Structural consistency of MBSE and MBSA models using Consistency Links

Romaric Demachy, Sébastien Guilmeau

Short paper - Structural consistency of MBSE and MBSA models using Consistency Links

Romaric DEMACHY
Sébastien GUILMEAU
romaric.demachy@irt-saintexupery.com
sebastien.guilmeau@irt-saintexupery.com
IRT Saint Exupéry
Toulouse, France, France

ABSTRACT

The systems designed in industrial fields such as aeronautics or aerospace are more and more complex. In order to handle this complexity as well as the increasing need of digital continuity, model-based solutions are more and more introduced for system design (*MBSE*). In this picture, in order to ensure the consistency between the system design and the safety analysis, and thus increase the confidence in safety analyses results, our proposal is to ensure the consistency between *MBSE* and *MBSA* (Model Based Safety Analysis), which represent different views of the same system. To do so, we define Consistency Links (*CL*) that make a bridge between the structural items of each model. Associated with dedicated rules, that can be systematically checked, the *CL* can be used to drive the *cross-review* of models done by system engineers (*SE*) and safety specialists (*SA*) to increases detection of inconsistencies between models.

The work presented here is part of the S2C project and involves industrial partners from the space and the aeronautical industry. It is led jointly by IRT Saint Exupéry and IRT SystemX.

Keywords : Model-Based System Engineering, Model-Based Safety Assessment, digital continuity

1 PROBLEM STATEMENT

When designing a system which must comply with safety requirements, the process shall ensure, as early as possible and all along the design phase, that the system architecture is compatible with them.

The safety assessment is performed by *SA* teams and imply the usage of safety methods and tools. Methods such as Fault Tree Analysis have been used for decades, but *MBSA* approach has emerged and is now recognized as an acceptable mean of compliance by aeronautic regulation authorities. Indeed, this approach is identified in ARP4761A, which gives guidelines and methods of performing the safety assessment for certification of civil aircraft, and will be soon published by SAE International. Ensuring the correctness of *SA* models with regards to the system design is mandatory for the relevance of the safety assessment. In current practices, this relies on exchanges and *cross-review* between the *SE* team and the *SA* team.

As the system design process also progressively relies on *MBSE* approaches, there is an opportunity to ease this review process by taking advantage of the provided model's formalism. The problem becomes : how to ensure a better consistency between the *MBSE* model and the *MBSA* model ? As these analyses are currently performed on specific tools dedicated to either *MBSE* or *MBSA* analysis, we choose here to focus on the case where two different (i.e. each model has its own objectives and modeling choices) and heterogenous (i.e. each model uses a specific modeling language) models are used. In this paper, the problem is narrowed taking into account the following constraints :

- Constraint A (CA): For *MBSE* models: only architecture description models (Capella, SysML, ...) and exclude specialised models (digital mock-up, electrical wiring models,etc...)
- Constraint B (CB): For *MBSA* models: we consider failure propagation models (Altarica,...)
- Constraint C (CC): Only the structural consistency is covered in this paper, the consistency of the behavior being a more difficult issue

The benefits of using such models to ease the review between system and safety experts in the aeronautical context has been discussed in [1]. The problem of synchronisation of architecture models and safety models has been adressed in [2]. This thesis is based on the analysis that the information exchanged between these assets are informal. In [3] and [4] a processus for the synchronisation of *MBSE* and *MBSA* models has been proposed, consisting in the projection onto a dedicated language called S2ML (System Structure Modeling Language). [5] proposes an approach consisting in a digital collaborative space based on the federation of modeling languages into a common ontology.

The work presented here consists in a projection into a pivot meta-model in order to evaluate the structural consistency. The proposed method is implemented it in a tool and experimented on a representative case study.

Section 2 describes the principles of the method and details the consistency link concept. Section 3 gives details on how the method has been implemented in a tool for a Proof of Concept. Section 4 shows how the method and its implementation have been experimented on a representative case study, and what are the qualitative gains that have been identified. Section 5 lists the perspectives for future activities.

2 CONSISTENCY LINKS METHOD

To ease the consistency review, the proposed method consists in defining consistency links (*CL*) between groups of artifacts of each model with the following semantic : "The *MBSE* model element(s) and the *MBSA* model element(s) linked together represent the same object". Then, we can decompose the review to address only small and well defined perimeters at a time.

A *CL* carries, also, the consistency validation by reviewers. This is made concrete by the elements associated to the *CL* : a rationale that captures justification and assumptions, a validation status, meta-data such as the the review date and authors. The meta-model of the *CL* object and associated concept is represented on the figure 1 below.

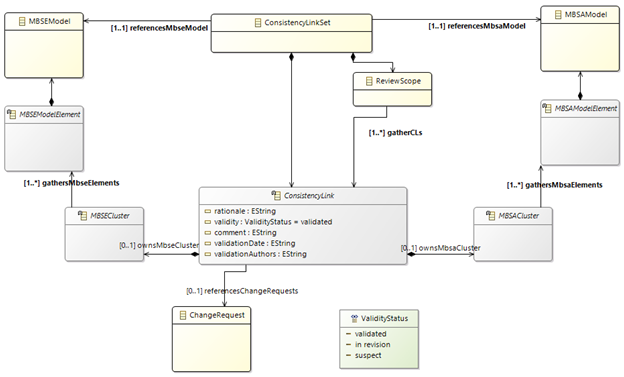

Figure 1: Metamodel of consistency link.

In this paper, the *CL* has been particularized for the functional architecture, although it could be adapted to other viewpoints, such as the logical or the physical architecture. Two types of CL are defined : *CL* for Functions (*CLF*), and *CL* for functional flows (*CLfl*). These concepts are illustrated on figures 2 and 3 below.

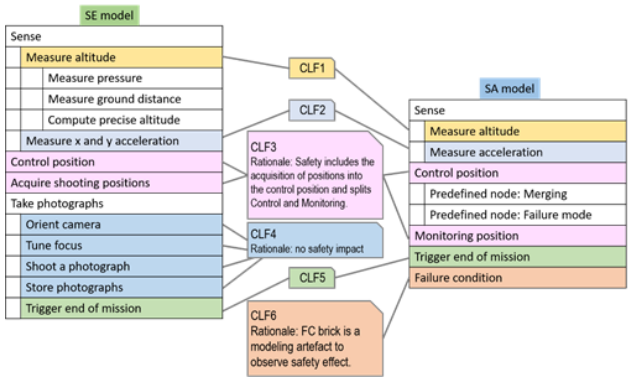

Figure 2: Consistency links for functions.

Coverage and consistency rules for these CL have also been defined.

The coverage rules ensure that all leaves of the functional breakdown and all functional flows are covered by one and only one *CL*

- Rule 1 : Each leaf function (i.e. lowest function in functional breakdown structure) of each model shall either be linked by one *CLF*, or have one hierarchical function (at any level of breakdown) that is linked by one *CLF*.
- Rule 2 : Each flow whose source and destination functions are linked to different CLF shall be linked by a *CLfl*.

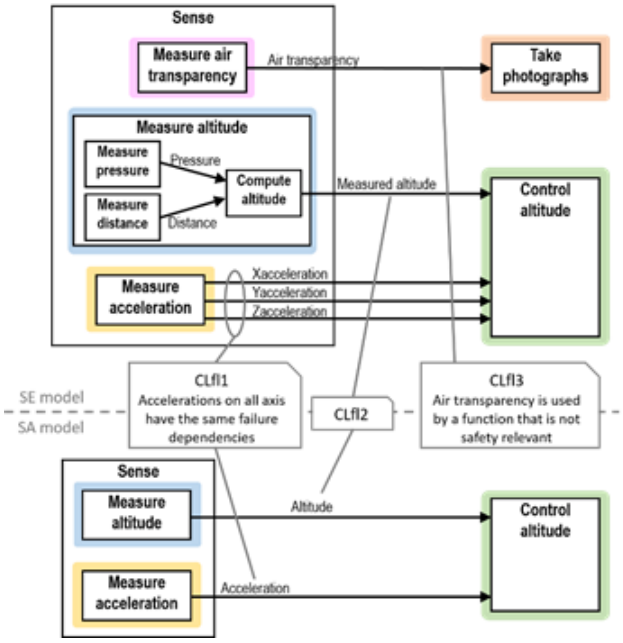

Figure 3: Consistency links for functional flows.

The consistency rules ensure that the defined *CLF* and *CLfl* are globally consistent.

- Rule 3 : In each model, two flows that are linked to 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*.
- Rule 4 : Given a *CLfl*, the source *CLF* from MBSE model shall be the same as the source *CLF* from MBSA model. Symmetrically, the destination *CLF* from MBSE model shall be the same as the destination *CLF* from MBSA model.

Checking that the *CL* set is compliant to these rules can be easily automated.

3 IMPLEMENTATION

The implemented process is illustrated by the figure 4, and starts with the *SE* and *SA* domain's models that are to be *CL*-linked. Constraints CA and CB induce an horizontal "language gap" because methods and tools (*M&T*) differs between domains. To work around this, models are translated automatically to abstracted ones (considering their relative *M&T*). Pragmatically, those new models are compound data flow graphs where edges are only between the more nested nodes. That means hierarchical functions of functional decomposition are the compound nodes with no flows between them. *SEIM* (Systems Engineering Information Model) meta-model rules both new models. It is limited to a subset of concepts required by structural consistency needs due to constraint CC. So, *SEIM* introduces a vertical "language gap", filled by the definition of a transformation logic producing *SEIM* concepts from domain's tools ones. The automation (i.e. concepts' extraction from domain's model then transformation to *SEIM* ones) is the last step of abstraction activity. *SEIM* policy is not to merge both domain's languages

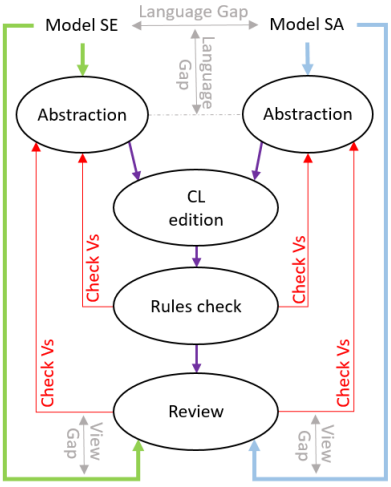

Figure 4: Implementation synopsys.

in an universal one, but rather to be the minimal intersection between domain's meta-models to fulfill targeted needs. This policy reduce the analysis workload on tools' meta-models.

Next process' step is the edition of *CL* via a graphical user interface (*GUI*), see figure 5. Through it, zero or more *SE* abstracted

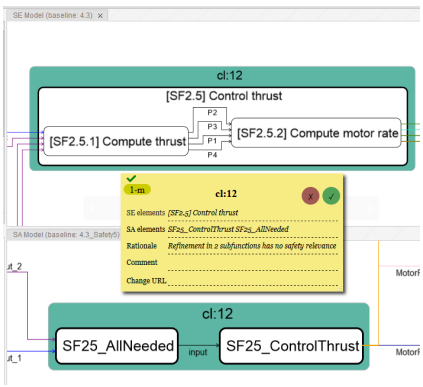

Figure 5: Partial view of the Graphical User Interface.

artefacts are linked with zero or more *SA* abstracted ones. As the new models may have not the same abstraction's depth (inherent to the initial models), abstracted hierarchical compound nodes can be linked too. This allows a customized alignment between abstracted models limited to structure in this paper. This alignment is dependent of the granularity of domain's models, that means a coarsest model will drive the alignment of models. Complementary data (like rationale) are added, also, to get grips with realized grouping for the future *cross-review*. *GUI* has graphical capabilities that auto-layouts part of both abstracted models simultaneously, so that editor can navigate freely through them or redisplay artefacts he grouped previously via *CL*. At each saving action, the defined rules are automatically checked using *CL* and models. Linking may reveal inconsistencies originated by erroneous *CL* edition or by inconsistencies between models. In the first case, *CL* are updated while correction and publication of model has to be done for the second one. At end of the step, both disjunctive partition of the respective intermediate and abstracted models is reached.

When all rules are passed, *cross-review* between *SE* and *SA* experts starts from *CL* editor's proposed-partitioning. *GUI* graphical capabilities are used again so that both contributors share the same common representation. As the *GUI* represents an abstraction of original models, it exists a "view gap" (see 4). For Structural concerns, this discrepancy is limited. Round trips between authoring tools and *GUI* remain easy. After experts agree on a *CL*-scoped consistency they change its status and update possibly its rationale too. When all *CL*-scoped *cross-review* are done, a global and justified consistency status can be acted regarding the models.

Implementation considered also the iteration problem of models and *CL*. During design phase, models are updated (corrected, enlarged, etc,...) so the consistency status becomes suspicious at each evolution. But for the versioned models and *CL*, the rules of method and the automated step are reused to identify the flaws and/or corrupted *CL*. So incremental *cross-review* can be achieved by *SE* and *SA* experts to foster inspection only on impacted part of their models.

4 VALIDATION

In order to assess the feasibility and evaluate the gains of the method presented in 2 and implemented in the consistency management tool as described in 3, it has been experimented with the AIDA case study [1]. This case study is a drone system which aims at assisting the pre-flight check of a commercial aircraft. It consists mainly in an architecture description model in Capella [2] which is used here as the MBSE model. For the purpose of the S2C project, an *MBSA* model has been developed in the SimfiaNeo tool, edited by Apsys[3]. This case study is representative of a medium size aeronautical system : obviously not as complex as an aircraft, but with enough depth and complexity to assess the feasibility of the method in an industrial case.

The validation activity has explored several phases of the lifecycle of such models : the initial creation of the *CL* set, the update of the *CL* set following changes in one of the models, the *cross-review* led jointly by *SE* and *SA* specialist to validate each *CL*. The *cross-review* exercise has been done as closest as possible to real conditions, with a system engineer and a safety specialist.

The tables 1 and 2 show some metrics to illustrate the size of the case study. Table 1 shows the number of model items, before and after the abstraction step. This illustrates the benefits of the abstraction step, which "flattens" the flows in the SimfiaNeo model. As Capella already applies the principle of direct flows between leaves elements, the abstraction step does not further reduce the number of flows in the *MBSE* abstracted model. Table 2 shows the number of *CL* created along with their cardinality (number of elements from each model in a *CL*). It illustrates the flexibility proposed by the method, which enables to associate any number of elements from each model in a same *CL*. In particular, the possibility

[1] AIDA is a public case study developped by the System Engineering center of competence of IRT Saint Exupéry. It is fully open-source and available here : https://sahara.irt-saintexupery.com/AIDA/
[2] https://www.eclipse.org/capella/
[3] https://www.apsys-airbus.com/

Type of model element	*MBSE* model	Abstracted *MBSE* model	*MBSA* model	Abstracted *MBSA* model
Functions	159	159	148	148
Funct. flows	285	285	438	196

Table 1: Complexity of the *MBSE* and *MBSA* models

Cardinality	Numb. of *CLF*	Numb. of *CLfl*
1 *MBSE* to 1 *MBSA*	33	60
1 *MBSE* to n *MBSA*	2	7
n *MBSE* to 1 *MBSA*	7	20
n *MBSE* to m *MBSA*	2	3
0 *MBSE* to 1..n *MBSA*	1	9
1..n *MBSE* to 0 *MBSA*	6	36
Total	**51**	**135**

Table 2: Complexity of the resulting CLset

to associate elements from one model only to a *CL* (i.e. those for which the cardinality is 0-1..n) is useful for model elements that are relevant in only one of the model. For example :

- The *MBSE* model may contain functions that have no safety impact and are not represented in the *MBSA* model. This can occur when a preliminary analysis, such as a Functional Hazard Assessment (FHA), has been realised, or thanks to "expert" knowledge of the system.
- The *MBSA* model contains *SA* specific artifacts, such as the failure conditions observers, that are not represented in the *MBSE* model.

The Rationale attribute of the *CL* allows to capitalize the modelling choices and associated assumptions. The *cross-review* will particularly focus on the validation of these "not 1-1" *CL*.

The validation activity has shown the following qualitative gains for the proposed method :

- the *CL* comes on top of the existing *MBSE* and *MBSA* models, without generating additional modeling constraints,
- the coverage and consistency rules associated to the *CL* set have shown efficiency in the detection of mismatches in the flow consistency, while being flexible enough to address a large number of model elements at the desired level of details,
- the *CL* are helpful for the detection and propagation of model changes,
- the *CL* offer a structure for an efficient *cross-review* focused on model changes,
- the tooling support for consistency link definition and *cross-review* is feasible outside the captive authoring tool, although the developed tool could be matured for a better user experience,
- the *CL* are relevant for discussions and justifications capitalization.

Globally, the *CL* method has proven to be useful to increase the confidence in the structural consistency of models. The induced workload may be slightly increased, which can be put in balance with the avoidance of running future biased analyses due to inconsistent models.

5 PERSPECTIVES AND FUTURE WORK

The work presented here is a first attempt to ensure the consistency between *MBSE* and *MBSA* models. It focuses on the structural consistency of the functional architecture. The tool implementing the method has the maturity of a Proof Of Concept.

Several axes of improvement can be identified :

- The method could be extended to cover also the logical and physical architectures. Topics such as the allocation of functions on logical or physical components could be addressed. This would be particularly relevant as the safety assessment is usually performed at those levels of representation, and not only on the functional aspects.
- The tool can be improved in order to provide better user experience : rationalization of the displayed information, improvement of navigation and user displayed messages. Additional capabilities such as report edition or assistance algorithm for the creation of *CL* (ex: *CL* suggestion based on the similarities of the objects names in both models) could also be considered.

In addition of the local consistency handled by the CL at structural level, it is important to assess the consistency between the *MBSE* and *MBSA* behavioral level. Two approaches are possible either by simulation on overall models or by static local analysis on common models perimeters.

6 CONCLUSION

With the emergence of model-based approaches for the design of complex systems, and because these systems are sometimes subject to strong safety requirements emitted by the regulation authorities, the problem of ensuring the consistency between *MBSE* and *MBSA* models of a same system arises.

Within the frame of the S2C project, we proposed a method to address the topic of structural consistency. An object called "Consistency Link" (*CL*) has been defined, and particularized to address the functional architecture. These *CL* are constrained by coverage and consistency rules.

For validation purpose, the method has been implemented in a tool. Although some improvements of the tool are needed to make it usable in a industrial context, it helped to assess the validity of the method.

The AIDA study case has been used to experiment the method. It has shown qualitative gains for the consistency *cross-review* activity, and an overall improvement in the trust one can have in the safety assessment of the system.

The method only addresses at the moment the problem of structural consistency of the functional architecture. While some improvement axes for the method have been identified, the S2C project currently focus on possible approaches to evaluate the behavior consistency.

REFERENCES

[1] T. Prosvirnova, E. Saez, C. Seguin, and P. Virelizier. Handling consistency between safety and system models. In *IMBSA 2017 (International Symposium on Model-Based and Assessment)*, pages pp.19–34.

[2] Anthony Legendre. Ingénierie système et sûreté de fonctionnement : Méthodologie de synchronisation des modèles d'architecture et d'analyse de risques.

[3] Michel Batteux, Tatiana Prosvirnova, and Antoine Rauzy. System structure modeling language (s2ml). 2015.

[4] Michel Batteux, Tatiana Prosvirnova, and Antoine Rauzy. Model synchronization: A formal framework for the management of heterogeneous models. In Yiannis Papadopoulos, Koorosh Aslansefat, Panagiotis Katsaros, and Marco Bozzano, editors, *Model-Based Safety and Assessment*, pages 157–172, Cham, 2019. Springer International Publishing. ISBN 978-3-030-32872-6.

[5] Laurent Wouters, Stephen Creff, Emma Effa, and Ali Koudri. Collaborative systems engineering: Issues & challenges. pages 486–491, 04 2017. doi: 10.1109/CSCWD.2017.8066742.