the specialist workload, SE and SA reviewers spent (qualitatively) less time in review on the whole models with a specific consistency management tool. The time spent by the SA specialist is slightly increased due to CL activities (besides the authoring of its model). This is balanced regarding the avoidance of running future biased analyses based upon inconsistent model.

By applying the method and associated tool on the studied models, the confidence onto the consistency between them has increased but, also, onto the method too.

5 Perspectives

This section aims to point some working directions regarding the return of experience on the tooled validation of method and introspection over the method itself. Working directions can be organized as follow:

- To improve efficiency of method application
- To extend the method with small-scale changes and benefits
- To extend the method with large-scale changes and benefits

It shall be notice that an application of the method in an industrial context is possible but the leader of such activité shall be the enterprise itself helped (if required by IRT).

5.1 Perspectives: Improvement of efficiency of method application

Firstly, a way to improve the method application is to improve the user experience. This can be achieved by improving the tool. The perspective is to optimize the displayable space available against the amount of information to access

- The space displayable is fragmented into the 6 zones while each one has already a huge amount of data in it. Often user has to struggle with the dimensioning of these zones to get the consistent mindset of the situation. Some automatic windowing helpers can help user to reduce these frictions, and this may reduce time of the review.
- As the tooling is only part of a PoC, messages returned to user are succinct but sufficient for people who practiced the method and tool. So more user-friendly messages or navigation to documentation explaining the messages will therefore help the user become familiar with the method.

Secondly, the current tool does not produce any hard copy carrying the state of the current review. Such concrete artefact (like a report) can allow more people to access the work done by SE and SA reviewers without any tool access considerations that currently limit the inconsistency detection by other than SE and SA reviewers.

Lastly, during editing phase, time is spent on the finding of associations to be done between model's artefacts. This time can be reduced if the tool proposes some of them to the user. Such proposal can be done for example, by inference on similarities between models like name of functions/flows or by topologies coupled with already defined CLs. The more CL are created quickly, the more reduced is the remaining associations to be done (with or without proposal of the tool). Furthermore, such help will also benefit in CL repairs when model evolutions break the CLset integrity. As it will be only proposals, user can overpass it to get the association he wants.

5.2 Perspectives: Extension of the method with small-scale changes and benefits

Firstly, to extend the method our proposition is to improve the impact analysis that CR process requires. As CR proposal were used to identify impact on SE artefact and the CLs are linked to them, CLs can be a way to improve the CR impact analysis on SA side.

Secondly, the balance about the limits of the formalization shall be assessed to identify pros and cons. Limits for formalized validation criteria are not enough defined in the current state of the method. A deeper formalization will make reviewers loose flexibility over the review but will focus them on issues required by the formalization only.

Thirdly, as checks (or triggers) are not order, having contextual rules for prioritization them can improve the day-to-day work. That means regarding a given project or development phase of project, prioritizing information can help to find inconsistency.

Fourthly, because convergence of SA model to SE one is composed of small and continuous task, it can be more effective than when composed of a large task in the end. Therefore, an indicator and some associated thresholds can be set up to ease passage between those tasks. This was foreseen (see Figure 4 Item 53 and 54) but the PoC left over this track. Such indicator can warn the SE author that he degrades the previous consistency state done by the SA specialist because of the increasing amount of errors on CLs the SE's changes induced. The knowledge of an indicator that gradually passes threshold may be a hint for him. Therefore, he can publish an intermediary version that may ease the SA specialist's convergence tasks. Besides the configuration and baseline management, the SE's intermediate models help to reduce gradually the SA specialist activities because they scope gradually the corrections on its own model and the CL. It results a progressive alignment of SA models to the SE final one.

Fifthly, the convergence by small task fosters a kind of “parallelization process”. Because, the amount of convergence tasks in SA specialist’s backlog determine the delay from the SE’s specialist authoring situation. This contrasts to the “serial process” where SA specialist has to wait SE specialist’s model development process reached its end before he starts to work on its model and associated CL.

5.3 Perspectives: Extension of the method with large-scale changes and benefits

Firstly, The defined method is applied on functional architecture only. Applying the same method on equipment and physical point of view would open perspectives. In particular, the physical point of view is compulsory to perform safety analyses.

Secondly, another point of adaptation of the current method to some considerations may be taken into account. Despite the front scene of CL set (that contains the end user definitions), the backstage is different as the CL definitions define a CL model whose one implementation can be the SE model (if we consider the SE part of the definitions) or another implementation can be the SA model one (if we consider the SA part of the definitions). Exploitation of this abstracted model may be useful to rationalize SE or SA models by pointing useless decomposition level (for example, induced by the successive modelling activities done along the life of the model itself, etc.) and foster to reduce the modeling gaps between models.

Thirdly, to extend the method to behavioral considerations, it is also possible to consider the values that flows can take. In general, SA flows have enumerated values while SE flows have numerical values. If it were possible to define equivalence between SA values and SE ranges, it would help to find behavior inconsistency. The Figure 39 represents two SE’s flows “CL flow-linked” to a single SA flow. But this CL flow can be enriched with the matching of partition of their respective domain, i.e. cartesian cross product of numerical domains segmented into distinct parts for the SE while a single partition is done for the SA side.

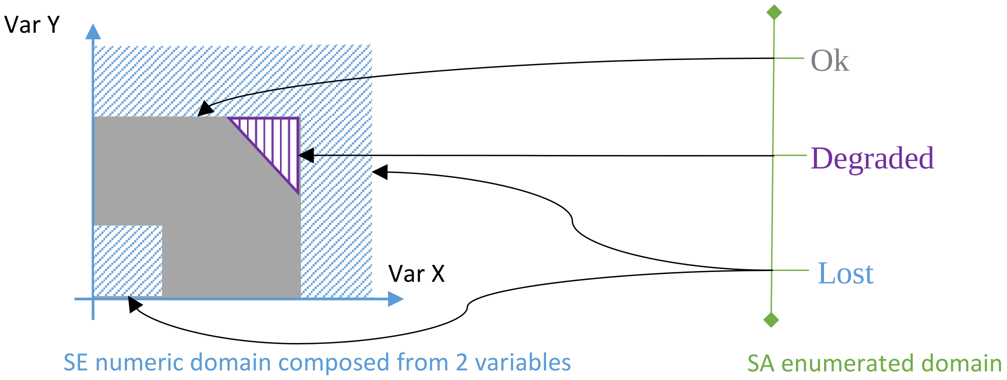


Figure 39: Matching of SA’s partitions domain against SE’s partition domain

Fourthly, the severity of the intra domain correctness rule (see (b)) leads to the situation described in Section 4.2.4(a). Releasing this constraint will had flexibility but will have impact on inconsistencies that cannot be detected. So an assessment on such a topic may be useful to determine the gain for the user against the loss of confidence.

Fifthly, the defined method manages only functional architecture. In the previous section, we have considered to apply it to the physical architecture. Nevertheless, this would not solve the issue of consistency of allocations between each architecture layer (functional to physical one). The method could be extended to manage this issue and cross check allocations done in each model.

Sixthly, the assertions in the SA model are not exploited yet to identify the dependencies between outputs and inputs of a function. These dependencies can be leveled-up to dependencies between CL flows that interfaces a CL function. Such dependencies between CL flows can be exposed to SE reviewers which can (by its knowledge of its own functions under the CL and the CL flows association done with its own flow) confirm or inform the assumption of the SA specialist on its assertions.

Lastly, all the extensions until now are based upon reasoning on the definition and/or static information available but nothing about dynamic behavioral. Therefore, a working direction can be an extension to such a topic by assessing cross checks between, SE specialist’s expected behaviors against the SA simulation capabilities. Reciprocally, SA specialist may wonder about behavior induced by its own modeling and can suggest a validation of such behaviors by the SE specialist.

Appendix A: Consistency link detailed properties and relations

Generic consistency link

On top of properties mentioned in Section 3.1.2, a consistency link shall also have the following properties:

1) Author (MBSA specialist)
2) Categories or filtering marks

Even if version management is not addressed yet, we already know that the following properties will be necessary:

3) Date of last change of CL
4) Version of the CL
5) MBSE model version and MBSA model version addressed by the CL
6) MBSE model version and MBSA model version addressed by the last validation of the CL

Figure 40 describes the generic concept of consistency link using EMF metamodel representation.

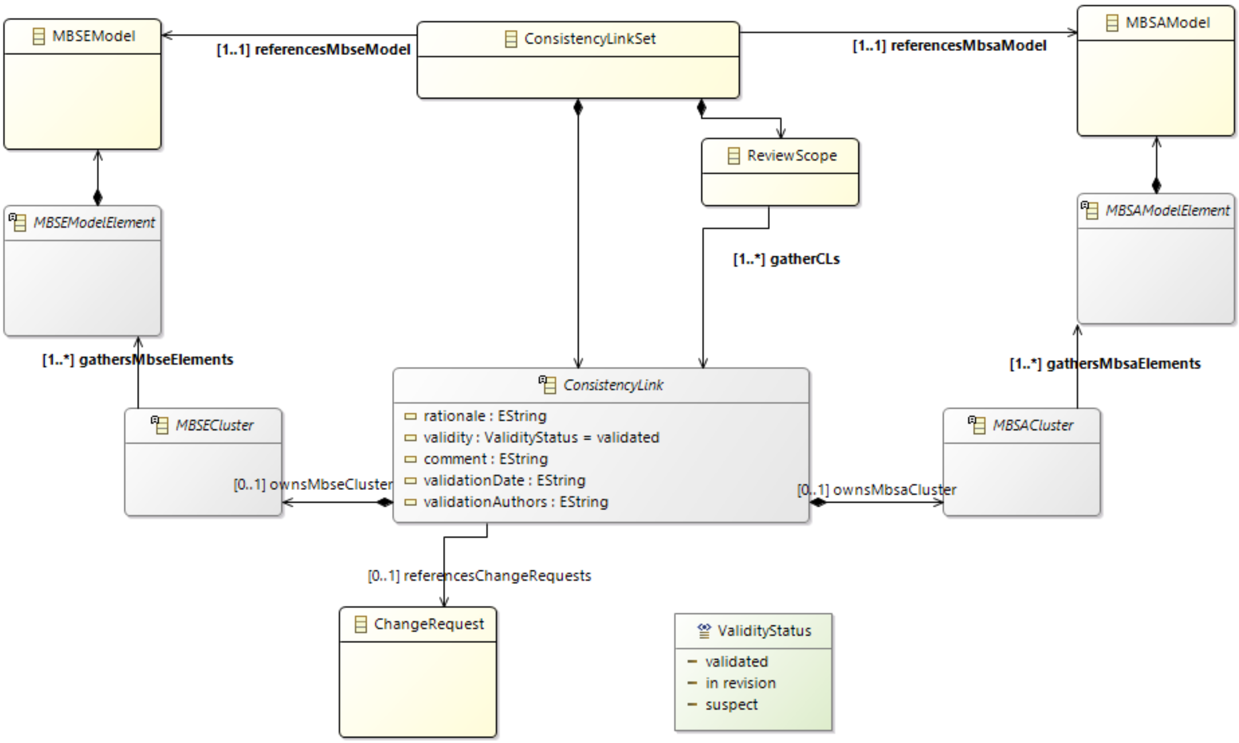


Figure 40: Metamodel of consistency link

Consistency link for functions (CLF)

Figure 41 describes the consistency link for functions from Section 3.2.2. Let note that in all diagrams of the appendix, the blue relations are derived (specialized) from a previously defined relation (mainly defined in Figure 40Figure 44).

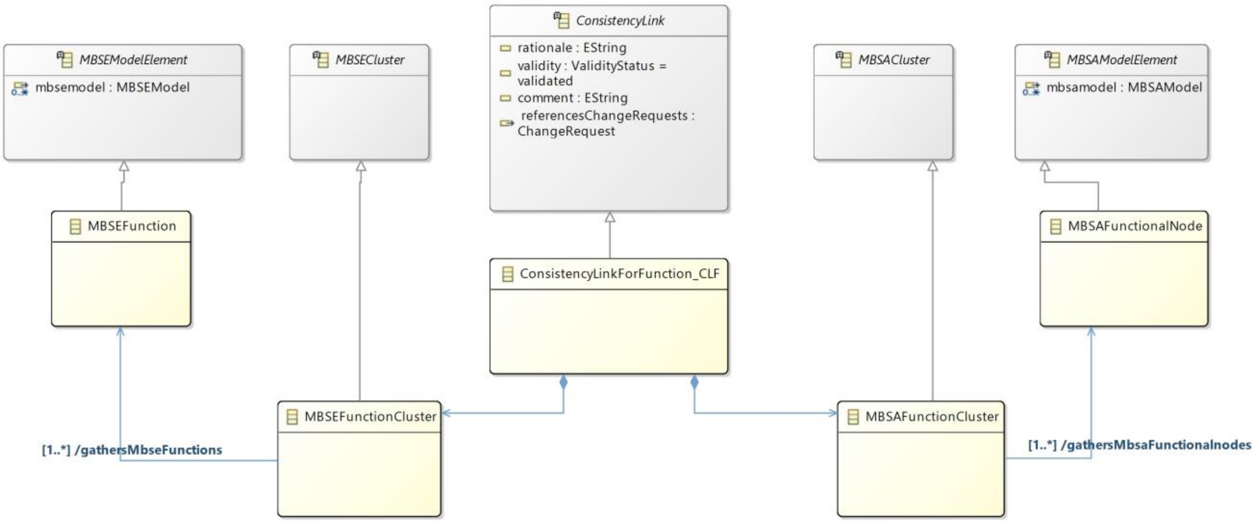


Figure 41: Metamodel of consistency link for functions

Consistency link for functional flows (CLfl)

Figure 42 describes the consistency links for functional flows from Section 3.2.3.

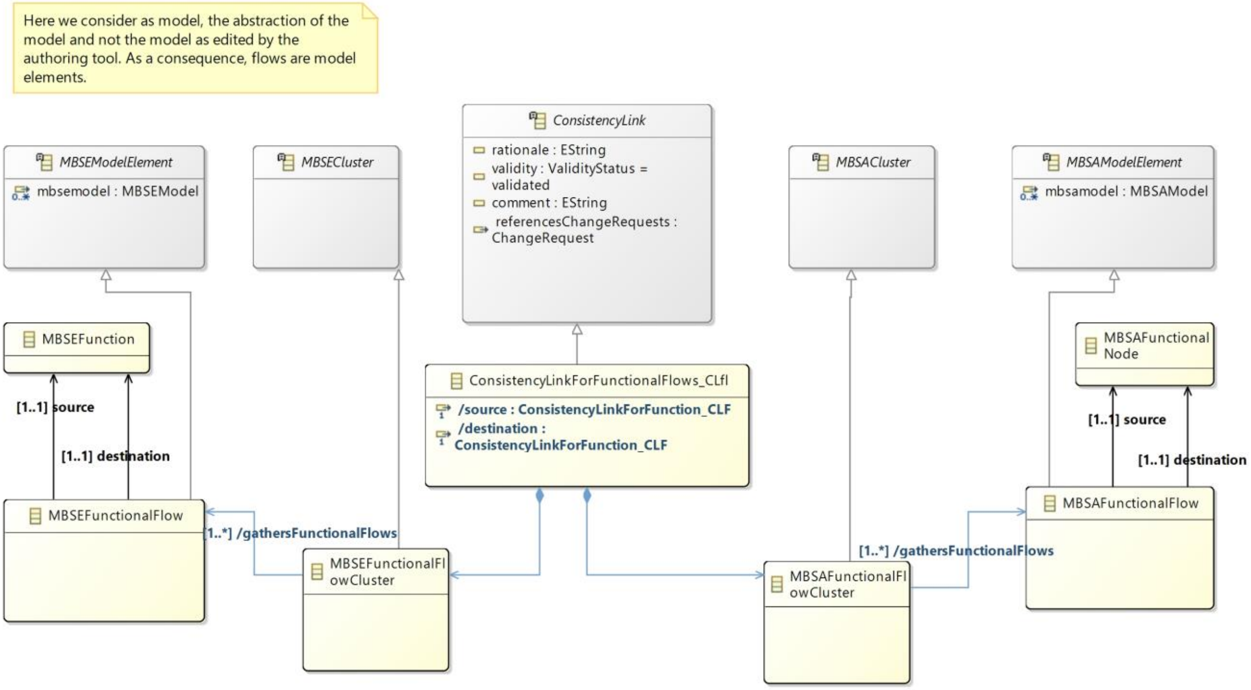

Here we consider as model, the abstraction of the model and not the model as edited by the authoring tool. As a consequence, flows are model elements.
ConsistencyLink
rationale : EString
validity : ValidityStatus = validated
comment : EString
referencesChangeRequests : ChangeRequest
MBSEModelElement
mbsemodel : MBSEModel
MBSECluster
MBSACluster
MBSAModelElement
mbsamodel : MBSAModel
MBSEFunction
MBSAFunctional Node
ConsistencyLinkForFunctionalFlows_CLfl
/source : ConsistencyLinkForFunction_CLF
/destination : ConsistencyLinkForFunction_CLF
[1..1] source
[1..1] destination
MBSEFunctionalFlow
MBSAFunctionalFlow
[1..*] /gathersFunctionalFlows
MBSEFunctionalFlowCluster
MBSAFunctionalFlowCluster

Figure 42: Metamodel of consistency link for functional flows

Relations between CLF and CLfl

The following figure explains how CLF and CLfl are related, notably thanks to the rule defined in Section 3.2.3(c).

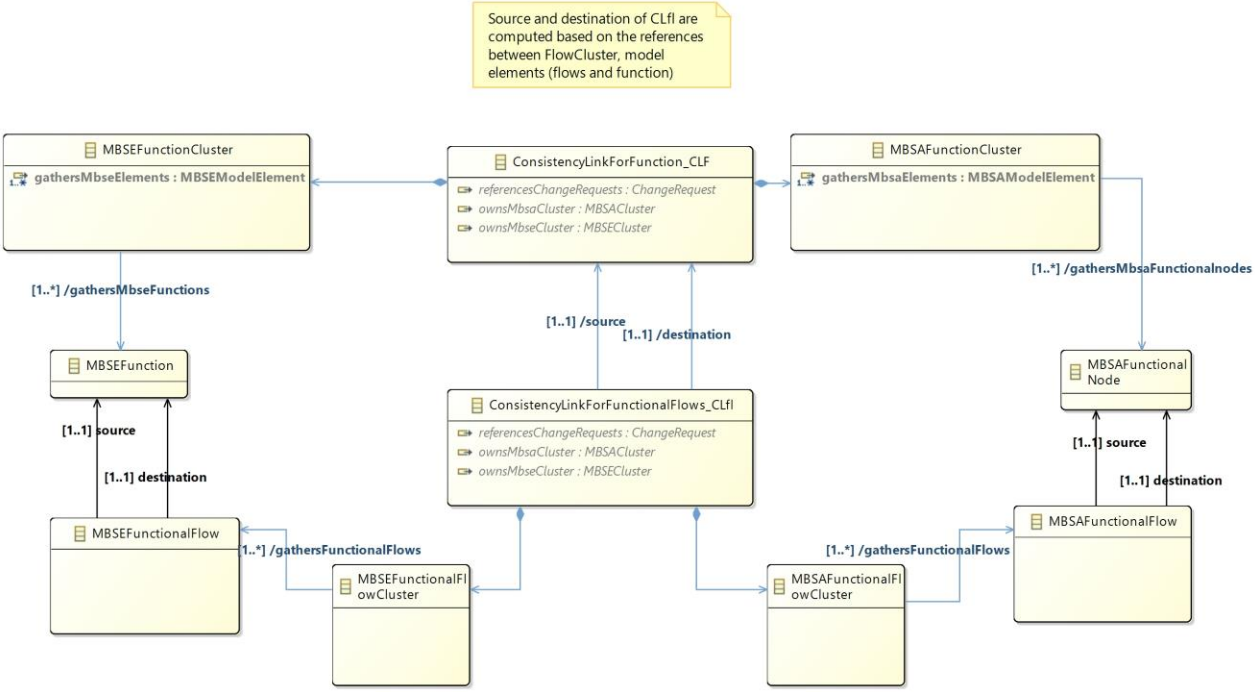

Source and destination of CLfl are computed based on the references between FlowCluster, model elements (flows and function)
MBSEFunctionCluster
gathersMbseElements : MBSEModelElement
ConsistencyLinkForFunction_CLF
referencesChangeRequests : ChangeRequest
ownsMbsaCluster : MBSACluster
ownsMbseCluster : MBSECluster
MBSAFunctionCluster
gathersMbsaElements : MBSAModelElement
[1..*] /gathersMbseFunctions
[1..*] /gathersMbsaFunctionalnodes
[1..1] /source
[1..1] /destination
MBSEFunction
MBSAFunctional Node
ConsistencyLinkForFunctionalFlows_CLfl
referencesChangeRequests : ChangeRequest
ownsMbsaCluster : MBSACluster
ownsMbseCluster : MBSECluster
[1..1] source
[1..1] destination
[1..1] source
[1..1] destination
MBSEFunctionalFlow
[1..*] /gathersFunctionalFlows
MBSEFunctionalFlowCluster
[1..*] /gathersFunctionalFlows
MBSAFunctionalFlowCluster
MBSAFunctionalFlow

Figure 43: Overview of relations between consistency links for functions and consistency links for flows

Appendix B: Abstraction of functional architecture

Example of abstraction

This section shows the same system modeled with different methods and tools. When any of the models shown in Figure 45 to Figure 48 is abstracted, the resulting abstraction is the same, illustrated in Figure 44.

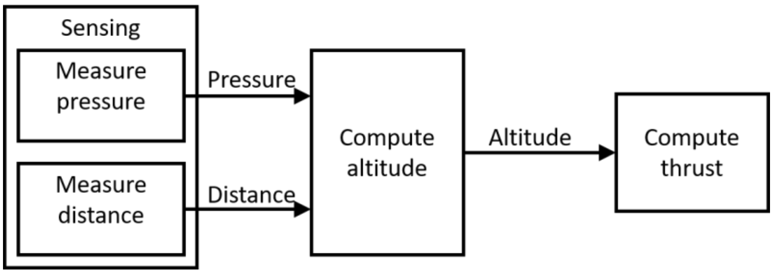


Figure 44: Example of abstracted functional architecture

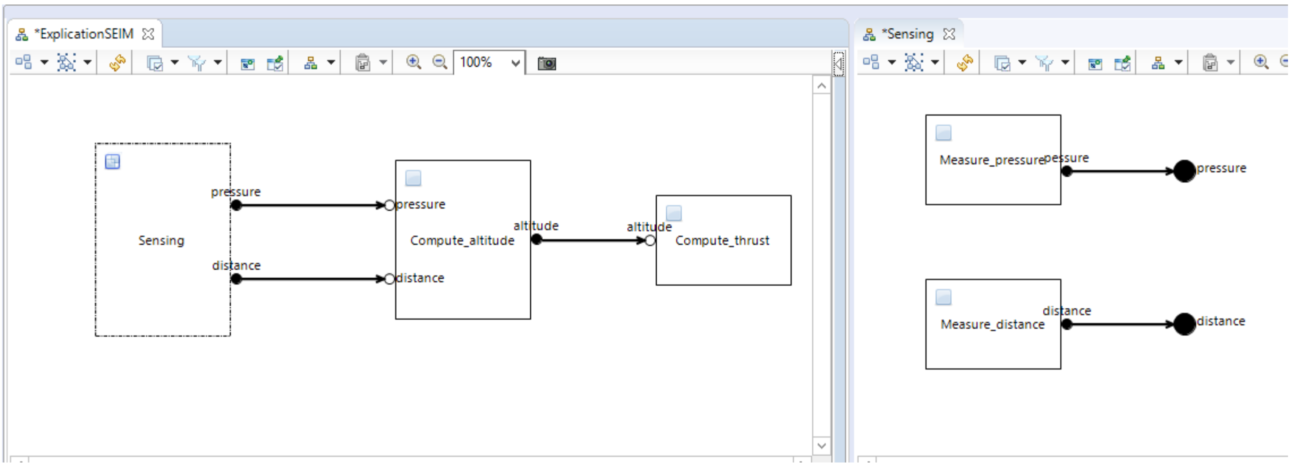


Figure 45: Example modeled with SimfiaNeo using Bricks

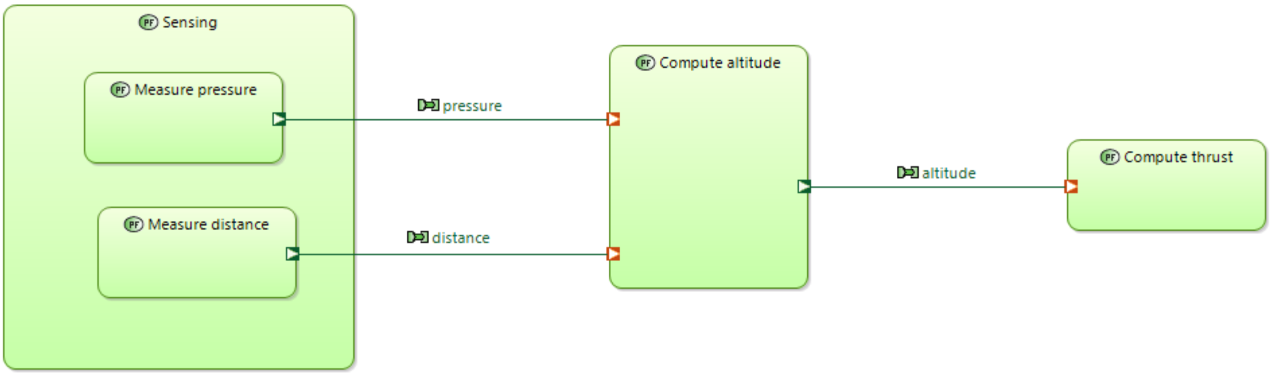


Figure 46: Example modeled with Capella

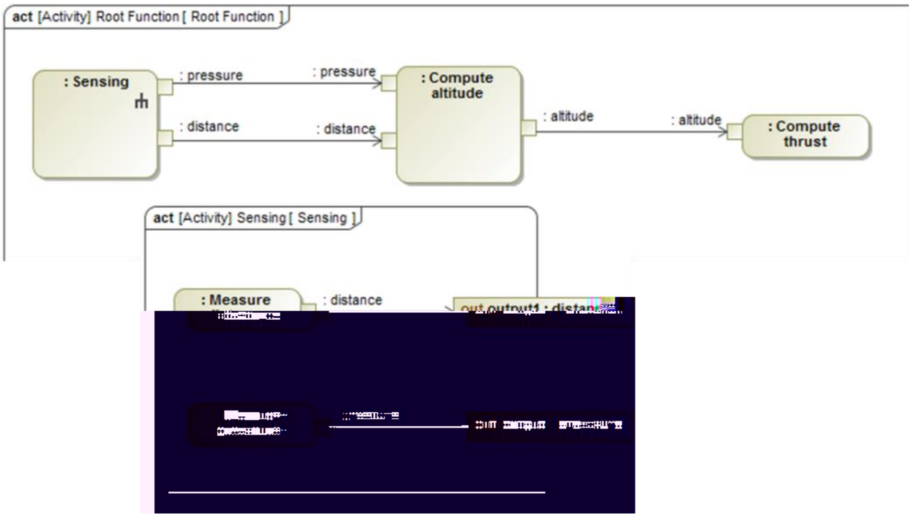


Figure 47: Example modeled with Cameo using SysML activities

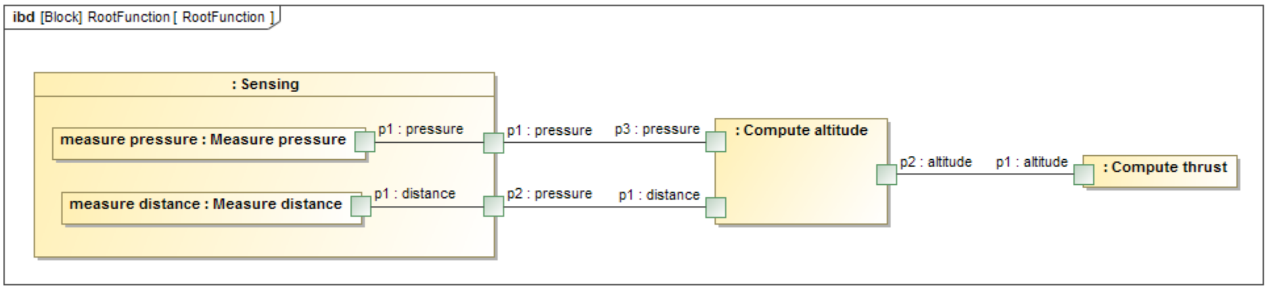


Figure 48: Example modeled with Cameo using SysML blocks

Tooling the abstraction

Automatizing the abstraction for Capella and both Cameo modeling methods of Figure 47 and Figure 48 has been done during MOISE project. This abstraction is tooled by a software component called a “connector” to TeePee, a proof-of-concept platform to manage heterogeneity and, more precisely to the functional architecture viewpoint of TeePee (see LIV- S -014- S4.22-42-511-V1 for more information).

S2C safety modeling needs require a MBSA modeling tool and its connector to TeePee. Both SimfiaNeo and its TeePee connector have been made available to S2C. Table 9 gives an overview of the mapping between abstracted functional architecture (functional flow viewpoint) and modeled functional architecture (with SimfiaNeo). Following the same mapping, this abstraction and associated tooling could be applied to other MBSE and MBSA authoring tools.

Model kind Viewpoint element	MBSE model in Capella	MBSA model in SimfiaNeo
Function	Actor Function and Function	BrickInstance
FunctionPort	Function Port	ConnectorInstance
Functionsports relation (Has Function Ports)	As a “subpart” of the Function	ownedConnectorInstances
FunctionalFlow	Functional Exchange	ownedConnectorLinks and Link
sourceFunctionPort relation (Comes from Source Function Port)	Source Function Output Port (automatically computed)	sourceConnectorInstance
targetFunctionPort relation (Goes to Target Function Port)	Target Function Input Port (automatically computed)	targetConnectorInstance

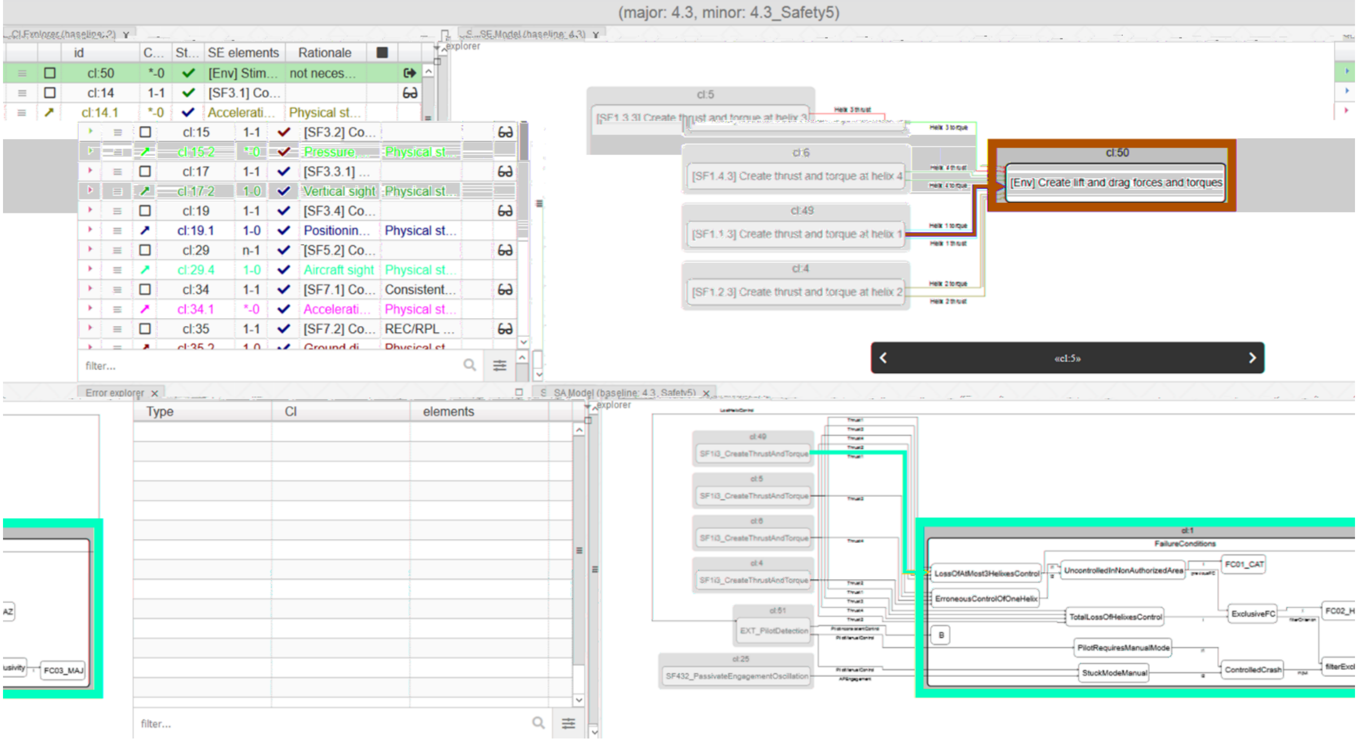
Table 9: Mapping between functional flow viewpoint, MBSE, and MBSA authoring tools

Appendix C: Consistency review example on consistency link cl:79

The CL79 has been created after a SE model change between V4.3 and V4.4: the function SF1.5 has been added, to represent the function of the drone structure which consists in "gathering" the thrust and torque of each propeller into a global drone thrust and torque which interact with the environment.

Figure 49 shows the modelling in both models in system version V4.3 (before the CR is taken into account):

- In SE model, the thrust and torque of each propeller (flow in green in the figure below) interact directly with the environment (CL50 in green)
- In SA model, the environment is not represented. The thrust and torque of each propeller (flow in orange in the figure below) is "observed", as contributors to the Failure Conditions (CL1 in orange below)



(major: 4.3, minor: 4.3_Safety5)

CI Explorer (baseline: 2)

id	C...	St...	SE elements	Rationale
cl:50	*-0	✓	[Env] Stim...	not neces...
cl:14	1-1	✓	[SF3.1] Co...	
cl:14.1	*-0	✓	Accelerati...	Physical st...
cl:15	1-1	✓	[SF3.2] Co...	
cl:15.2	*-0	✓	Pressure,...	Physical st...
cl:17	1-1	✓	[SF3.3.1] ...	
cl:17.2	1-0	✓	Vertical sight	Physical st...
cl:19	1-1	✓	[SF3.4] Co...	
cl:19.1	1-0	✓	Positionin...	Physical st...
cl:29	n-1	✓	[SF5.2] Co...	
cl:29.4	1-0	✓	Aircraft sight	Physical st...
cl:34	1-1	✓	[SF7.1] Co...	Consistent...
cl:34.1	*-0	✓	Accelerati...	Physical st...
cl:35	1-1	✓	[SF7.2] Co...	REC/RPL ...
cl:35.2	1-0	✓	Ground di...	Physical st...

filter...

Error explorer

Type	Cl	elements

filter...

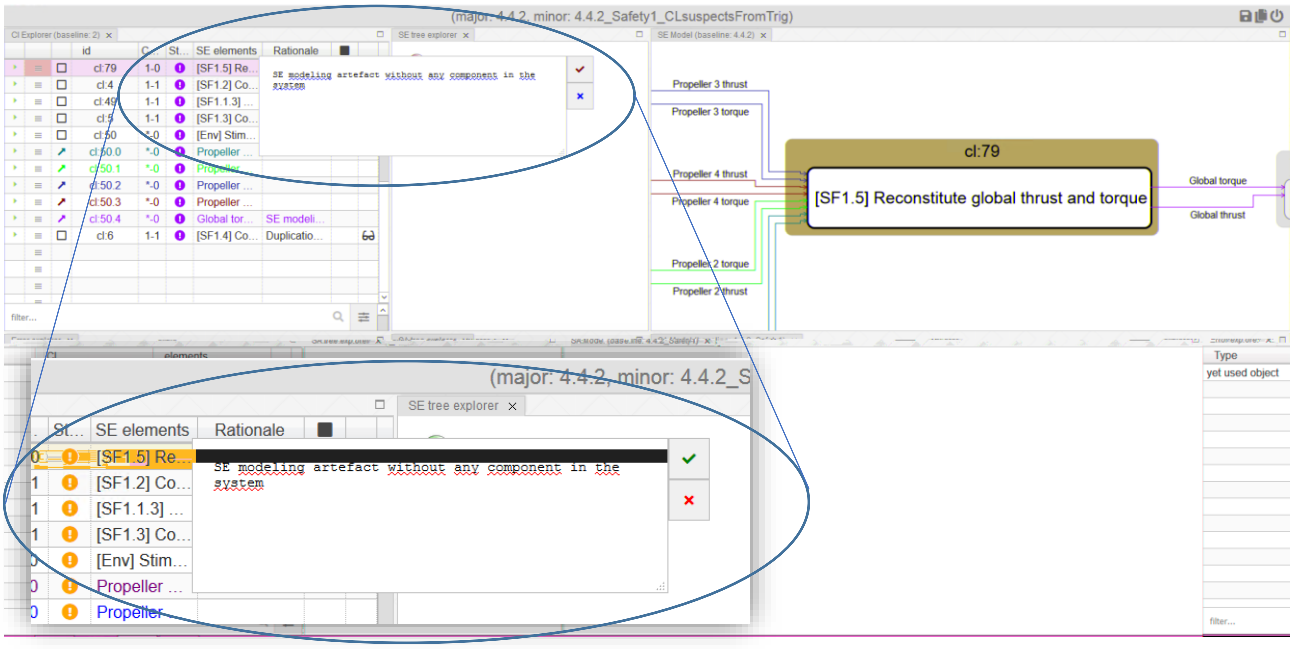
SE Model (baseline: 4.3)
explorer
cl:5
[SF1.3.3] Create thrust and torque at helix 3
cl:6
[SF1.4.3] Create thrust and torque at helix 4
cl:49
[SF1.1.3] Create thrust and torque at helix 1
cl:4
[SF1.2.3] Create thrust and torque at helix 2
cl:50
[Env] Create lift and drag forces and torques
Helix 3 thrust
Helix 3 torque
Helix 4 thrust
Helix 4 torque
Helix 1 torque
Helix 1 thrust
Helix 2 torque
Helix 2 thrust
«cl:5»
SA Model (baseline: 4.3_Safety5)
explorer
cl:49
SF1i3_CreateThrustAndTorque
cl:5
SF1i3_CreateThrustAndTorque
cl:6
SF1i3_CreateThrustAndTorque
cl:4
SF1i3_CreateThrustAndTorque
cl:51
EXT_PilotDetection
cl:25
SF432_PassiveEngagementOscillation
cl:1
FailureConditions
LossOfAtMost3HelixesControl
ErroneousControlOfOneHelix
UncontrolledInNonAuthorizedArea
FC01_CAT
TotalLossOfHelixesControl
ExclusiveFC
FC02_H
PilotRequiresManualMode
StuckModeManual
ControlledCrash
filterExcl
B
AZ
FC03_MAJ

Figure 49 : Drone interaction with environment in V4.3

Figure 50 shows the modelling after the CR is taken into account, and before the review session #2:

- In SE model, SF1.5 has been added between SF1.1/2/3/4 and the environment
- In SA model, the safety specialist considered at first (but wrongly) that this function was only a modelling artefact in SE model and that taking it into account in SA model was not necessary.

The cl:79 has been created to cover SF1.5, with the cardinality "1-0" (meaning it is associated to a SE function only, see details in 3.2.2(c)). The rational explains the safety specialist understanding. It initial status is "Suspect".

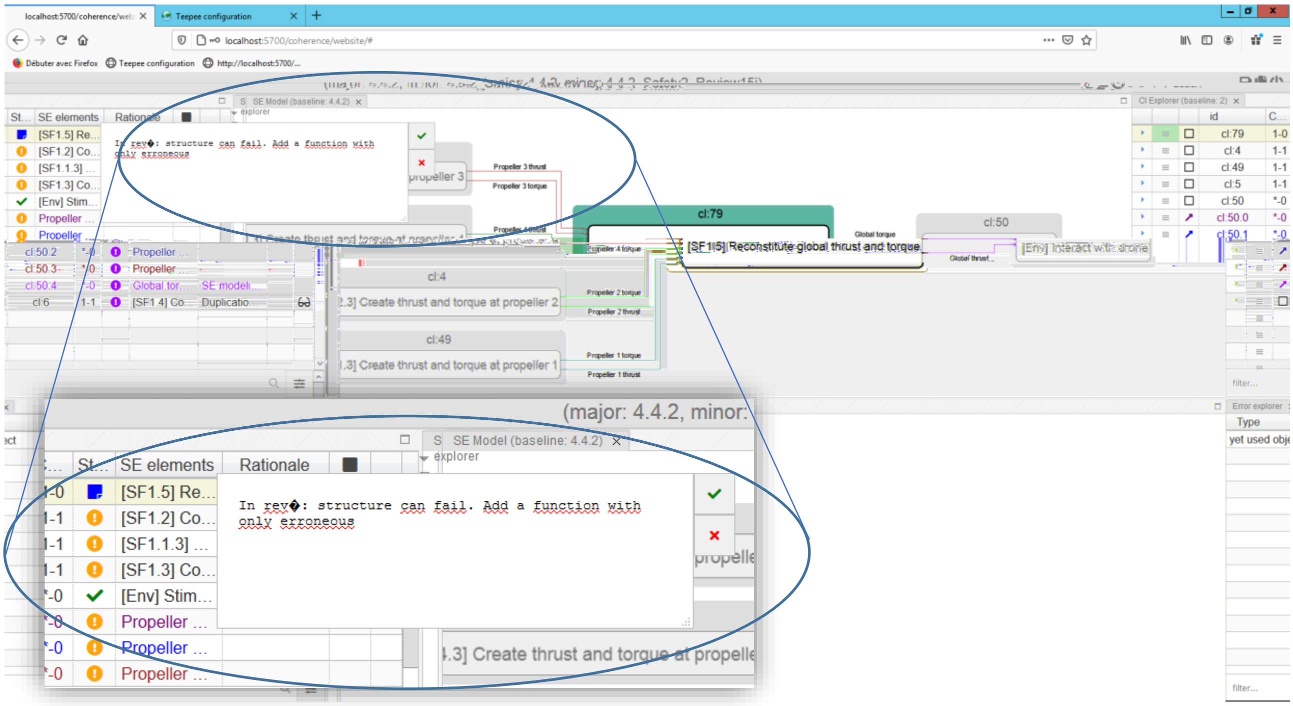


id	C...	St...	SE elements	Rationale
cl:79	1-0		[SF1.5] Re...	
cl:4	1-1		[SF1.2] Co...	
cl:49	1-1		[SF1.1.3] ...	
cl:5	1-1		[SF1.3] Co...	
cl:50	*-0		[Env] Stim...	
cl:50.0	*-0		Propeller ...	
cl:50.1	*-0		Propeller ...	
cl:50.2	*-0		Propeller ...	
cl:50.3	*-0		Propeller ...	
cl:50.4	*-0		Global tor...	SE modeli...
cl:6	1-1		[SF1.4] Co...	Duplicatio...

SE modeling artefact without any component in the system

Figure 50: Drone interaction with environment before review session #2

During the review session #2, the cl:79 has been reviewed a first time. It was confirmed that SF1.5 is a impacting function of the system, allocated to the drone structure, and that failure modes can be identified (structure failure leading to the loss of one propeller). The CL status is set to “In revision” and a comment is added in the rationale. This can be seen on Figure 51 below

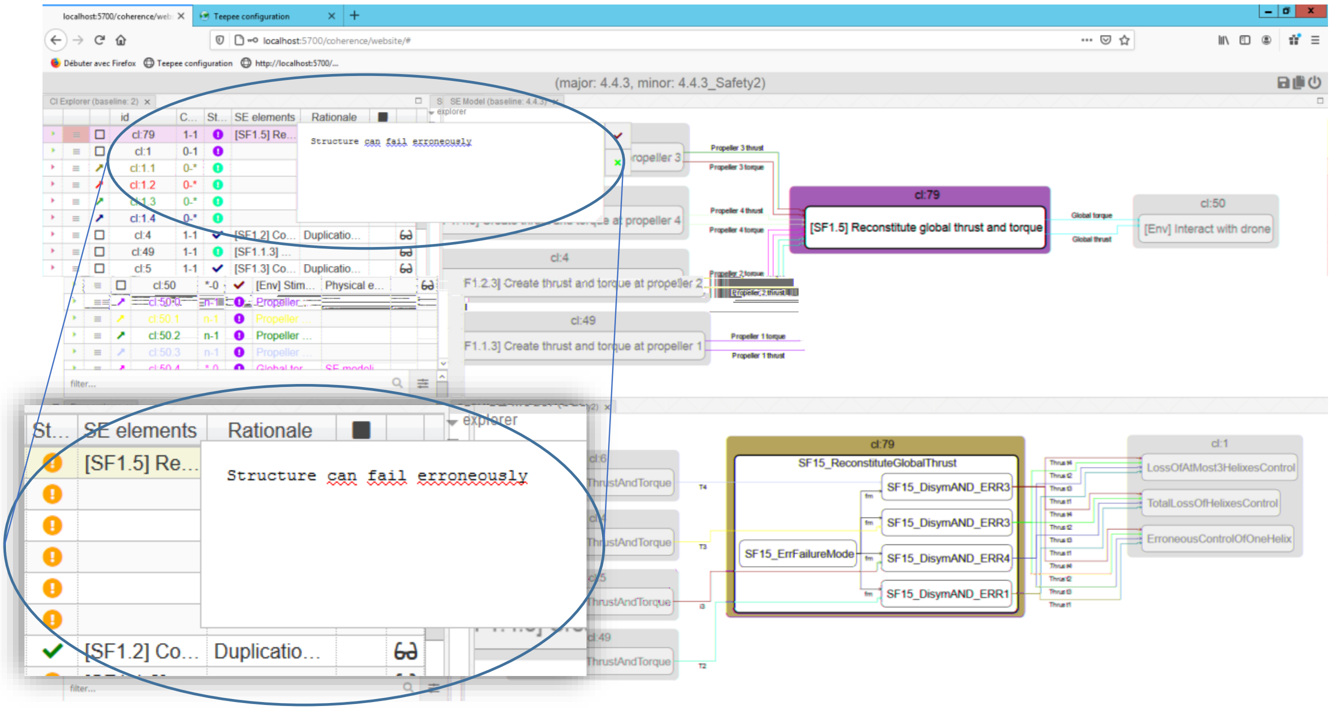

localhost:5700/coherence/web
Teepee configuration
localhost:5700/coherence/website/#
Débuter avec Firefox
Teepee configuration
http://localhost:5700/...
SE Model (baseline: 4.4.2)
CI Explorer (baseline: 2)
Propeller 3 thrust
Propeller 3 torque
cl:79
[SF1.5] Reconstitute global thrust and torque
cl:50
[Env] Interact with drone
cl:4
.2.3] Create thrust and torque at propeller 2
cl:49
.1.3] Create thrust and torque at propeller 1
Global torque
Global thrust
Error explorer
Type
yet used obj
filter...

St...	SE elements	Rationale
	[SF1.5] Re...	In rev�: structure can fail. Add a function with only erroneous
	[SF1.2] Co...	
	[SF1.1.3] ...	
	[SF1.3] Co...	
✓	[Env] Stim...	
	Propeller ...	
	Propeller ...	
	Propeller ...	

Figure 51: Review session #2 results on CL79

Between session #2 and session #3, the SA model is updated: SF1.5 is modelled in the SA model. The cl:79 is updated and covers now some elements in the SA model. Its cardinality is now 1-1, the status is set again to « Suspect » because it has changed and the rationale has been completed with justification on the modelling choice.

The content before the session #3 is represented below on Figure 52.

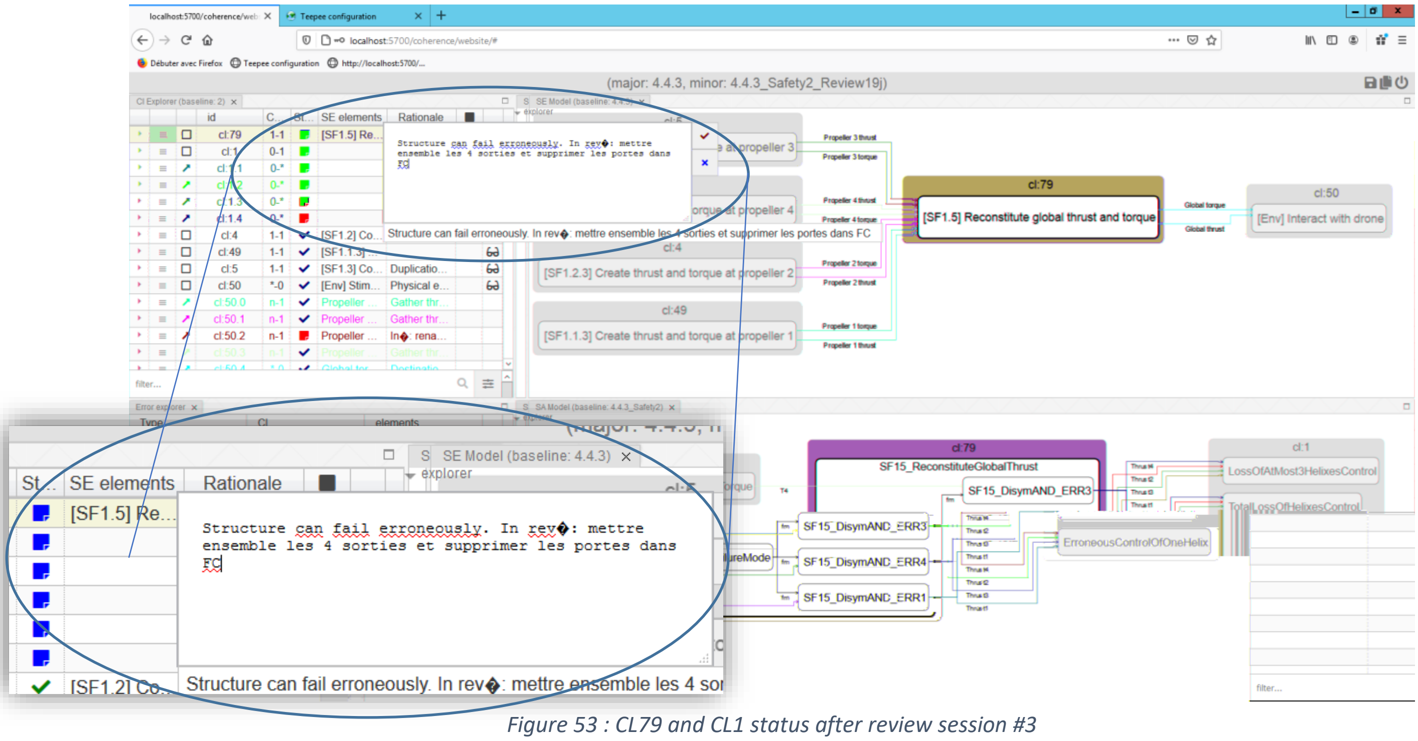


(major: 4.4.3, minor: 4.4.3_Safety2)

St…	SE elements	Rationale
!	[SF1.5] Re…	Structure can fail erroneously
✓	[SF1.2] Co…	Duplicatio…

Figure 52 : CL79 updated before review session #3

During the review session #3, discussions make appear that SF1.5 enable to simplify the FC observer. Indeed, the combination of failure modes from SF1.1 to SF1.4 is now made in SF1.5 and then could be deleted from the FC observer (scope of cl:1). As a result, both cl:79 and cl:1 statuses are set to “In revision”.



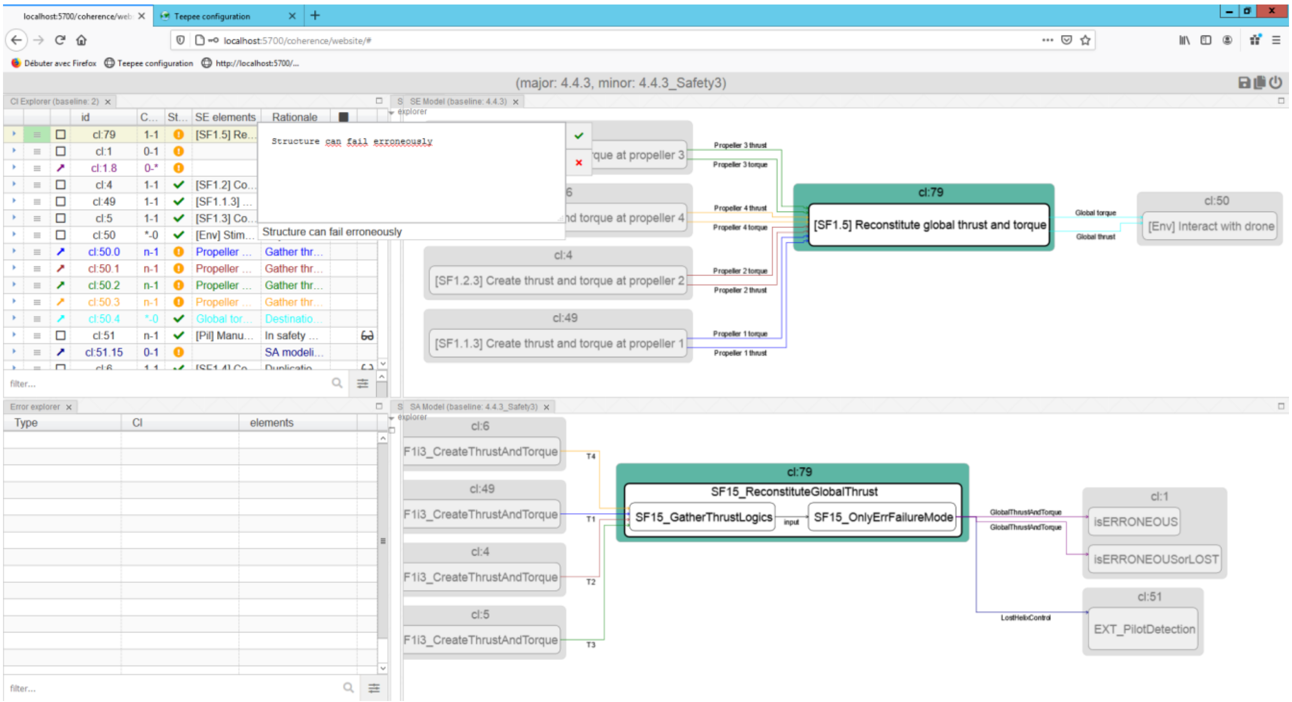
(major: 4.4.3, minor: 4.4.3_Safety2_Review19j)

St…	SE elements	Rationale
🟦	[SF1.5] Re…	Structure can fail erroneously. In rev�: mettre ensemble les 4 sorties et supprimer les portes dans FC
✓	[SF1.2] Co…	Structure can fail erroneously. In rev�: mettre ensemble les 4 sorties et supprimer les portes dans FC

localhost:5700/coherence/web
Teepee configuration
CI Explorer (baseline: 2)
id
cl:79
cl:1
cl:1.1
cl:1.2
cl:1.3
cl:1.4
cl:4
cl:49
cl:5
cl:50
cl:50.0
cl:50.1
cl:50.2
[SF1.2] Co…
[SF1.3] Co…
[Env] Stim…
Duplicatio…
Physical e…
Propeller …
Gather thr…
filter…
Error explorer
SE Model (baseline: 4.4.3)
SA Model (baseline: 4.4.3_Safety2)
explorer
cl:79
[SF1.5] Reconstitute global thrust and torque
cl:50
[Env] Interact with drone
cl:4
[SF1.2.3] Create thrust and torque at propeller 2
cl:49
[SF1.1.3] Create thrust and torque at propeller 1
SF15_ReconstituteGlobalThrust
SF15_DisymAND_ERR3
SF15_DisymAND_ERR4
SF15_DisymAND_ERR1
cl:1
LossOfAtMost3HelixesControl
TotalLossOfHelixesControl
ErroneousControlOfOneHelix

Figure 53 : CL79 and CL1 status after review session #3

The safety specialist then updated again the SA model before review session #4.



localhost:5700/coherence/website/#

(major: 4.4.3, minor: 4.4.3_Safety3)

CI Explorer (baseline: 2)

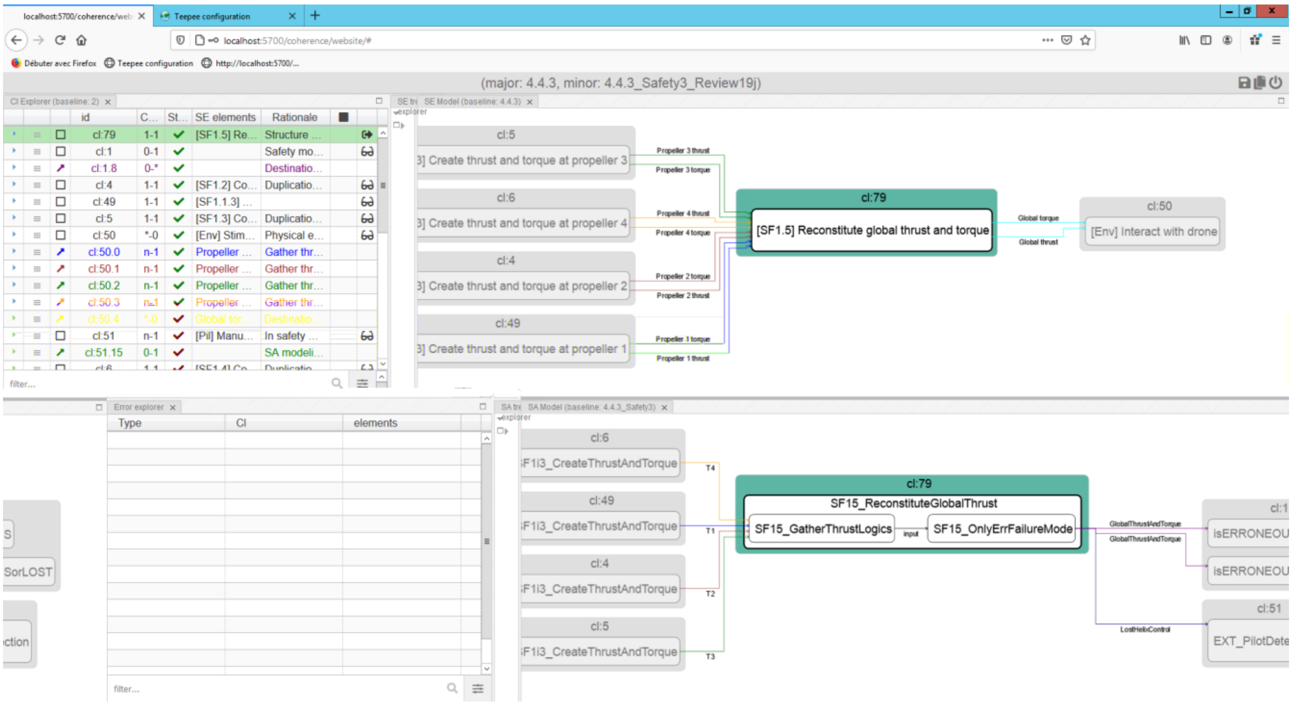
id	C...	St...	SE elements	Rationale
cl:79	1-1	!	[SF1.5] Re...	
cl:1	0-1	!		
cl:1.8	0-*	!		
cl:4	1-1	✓	[SF1.2] Co...	
cl:49	1-1	✓	[SF1.1.3] ...	
cl:5	1-1	✓	[SF1.3] Co...	
cl:50	*-0	✓	[Env] Stim...	
cl:50.0	n-1	!	Propeller ...	Gather thr...
cl:50.1	n-1	!	Propeller ...	Gather thr...
cl:50.2	n-1	!	Propeller ...	Gather thr...
cl:50.3	n-1	!	Propeller ...	Gather thr...
cl:50.4	*-0	✓	Global tor...	Destinatio...
cl:51	n-1	✓	[Pil] Manu...	In safety ...
cl:51.15	0-1	!		SA modeli...

Structure can fail erroneously

Structure can fail erroneously

Figure 54 : CL79 before review session #4

The last review session confirmed the agreement on this modelling. The CL79 is finally set to “Valid”.



localhost:5700/coherence/website/#

(major: 4.4.3, minor: 4.4.3_Safety3_Review19j)

CI Explorer (baseline: 2)

id	C...	St...	SE elements	Rationale
cl:79	1-1	✓	[SF1.5] Re...	Structure ...
cl:1	0-1	✓		Safety mo...
cl:1.8	0-*	✓		Destinatio...
cl:4	1-1	✓	[SF1.2] Co...	Duplicatio...
cl:49	1-1	✓	[SF1.1.3] ...	
cl:5	1-1	✓	[SF1.3] Co...	Duplicatio...
cl:50	*-0	✓	[Env] Stim...	Physical e...
cl:50.0	n-1	✓	Propeller ...	Gather thr...
cl:50.1	n-1	✓	Propeller ...	Gather thr...
cl:50.2	n-1	✓	Propeller ...	Gather thr...
cl:50.3	n-1	✓	Propeller ...	Gather thr...
cl:50.4	*-0	✓	Global tor...	Destinatio...
cl:51	n-1	✓	[Pil] Manu...	In safety ...
cl:51.15	0-1	✓		SA modeli...

Figure 55: CL79 set to valid after review session #4

End of document